SDI Solution
semantic data integration

КРОСС-ОТРАСЛЕВАЯ ПЛАТФОРМА «SEMANTIC»

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления.
Никакая часть данного документа не может быть воспроизведена или передана в любой форме и любыми способами в каких-либо целях без письменного разрешения АО "ЭСДИАЙ СОЛЮШЕН".
© "ЭСДИАЙ СОЛЮШЕН". Все права защищены.

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ **3**

Введение **17**

Термины и определения **18**

1 Назначение и условия применения Системы **22**

2 Подготовка к работе **24**

2.1 Состав и содержание дистрибутивного носителя данных 24

2.2 Порядок загрузки данных и программ 24

2.3 Порядок проверки работоспособности 24

3 Описание операций **25**

3.1 Запуск приложения 25

3.2 Интерфейс облачного решения Системы 27

3.2.1 Стартовая страница *Системы* 28

3.2.2 Представление справочников 30

3.2.3 Вид рабочего окна 33

3.2.4 Пользователь *Системы* 34

3.3 Дерево объектов НСИ и групп 37

3.3.1 Закрепление справочной группы 38

3.3.2 Взаимосвязи между группами 39

3.4 Поиск в дереве справочных групп. «Худое» дерево 43

3.5 Представление данных в таблице 44

3.5.1 Инструментальная панель таблицы данных 46

3.5.2 Добавление и удаление колонок на сетке данных 47

3.6 Работа с информацией в объектах НСИ 51

3.6.1 Карточка группы 51

3.6.2 Карточка иерархического классификатора 55
3.6.3 Карточка записи 57
3.6.4 Сравнение информации об объектах 62
3.6.5 Полное описание справочного объекта 65
3.6.6 Открытие документов ассоциированными приложениями 66
3.7 Поиск данных 67
3.7.1 Поиск по всем справочникам *Системы* 67
3.7.2 Сортировка записей на сетке данных 68
3.7.3 Использование фасетного поиска 68
3.7.4 Быстрый поиск 70
3.7.5 Расширенный поиск 72
3.8 Использование Избранного 84
3.8.1 Добавление информации в Избранное 84
3.8.2 Создание новой папки в разделе «Избранное» 85
3.8.3 Редактирование избранного 86
3.8.4 Раздел «Избранное» 87
3.8.5 Работа с избранным на стартовой странице 88
3.9 Создание пользовательского объекта НСИ 88
3.10 Группирование объектов НСИ 91
3.11 Интеграция справочников в SolidWorks 92
3.11.1 Указание материала в чертеже изделия 92
3.11.2 Добавление материала к 3D-модели 94
3.12 Работа с заявками 96
3.12.1 Поиск по заявкам 100
3.12.2 Создание одиночной заявки 103

3.12.3 Создание групповой заявки 112
3.12.4 Диспетчеризация заявок 114
3.12.5 Менеджер заявок 117
3.12.6 Работа Эксперта НСИ с заявками 118
3.12.7 Работа Согласующего с заявками 128
3.12.8 Статусы заявок 131
3.12.9 Настройка рабочего расписания 136
3.12.10 Настройка причины возврата заявок 140
3.12.11 Приоритеты 143
3.12.12 Отчёт SLA 148
3.12.13 Шаблоны заявок 153
3.12.13.1 Методика разработки шаблонов заявок 153
3.12.14 Создание новой группы шаблонов 154
3.12.15 Новый шаблон заявки 155
3.12.16 Настройка атрибутов заявки 160
3.12.17 Настройка атрибутов позиции 166
3.12.18 Эксперт НСИ и диспетчер заявок 175
3.12.19 Группы пользователей в заявках 176
3.12.20 Формулы обработки заявки 178
3.12.21 Маршрут согласования. Редактор этапов 180
3.12.22 Разработка маршрута согласования в карточке шаблона 180
3.12.23 Разработка маршрута согласования в Редакторе этапов 185
3.12.24 Условия включения в маршрут 188
3.12.25 Сроки обработки и согласования заявок. Уведомления о сроках 194
3.12.26 Аннулирование заявок 197

3.12.27 Настройка видимости 199
3.12.28 Экспорт / импорт шаблона заявки через XML 199
3.13 Редактирование справочных данных 201
3.13.1 Управление пакетами изменений 201
3.13.2 Согласование пакетов изменений 204
3.13.3 Редактирование значений атрибутов в карточке объекта 205
3.13.4 Редактирование значений атрибутов в таблице 207
3.13.5 Массовая вставка и редактирование значений атрибутов 208
3.13.6 Редактирование позиций, созданных по ограничительным таблицам 209
3.13.7 Редактор связей 211
3.13.8 Декларативная совместимость позиций 212
3.13.9 Вычисляемая совместимость справочных групп 215
3.13.10 Управление историческими данными 223
3.13.11 История изменений справочных данных 228
3.13.12 Детализация пакета изменений 230
3.14 Создание новых групп и позиций 241
3.14.1 Создание новой справочной группы 241
3.14.2 Перемещение и удаление групп 243
3.14.3 Регистрация новых справочников 245
3.14.4 Декларация атрибутов для позиций справочной группы 246
3.14.5 Копирование атрибутов справочной группы 253
3.14.6 Перемещение атрибутов 254
3.14.7 Настройка атрибутов группы 256
3.14.8 Постобработчик для обогащения данных 258

3.14.9 Настройка прав доступа на атрибуты и группы 265
3.14.10 Настройка ограничений 269
3.14.11 Создание новой позиции в таблице данных 275
3.14.12 Ввод значений с плавающей точкой 276
3.14.13 Назначение цвета позиции 277
3.14.14 Добавление и редактирование изображения объекта 278
3.14.15 Указание даты и времени 279
3.14.16 Перемещение и удаление позиций 279
3.14.17 Мастер создания позиции 282
3.14.18 Настройки справочных групп для работы с заявками 284
3.14.19 Создание справочной группы на основе метаданных 286
3.14.20 Статусы позиций 289
3.15 Дедубликация справочных данных 292
3.15.1 Создание варианта поиска дубликатов 293
3.15.2 Анализ результатов поиска дубликатов 298
3.15.3 Дедубликация выбранной позиции 303
3.15.4 Дедубликация позиций с внешней применяемостью 304
3.15.5 Дедубликация позиций с переходными ключами 306
3.16 Классификатор атрибутов 307
3.16.1 Просмотр списка атрибутов 307
3.16.2 Поиск атрибутов в классификаторе 308
3.16.3 Создание простого атрибута 310
3.16.4 Типы данных атрибутов 313
3.16.5 Создание и настройка атрибута связи. Агрегация и композиция 314
3.16.6 Иерархические связи и создание организационной структуры 319

3.16.7 Настройка атрибута типа «Число с плавающей точкой» 321
3.16.8 Настройка контекстных значений атрибута 323
3.16.9 Маска для значения атрибута 324
3.16.10 Создание атрибутов из справочной группы 325
3.16.11 Загрузка атрибутов в Классификатор 326
3.16.12 Дедубликация атрибутов 327
3.17 Формулы для вычисления значений и обогащения данных .. 331
3.17.1 Описание редактора формул 333
3.17.2 Создание шаблона строки 336
3.17.3 Параметры вывода 337
3.17.4 Условия вывода 339
3.17.5 Тестирование формулы 339
3.18 Ограничительные таблицы 340
3.18.1 Создание новой ограничительной таблицы 342
3.18.2 Настройка ограничительной таблицы 344
3.18.3 Генерация ограничительных таблиц по существующим позициям .. 346
3.19 Экспорт и импорт справочных данных 347
3.19.1 Экспорт и импорт позиций и групп в Excel© и CSV 348
3.19.2 Экспорт и импорт данных в формат XML 354
3.19.3 Шаблон группы 356
3.20 Иерархические классификаторы 358
3.20.1 Создание иерархического классификатора 359
3.20.2 Настройка классификатора 360
3.20.3 Классификация позиций 363
3.20.4 Экспорт и импорт классификационных групп 365

3.21 Единицы измерения 366
3.22 Репозиторий семантических связей. Стереотипы совместимости 368
3.23 Интеграция Semantic MDM с Контур.Фокус 373
3.23.1 Работа в Контур.Фокус в режиме online 373
3.23.2 Работа в условиях разделения внутренней и внешней сети 375
3.24 Интеграция справочников в САПР ТП Timeline 376
3.24.1 Выбор и изменение типа производства 379
3.24.2 Добавление сведений о материале и заготовке 380
3.24.3 Создание новых операций в ТП с использованием справочников 381
3.24.4 Редактирование операции 383
3.24.5 Создание новых переходов в операциях 384
3.24.6 Редактирование переходов 386
3.24.7 Добавление оборудования в операцию 386
3.24.8 Добавление оснастки 388
3.25 Отчеты 388
3.25.1 Формирование отчетов 389
3.25.2 Просмотр и выгрузка отчетов 391
3.25.3 Отчёты по заявкам 392
3.26 Качество данных 397
3.26.1 Справочники 399
3.26.2 Внешние системы 401
3.26.3 Атрибуты 403
3.26.4 Выбранный атрибут 406
3.27 Машинное обучение 410

3.27.1 Мониторинг и управление 412
3.27.1.1 Настройка группировки значений атрибутов 418
3.27.1.2 Настройка расписания переобучения 419
3.27.1.3 Управление процессом обучения 421
3.27.1.4 Данные для обучения 422
3.27.1.5 Данные для проверки 428
3.27.1.6 Общая точность предсказания 431
3.27.1.7 Точность предсказания по группам 434
3.28 Настройка пользователей и ролей 436
3.28.1 Привилегии пользователей 437
3.28.2 Создание новой роли 439
3.28.3 Добавление пользователя. Карточка пользователя 440
3.28.4 Загрузка и выгрузка пользователей в Excel 446
3.28.5 Парольная политика 447
3.28.6 Журнал работы пользователей 449
3.28.7 Матрица доступа 453
3.28.8 Статистика входа 461
3.28.9 Оценки приложения пользователями 461
3.28.10 Атрибуты, извлекаемые через LDAP 461
3.28.11 Связанная группа 462
3.29 Active Directory. Авторизация пользователей Windows 462
3.29.1 Настройка авторизации пользователей через Active Directory на сервере 462
3.29.2 Настройка авторизации через Active Directory на клиенте 463
3.30 Импорт и репликация 467
3.30.1 Подключение внешней информационной системы 468
3.30.2 Способ передачи данных 472
3.30.3 Приоритизация экспорта 474

3.30.4 Настройка обменных классов 476
3.30.5 Заявки на импортируемые позиции 482
3.30.6 Копирование обменного класса. Работа через XML 484
3.30.7 Режим экспорта/импорта 485
3.30.8 Настройка связи внешней ИС с обменными классами 489
3.30.9 Настройка безопасности 493
3.30.10 Обработка агрегаций / композиций 495
3.30.11 Просмотр реплицированных объектов 496
3.30.12 Статусы реплицированных позиций 503
3.30.13 Загрузка данных в таблицу импорта из Excel 506
3.30.14 Переходные ключи 507
3.30.15 Отчёты 510
3.30.16 Импорт и репликация в карточке позиции 511
3.30.17 Экспорт таблиц и позиций из справочников 512
3.30.18 Контроль работы интеграций 515
3.30.19 Централизованное управление моделью данных 517
3.30.20 Описание компонентов 521
3.31 Настройки 526
3.31.1 Основные 527
3.31.2 Интеграция 537
3.31.3 Отладка 539
3.31.4 Расширения 540
3.31.4.1 Примеры используемых расширений 542
3.31.4.1.1 Контур.Фокус 542
3.31.4.1.2 Автоматическая обработка заявок 545
3.31.5 Обновления БД 546

3.31.6 Сервисные функции 548
3.31.7 Пользовательские отчеты SemanticReports 549
3.32 Методическое обеспечение 553
3.32.1 Методики обработки заявок 553
3.32.1.1 Заявки с возможностью автоматической обработки 553
3.32.1.1.1 Описание задачи (кейса) 554
3.32.1.1.2 Когда востребовано 554
3.32.1.1.3 Условия использования 554
3.32.1.1.4 Работа с заявкой 555
3.32.1.2 Заявки с динамическим списком атрибутов 558
3.32.1.2.1 Описание задачи (кейса) 558
3.32.1.2.2 Когда востребовано 558
3.32.1.2.3 Настройка шаблона заявки 558
3.32.1.2.4 Работа с заявкой 561
3.33 Скрипты GROOVY 561
3.33.1 Использование скриптов Groovy в вычисляемых атрибутах 562
3.33.1.1 Методы чтения данных из Semantic MDM 562
3.33.1.1.1 Чтение данных о позиции 562
3.33.1.1.2 Чтение значений атрибутов позиции 562
3.33.1.2 Методы для работы с числами (valueDouble) 564
3.33.1.3 Пример решения расчетной задачи по расчету нормы расхода 566
3.33.1.4 Пример решения расчетной задачи по расчету содержания драг. металлов по марке материала 571
3.33.1.5 Пример формирования строкового обозначения емкости для конденсаторов KEMET CDR 01-35 574
3.33.1.6 Пример формирования строкового обозначения максимального рабочего давления для датчиков давления 576
3.33.1.7 Пример форматирования обозначения типоразмера металлической сетки по ГОСТ 3826-82577
3.33.1.8 Пример вывода диапазона измерений для калибров 579
3.33.2 Использование скриптов Groovy для обогащения данных 580
3.33.2.1 Методы для обогащения данных в Semantic MDM 582
3.33.2.2 Методы для чтения данных из Semantic MDM 585
3.33.2.2.1 Чтение данных о позиции 587
3.33.2.2.2 Чтение значений атрибутов позиции 588

3.33.2.2.3 Чтение данных по атрибутам 593
3.33.2.2.4 Чтение данных по группам 597
3.33.2.3 Методы для работы с числами (valueDouble) 597
3.33.2.4 Методы для работы с XML и JSON форматом 601
3.33.2.5 Пример обогащения позиции данными, полученными из ML-сервиса .. 604
3.33.2.6 Пример обогащения позиции данными, полученными из Контур.Фокус 611
3.33.3 Использование скриптов Groovy при выполнении импорта 630
3.33.3.1 Структура сообщения импорта 632
3.33.3.2 Сопоставление импортируемого объекта с позицией Semantic MDM 635
3.33.3.3 Методы для работы с импортируемым объектом 636
3.33.3.4 Методы для чтения данных из Semantic MDM 645
3.33.3.4.1 Чтение данных о позиции 646
3.33.3.4.2 Чтение значений атрибутов позиции 647
3.33.3.4.3 Чтение данных по атрибутам из Semantic MDM 650
3.33.3.4.4 Чтение данных по группам из Semantic MDM 653
3.33.3.5 Методы для работы с XML и JSON форматом 654
3.33.3.6 Методы для подключения к внешним сервисам 657
3.33.3.7 Пример решения задачи сопоставления с позицией вспомогательного справочника 658
3.33.3.8 Пример программного определения группы для создания позиции через импорт 662
3.33.3.9 Пример обогащения данными импортируемого объекта из внешней системы 663
3.33.4 Использование скриптов Groovy при выполнении экспорта 671
3.33.4.1 Структура сообщения экспорта 673
3.33.4.2 Методы для чтения данных из Semantic MDM 682
3.33.4.2.1 Чтение данных о позиции 684
3.33.4.2.2 Чтение значений атрибутов позиции 685
3.33.4.2.3 Чтение данных по атрибутам из Semantic MDM 690
3.33.4.2.4 Чтение данных по группам из Semantic MDM 693
3.33.4.3 Методы для работы с XML и JSON форматом 694
3.33.4.4 Пример экспорта значений атрибутов, на основе которых было сформировано полное наименование позиции 697
3.33.4.5 Пример экспорта содержания драг. металлов, рассчитанного на основе содержания драг. металла в марке 699
3.33.5 Использование скриптов Groovy в условиях репликации 708
3.33.5.1 Методы для чтения данных из Semantic MDM 708

3.33.5.1.1 Чтение данных о позиции 708

3.34 Специальные view в БД Semantic для разработки отчетов.... 709

3.34.1 rpt_application_template: Шаблоны заявок..... 709

3.34.2 rpt_app_property: атрибуты заявки 710

3.34.3 rpt_app_property_declaration: Декларации атрибутов в шаблоне заявки 711

3.34.4 rpt_application: журнал заявок..... 714

3.34.5 rpt_application_attribute: Значения атрибутов заявок..... 716

3.34.6 rpt_application_participant: история изменения статусов заявок и их участников 719

3.34.7 rpt_app_status_journal: журнал заявки 720

3.34.8 rpt_reference_book: реестр справочников 722

3.34.9 rpt_reference_book_content: группы, входящие в справочники 722

3.34.10 rpt_m2_account: реестр учетных записей пользователей 722

3.34.11 rpt_m2_security_group: группы пользователей..... 723

3.34.12 rpt_m2_account_group: Вхождение пользователей в группу пользователей 723

3.34.13 rpt_m2_hierarchyhandle: заголовок группы или позиции 723

3.34.14 rpt_m2_item: позиции справочников..... 724

3.34.15 rtp_m2_classificationnode: группы справочников..... 725

3.34.16 rpt_m2_attributevalue: значения атрибутов позиций..... 725

3.34.17 rpt_m2_aggregation: значения атрибута связи 726

3.34.18 rpt_m2_attributedeclaration: декларации атрибутов в справочниках .. 727

3.34.19 rpt_m2_aggregationdeclaration: декларации атрибутов связи в справочниках 728

3.34.20 rpt_m2_attribute: простые атрибуты для справочников..... 729

3.34.21 rpt_m2_aggregationstereotype: атрибуты связи для справочников 730
3.34.22 rpt_m2_propertygroup: измеряемые сущности 730
3.34.23 rpt_m2_measurementunit: единицы измерений 730
3.34.24 rpt_m2_constraint: Ограничения 731
3.34.25 rpt_m2_constrainttable: Табличные ограничения 731
3.34.26 rpt_m2_tablecolumn: Колонки ограничительной таблицы 732
3.34.27 rpt_constraint_compliance: Соответствие позиций ограничениям 732
3.34.28 rpt_transitional_key: таблица переходных ключей 733
3.34.29 rpt_external_system: внешние системы 733
3.34.30 rpt_exchange_class: обменный класс 734
3.34.31 rpt_similarity_search_option: варианты поиска дубликатов 734
3.34.32 rpt_similarity_check: список позиций, для которых искали дубликаты 735
3.34.33 rpt_similar_item: список найденных дубликатов 735
3.34.34 rpt_m2_transaction: пакеты изменения 735
3.34.35 История по позициям 736
3.34.36 История изменения атрибутов связи 738
3.34.37 История изменения простых атрибутов 740
3.34.38 Хранение данных по заявкам 743
3.34.39 Хранение данных по НСИ 744
4 Аварийные ситуации 745
4.1 Действия при отказах технических средств 745
4.2 Действия по восстановлению информации при отказе системы хранения данных или обнаружении ошибок в данных 746
4.3 Действия при обнаружении несанкционированного вмешательства в данные 746

4.3.1.1 Обеспечение безопасности информации 746
4.3.1.2 Поддержание надлежащего уровня обеспечения безопасности 748

Введение

«Кросс-отраслевая платформа «Semantic» (далее *Система*) реализована на облачной платформе SemanticMDM с интеграцией подсистем «Машинное обучение» и САПР ТП «Timeline». *Система* относится к классу систем – MasterDataManagement и предназначена для организации единой системы управления нормативно-справочной информацией.

Содержание настоящего Руководства ориентировано на пользователей, которые умеют работать с браузерными приложениями.

Перед началом пользования «Кросс-отраслевой платформы «Semantic» пользователю необходимо:

1. Ознакомиться с настоящим руководством;

2.Получить доступ – логин и пароль учетной записи.

Термины и определения

Термины и определения, используемые в данном Руководстве приведены в Таблице 1.

Таблица 1. Термины и определения

п/п	Термин	Определение
1	CRON	Утилита (классический демон) в Linux и других UNIX-подобных системах, запускаемая самой системой и работающая в фоновом режиме без прямого взаимодействия с пользователем, позволяющая выполнять скрипты на сервере в назначенное время с заранее определенной периодичностью
2	GUID	GloballyUniqueIdentifier – глобальный идентификатор, который однозначно определяет эталонную запись в MDM-системе
3	Агрегация	Тип связи между двумя объектами НСИ, когда один объект содержит ссылку на значение атрибута (атрибутов) другого объекта
4	Айтем (Item)	Любая запись в системе
5	Атрибут объекта	Элементарная и неделимая часть информации, описывающая информационный объект, например, «плотность», «длина» и т.п.
6	База данных	База данных, служащая для централизованного хранения и предоставления клиентам справочных данных

п/п	Термин	Определение
7	Группа классификационная	Тип объекта НСИ, описывающий подмножество дочерних объектов НСИ, объединённых по определённым признакам сходства
8	Декларация	Атрибут, назначенный на конкретную справочную группу
9	Декларативная совместимость	Совместимость записей справочных групп, которая определяется пользователем вручную или импортируется из внешней системы
10	Классификация	Объединение объектов НСИ в группы по общим признакам
11	Композиция	Связь между справочными объектами типа «родитель – подчинённый», при которой удаление «родителя» ведёт к автоматическому удалению всех его подчинённых
12	Композируемая запись	Запись созданная во вспомогательном справочнике из родительской (композирующей) записи при помощи атрибута связи типа "композиция"
13	Нода	Группа записей, имеющая общего родителя.
14	Объект НСИ	Элемент справочной системы, описывающий своими атрибутами предмет материального (сортамент) или нематериального мира (код классификации)

п/п	Термин	Определение
15	Совместимость	Семантическая связь между справочными объектами, например, определение одного объекта, как аналога другого (в определённом контексте), создаёт между ними семантическую связь
16	Справочник	Массив объектов НСИ, выделенный в отдельную часть по определённым классификационным правилам
17	«Худое дерево»	Иерархическое представление справочных групп, при котором в выборку попадают только те группы, которые содержат записи, удовлетворяющие критериям поиска
18	Маппинг значений	Сопоставление, определение соответствия значений

В таблице 2 представлены основные сокращения, используемые в настоящем документе.

Таблица 2. Основные сокращения

п/п	Сокращение	Расшифровка
1	БД	База данных
2	ИУС	Информационно-управляющая система
3	НСИ	Нормативно-справочная информация
4	ПК	Персональный компьютер
5	ПО	Программное обеспечение

п/п	Сокращение	Расшифровка
6	ПИВИИ	Подсистема интеграции с внешними источниками информации
7	СВНСИ	Специалист по ведению НСИ
8	МУЗ	Модуль управления заявками (Журнал заявок)
9	ИС	Информационная система
10	ЕИ	Единица измерения
11	НСД	Несанкционированный доступ к *Системе*
12	Drag&Drop	Дословно переводится как «перетяни и кинь» или «бери и брось». По сути, это способ манипулировать элементами на экране устройства. Манипуляции происходят при помощи «мыши», которая «захватывает» элемент и переносит его в нужную часть экрана. «Захват» элемента происходит при помощи клика и удержания левой кнопки мыши в момент наведения курсора на элемент. Потом в состоянии «нажатого клика» элемент переносится в нужную часть экрана и в момент «отжатия клика» становится на новое место. Drag-and-drop нужно для простого действия — перенести объект на экране.
13	mdmId	Строка в сообщениях импорта / экспорта, содержащая идентификатор (GUID) объекта в *Системе*

1 Назначение и условия применения Системы

Система относится к классу систем – Master Data Management и предназначена для организации единой системы управления нормативно-справочной информацией. Система обеспечивает централизованное хранение и предоставление НСИ всем заинтересованным пользователям и автоматизированным приложениям.

«Кросс-отраслевая платформа «Semantic» реализована на облачной платформе Semantic MDM с интеграцией подсистемы «Машинное обучение» (далее – Подсистема ML), Система позволяет визуализировать текстовую и графическую информацию, просматривать характеристики различных позиций, включая нормативно-техническую документацию, производить фасетный и многокритериальный поиск по унифицированным характеристикам изделий. Уникальность системы заключается в реализованной объектной модели мастер-данных, которая позволяет сначала унифицировать всю терминологию предметной области, а затем на основе единого глоссария создавать различные объекты классификации с произвольным набором характеристик. Объекты различных справочников в Системе могут объединяться связями различных типов: агрегация, композиция и ассоциация, благодаря которым пользователь может перемещаться от одного объекта к другому в семантической сети мастер-данных.

Работа пользователя с Системой возможна при выполнении требований к программно-аппаратному обеспечению рабочей станции пользователя, представленной Таблица 3.

Таблица 3. Требования к аппаратно-программным средствам клиентской части

Параметр	Минимальные требования	Рекомендуемые разработчиком
Операционная система (ОС)	Windows XP	Windows 10 x64
Центральный процессор (CPU)	Intel Core i3-2xxx	Intel Core 2 Duo и лучше
Оперативная память (RAM)	2048 MB и выше	4 ГБ
Жёсткий диск (HDD)	-	40 GB
Разрешение экрана	1280×720	1920x1080
Веб-браузер	Mozilla Firefox (52+)	Google Chrome (60+) / Mozilla Firefox (56+) / Opera (49+) / Yandex Browser (17+) / Edge (41+), кроме Internet Explorer
Прочее ПО	-	MS Office / OpenOffice / LibreOffice
Сеть (LAN)	5 Мбит/с	10 Мбит/с и выше

Требования к аппаратно-программным средствам серверной части Системы представлены Таблица 4.

Таблица 4. Требования к аппаратно-программным средствам серверной части

Параметр	Минимальные требования
Операционная система (ОС)	AstraLinux CE 2.12 и выше
Центральный процессор (CPU)	Intel Core I7
Оперативная память (RAM)	32 GB
Жесткий диск (HDD)	128 GB
СУБД	СУБД Postgre Pro 13 и выше
Система контейнеризации	Docker-engine (не ниже версии 18.0.6)
Сеть (LAN)	100 Мбит/с

2 Подготовка к работе

2.1 Состав и содержание дистрибутивного носителя данных

В состав дистрибутивного носителя данных *Системы* входят:

- Postgre Pro 15
- SemanticSearch 23.2
- Wildfly 28, OpenJDK 18
- SemanticMQ 23.2
- SemanticReports 23.2
- Python 3.8
- SemanticBLOBs 23.2
- NGINX 1.20.1

2.2 Порядок загрузки данных и программ

Общая последовательность установки *Системы*, следующая:

1) Установка сервера БД Postgre Pro

2) Подготовка к запуску Docker-контейнеров

3) Настройка и запуск Docker-контейнеров

2.3 Порядок проверки работоспособности

Для проверки работоспособности необходимо зайти в *Систему* с административной учётной записью, обеспечивающей доступ ко всему функционалу *Системы*. Последовательно выполнить следующие действия:

- создание и регистрацию нового справочника;
- создание новой справочной группы;
- создание и декларация нового атрибута;
- добавление / изменение записи;
- публикация изменений.

Полная проверка работоспособности выполняется в процессе опытной эксплуатации *Системы*.

3 Описание операций

3.1 Запуск приложения

Для входа в Систему пользователь должен быть зарегистрирован в ней и иметь логин и пароль (см. п. 3.28).

В Semantic MDM может быть разрешена самостоятельная регистрация (см. п. 3.31.1). В этом случае необходимо, по соответствующей ссылке в браузере, зарегистрироваться на форме авторизации, подтверждение о регистрации будет отправлено на указанную электронную почту.

Для пользователей Windows возможен вход с паролем от операционной системы, если такая возможность настроена системным администратором (см. п 3.29).

Также, в некоторых случаях, может быть разрешён гостевой доступ для ознакомления с работой Системы. Для такого доступа на странице авторизации появится специальная кнопка Гостевой доступ (см. п. 3.31.1).

Система Semantic MDM загружается в браузере пользователя. Список поддерживаемых браузеров указан в требованиях к программному обеспечению.

Идентификация пользователя, запустившего Систему, осуществляется в диалоговом режиме.

Для входа в Систему следует на странице авторизации (Рисунок 1) ввести в соответствующих строках имя пользователя (логин, адрес электронной почты) и пароль и нажать кнопку **Войти**.

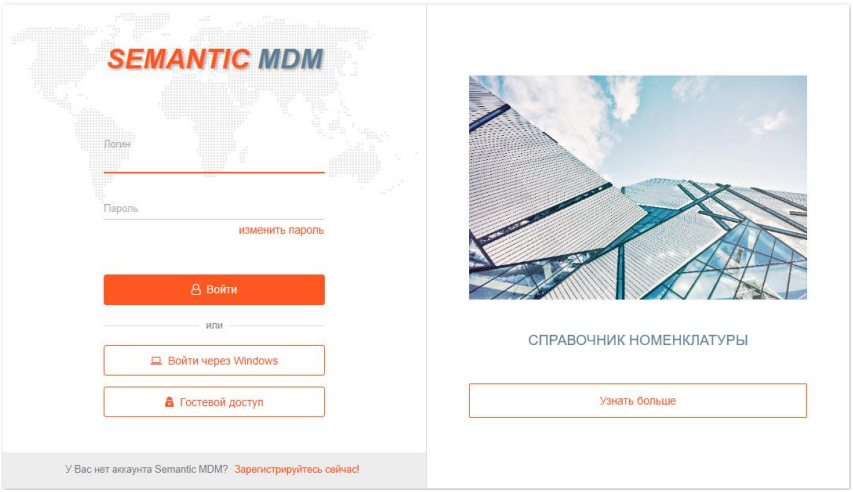


Рисунок 1 - Страница Авторизация пользователя

Команда изменить пароль предназначена для вызова диалога смены пароля. В диалоге следует указать логин и старый пароль, а затем ввести и подтвердить новый, после чего нажать кнопку **Изменить**.

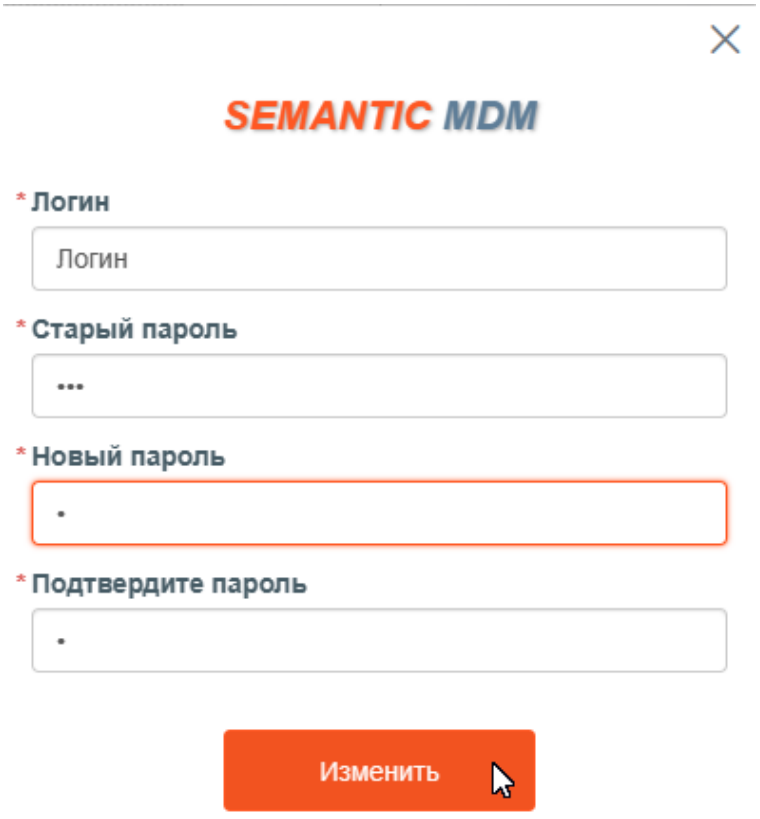


Рисунок 2 – Изменение пароля

В правой части страницы авторизации расположена кнопка Узнать больше (зависит от настройки интерфейса, подробнее см. п. «Основные»),

которая перенаправит пользователя на сайт системы управления НСИ Semantic MDM.

3.2 Интерфейс облачного решения Системы

Система имеет следующие подсистемы:

- **Стартовая страница** (см. п. 3.2.1);
- **Справочники** (см. п. 3.2.2)
- **Журнал заявок** (см. раздел 3.12);
- **Конфигурация данных** (см. разделы 3.16);
- **Импорт и репликация** (см. раздел 3.30);
- **Отчеты** (см. раздел 3.25);
- **Качество данных** (см. раздел 3.26);
- **Машинное обучение** (см. раздел 3.27);
- **Администрирование** (см. раздел 3.28);
- **Описание компонентов (**см. п. 3.30.20);
- **Настройки** (см. раздел 3.31).

Для выбора подсистемы необходимо щёлкнуть левой кнопкой мыши в области навигации (Рисунок 3), расположенной в левом углу рабочего окна, и активировать нужный раздел. Доступ и видимость в навигационном списке различных подсистем определяется ролью пользователя с соответствующим набором привилегий (см. п. 3.28.1).

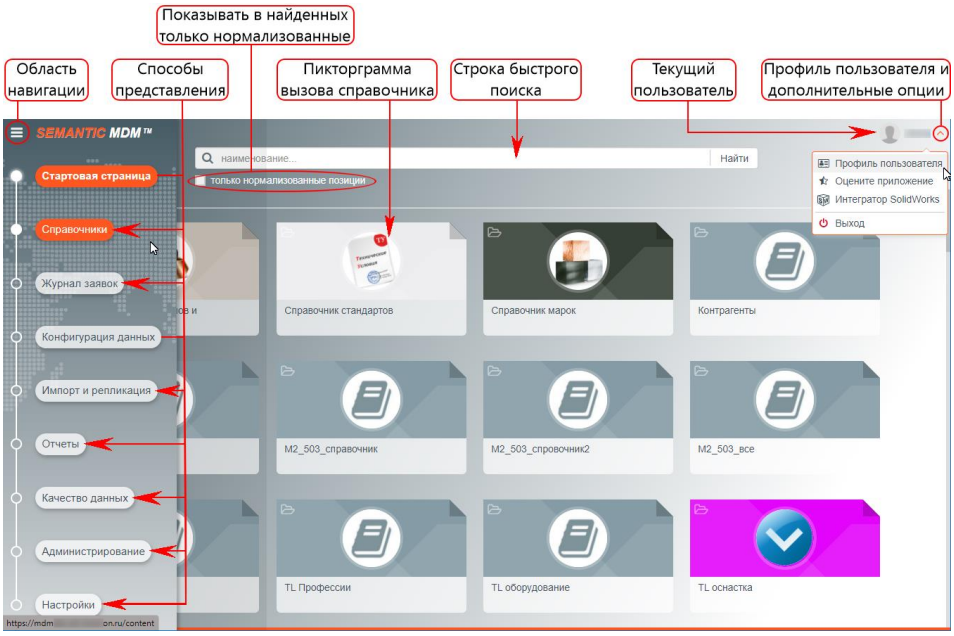


Рисунок 3 - Панель навигации для выбора подсистемы и доступ к профилю пользователя

3.2.1 Стартовая страница *Системы*

При запуске *Semantic MDM*, а также выборе раздела **Стартовая страница** (Рисунок 4) будет открыто начальное окно, которое содержит кнопки с пиктограммами справочников.

Стартовая страница *Системы* имеет следующие основные компоненты:

- кнопка навигации по подсистемам, присутствующая в любом окне;
- строка быстрого поиска по всем справочникам (доступным пользователю) с маркером, включающим показ только нормализованных позиций среди найденных;
- рабочая область с пиктограммами (кнопками вызова) справочников;
- сведения о текущем пользователе.

Чтобы открыть справочник необходимо нажать на кнопку с соответствующим изображением. Если справочников много, то они могут быть сгруппированы в виде папок (подробнее см. п 3.10). Для вызова справочника, находящегося в папке необходимо вначале щёлкнуть левой кнопкой мыши по пиктограмме группы, а затем по пиктограмме самого

справочника. Справочники и классификаторы (см. раздел 3.20) собраны в отдельные группы для удобства пользователей.

В области справочников отображаются пиктограммы тех справочников и классификаторов, которые доступны текущему пользователю. Доступ к справочникам и классификаторам для отдельных пользователей и групп настраивается администратором *Системы*.

Отдельная пиктограмма **Все справочники** обеспечивает доступ в одном рабочем окне к данным сразу во всех справочниках, разрешённых для текущего пользователя. Классификаторы открываются только со страницы - один классификатор на одной вкладке браузера.

Ниже области с пиктограммами справочников и классификаторов располагается панель Избранное куда пользователь может добавлять наиболее часто используемые объекты из различных справочников. Работа с избранным описана в разделе 3.8.

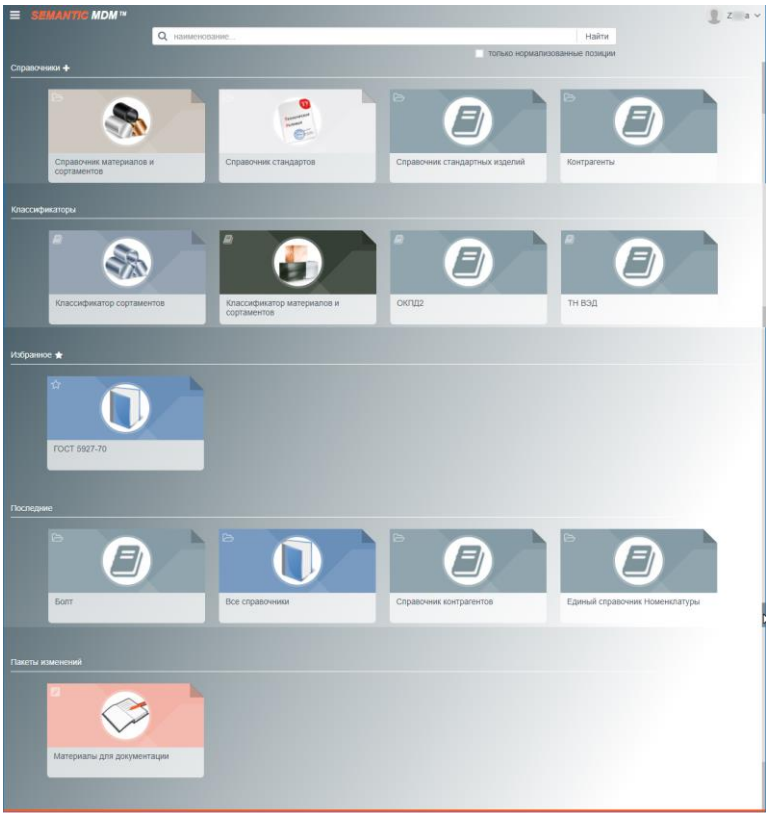


Рисунок 4 – Стартовая страница Системы

Ниже избранного располагается область (панель) с последними открывавшимися справочниками, которая называется *Последние*.

В самом низу располагается панель *Пакеты изменений*, которая доступна пользователям, имеющим привилегию для редактирования справочников (подробнее см. п. 3.13.1).

В правом углу окна располагаются сведения о текущем пользователе (подробнее см. п. 3.2.4).

3.2.2 Представление справочников

Рабочее окно подсистемы Справочники (Рисунок 5) содержит следующие основные компоненты:

- перечень всех справочных объектов по всем справочникам (область окна Дерево);
- строка адреса выбранного объекта;
- поле быстрого поиска;
- доступ к расширенному поиску;
- управление пакетами изменений;
- кнопка добавления объекта в Избранное;
- кнопка доступа к заявкам, требующим обработки (доступна только экспертам НСИ);
- кнопка переключения вида приложения;
- кнопка вызова справки;
- кнопка отправки сообщения в службу техподдержки;
- сведения о текущем пользователе;
- таблица позиций НСИ (область окна Таблица) с собственной панелью инструментов;
- карточка со сведениями о выбранной позиции или группе (область окна *Карточка*).

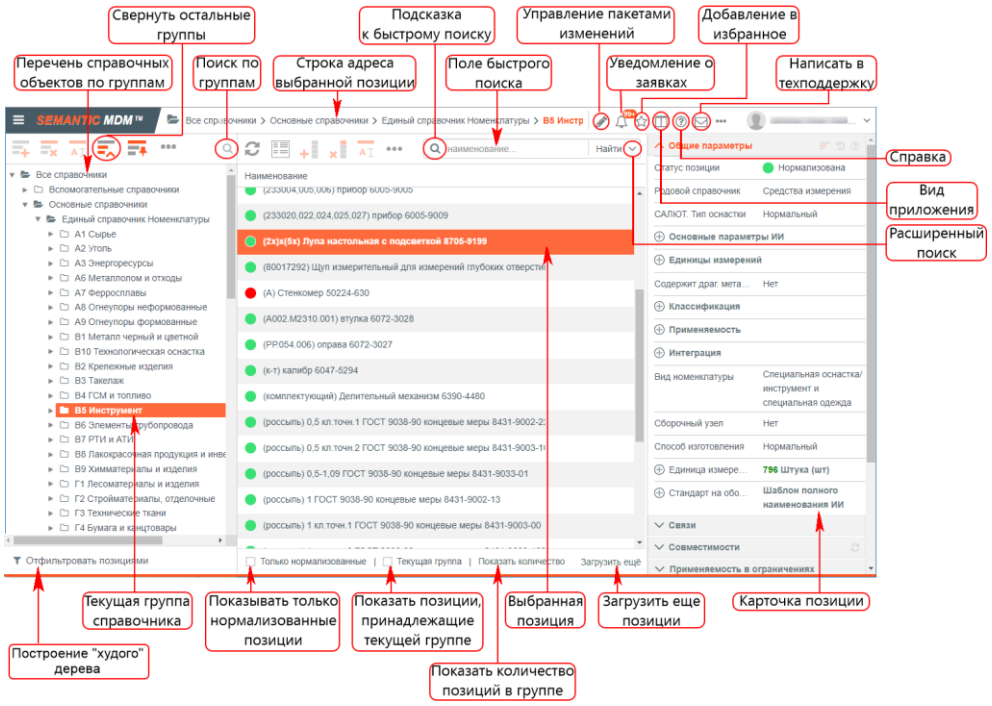


Рисунок 5 - Вид рабочего окна Semantic MDM с открытым справочником

Отдельные панели рабочего окна (дерево, таблица и карточка позиции) разделены подвижными невидимыми границами (сплиттерами). При наведении на сплиттер курсор меняет вид на двунаправленную стрелку. Зажав левую кнопку мыши, можно переместить границы, увеличив или уменьшив видимую область данных. Щелчком левой кнопки мыши по центральной части сплиттера  можно скрыть дерево или карточку. Повторный щелчок восстановит скрытые области.

Перечень групп справочных объектов представляет собой древовидную структуру. Чтобы раскрыть состав какого-либо объекта дерева, нужно выполнить двойной щелчок мышью на объекте, чтобы свернуть состав какого-либо объекта дерева, нужно повторить двойной щелчок мышью. Для развёртывания или сворачивания дерева можно также воспользоваться кнопкой «▶» слева от наименования объекта в дереве.

Дерево справочников имеет собственную панель инструментов, состоящую из следующих кнопок:

Таблица 5. Панель инструментов Дерева справочников

Кнопка		Функция
	Добавить группу	Позволяет создать новую группу в дереве классификатора. Кнопка активна только в режиме редактирования.
	Удалить группу	Позволяет удалить любую группу в дереве классификатора. Кнопка активна только в режиме редактирования.
	Переименовать	Позволяет изменить наименование объекта в дереве. Кнопка активна только в режиме редактирования
	Свернуть остальные группы	Сворачивает все ветви объектов НСИ, кроме той, которая содержит текущий объект (объект дерева, на котором установлен курсор)
	Закрепленные группы	Обеспечивает закрепление выбранных пользователем справочных групп и скрытие всех остальных (см. п.3.3.1)
	Взаимосвязи между группами	Вызывает дополнительное окно **Взаимосвязи между группами**, где можно ознакомиться с существующими в текущей справочной группе связями с другими группами и используемыми для этого атрибутами связи для агрегации / композиции (см. п.3.3.2)
	Поиск по группам	Открывает ячейку для ввода наименования группы для поиска её в классификаторе объекта НСИ

Если панель дерева справочных групп слишком узкая, то не умещающиеся кнопки скрываются под одной – комплексной (Рисунок 6). При увеличении области расположения дерева (путем перемещения сплиттера вправо) кнопки возвращаются к прежнему виду.

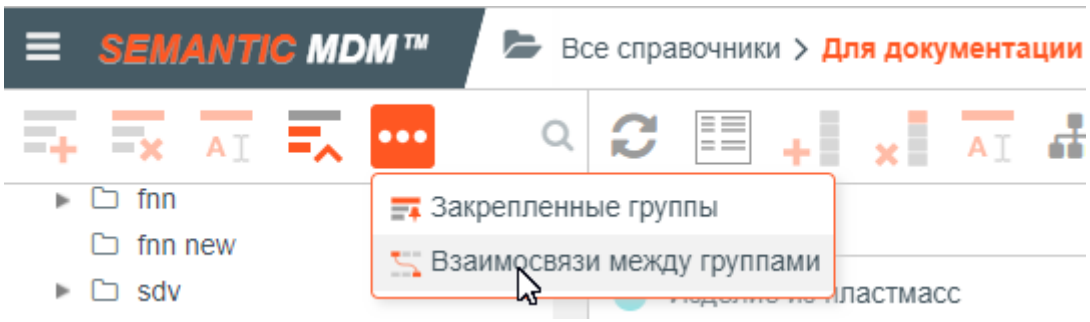


Рисунок 6 - Скрытие кнопок дерева справочников

В **строке адреса выбранного объекта** отображается полный маршрут к объекту дерева, выбранному пользователем. Объекты разных уровней справочника разделяются символом «›». Нажатие на это знак открывает выпадающий список подгрупп, входящих в состав группы, указанной слева от символа «v». Кликнув на наименование подгруппы в выпадающем списке,

пользователь может перейти на другую ветвь (подгруппу) справочника (Рисунок 7).

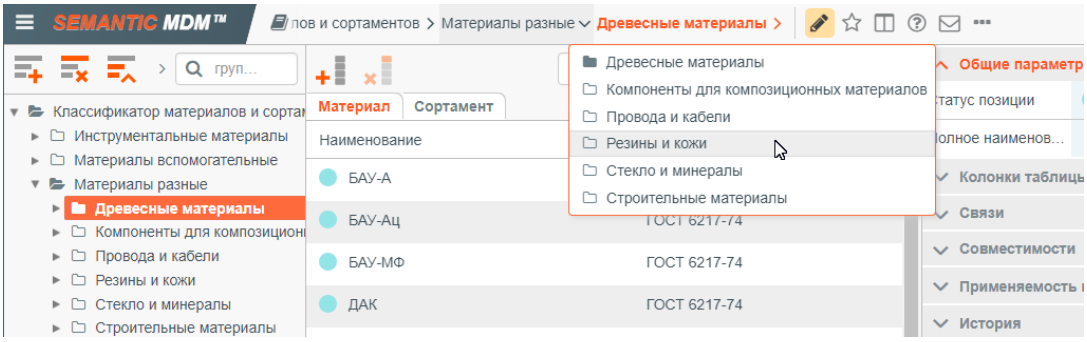


Рисунок 7 - Выбор другой ветки справочника

Поле быстрого поиска служит для ввода искомого значения (см. п. 3.7.4).

Таблица записей содержит все записи, подчинённые выбранной в дереве группе и ее дочерним группам. В нижней части таблицы можно выбрать показывать только нормализованные записи (для пользователей с привилегиями редактирования данных), только записи текущей справочной группы, а также количество записей в выбранной группе и всех её дочерних группах.

Карточка имеет несколько элементов, представление и содержание которых отличается для справочных групп, классификаторов и записей. Описание содержания карточек дано в разделе 3.6.

Описанная структура характерна для всех справочников *Системы*.

3.2.3 Вид рабочего окна

По умолчанию рабочее окно имеет три области: дерево справочника, таблица данных и карточка текущей позиции (группы).

За вид страницы *Системы* отвечает кнопка ***Вид***, расположенная в верхней части рабочего окна (Рисунок 8).

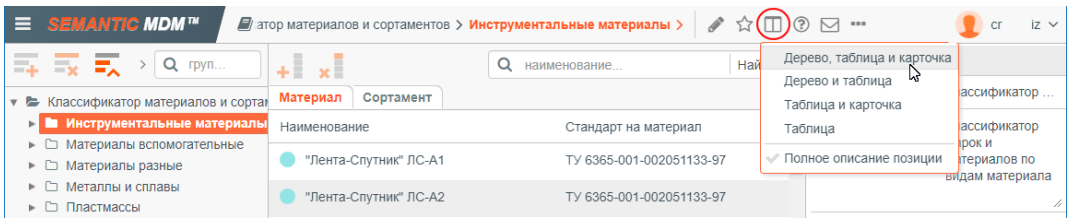


Рисунок 8 - Изменение вида страницы WEB-портала

Чтобы изменить вид страницы достаточно нажать на кнопку ***Вид*** и выбрать один из предлагаемых вариантов. Аналогичные действия могут быть выполнены с использованием невидимых подвижных границ - сплиттеров. При наведении на сплиттер курсор меняет вид на двунаправленную стрелку. Зажав левую кнопку мыши, можно переместить границы, увеличив или уменьшив видимую область данных. Щелчком левой кнопки мыши по центральной части сплиттера  можно скрыть дерево или карточку. Повторный щелчок восстановит скрытые области.

Также по кнопке ***Вид*** можно включить опцию **Полное описание позиции** (см. подробнее п. 3.6.5).

3.2.4 Пользователь *Системы*

Каждый пользователь Системы идентифицируется либо по регистрационным данным, либо как гость (при гостевом доступе). Имя текущего пользователя отображается в правом верхнем углу окна.

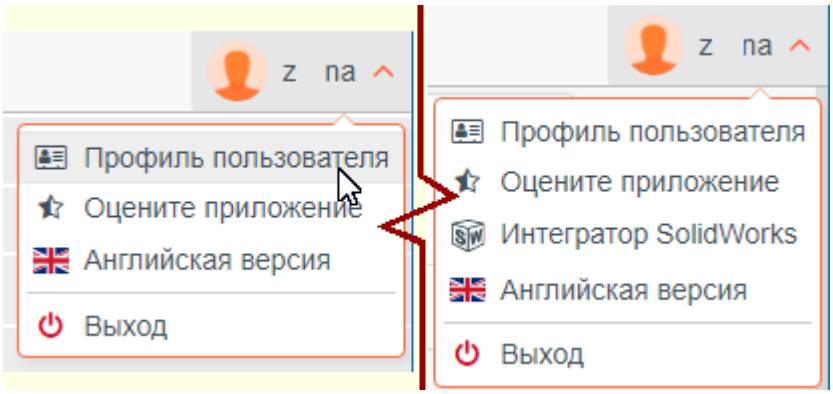


Рисунок 9 - Доступ к профилю пользователя

Щелчком по имени пользователя можно вызвать список доступных настраиваемых опций (Рисунок 9): **Профиль пользователя, Оцените приложение, Английская версия, Интегратор SolidWorks, Выход** .

Профиль пользователя – окно, состоящее из трех вкладок - **Общее**, **Роли** и **Привилегии** (Рисунок 10). На вкладке **Общее** дано описание текущего пользователя с возможностью настроить опцию *Дублировать уведомления МУЗ по email* для дублирования уведомлений о получении заявок для обработки (см. раздел 3.12) на электронную почту (только для пользователей, обрабатывающих или согласовывающих заявки). Кнопка ***Сбросить настройки*** позволяет вернуть состояние профиля пользователя к настройкам по умолчанию.

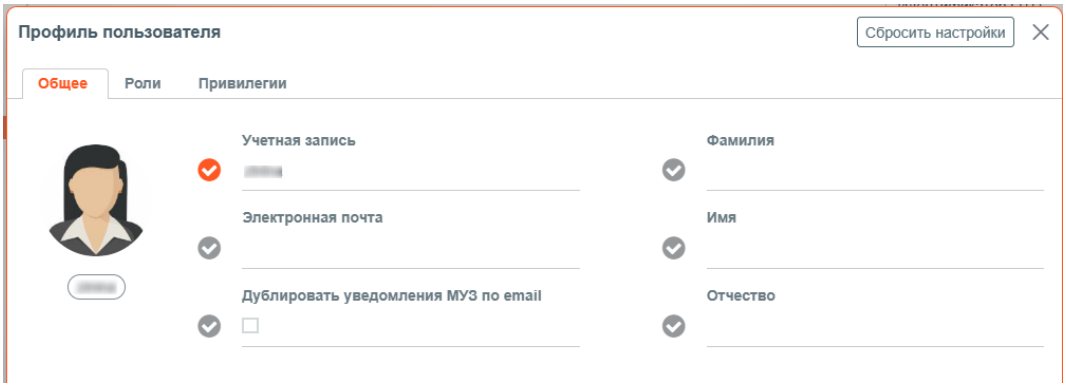


Рисунок 10 - Профиль пользователя. Общие сведения

На вкладке **Роли** располагается список ролей, которые может исполнять текущий пользователь.

Вкладка **Привилегии** (Рисунок 11) содержит перечень привилегий пользователя суммарно по всем ролям, к которым он относится. Также на вкладке можно ознакомиться со сведениями о замещаемых пользователях и собственных заместителях. Привилегии настраиваются для каждой роли (см. раздел 3.28) в подсистеме **Администрирование**. Замещение настраивается в подсистеме **Журнал заявок**.

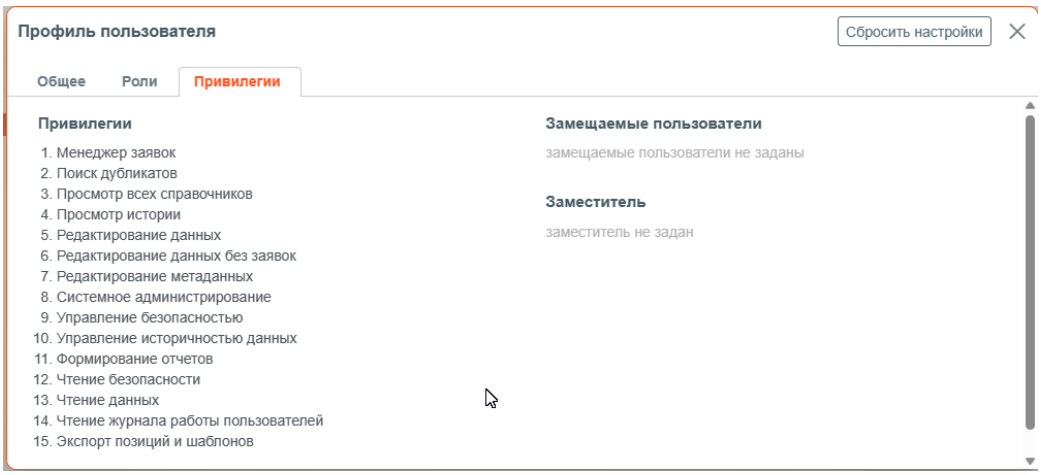


Рисунок 11 - Профиль пользователя. Сведения о привилегиях и замещении

Оцените приложение – вызов окна для оценки приложения, в котором можно указать условную оценку от единицы (одна звёздочка) до пяти и дать развёрнутый комментарий своей оценке (Рисунок 12).

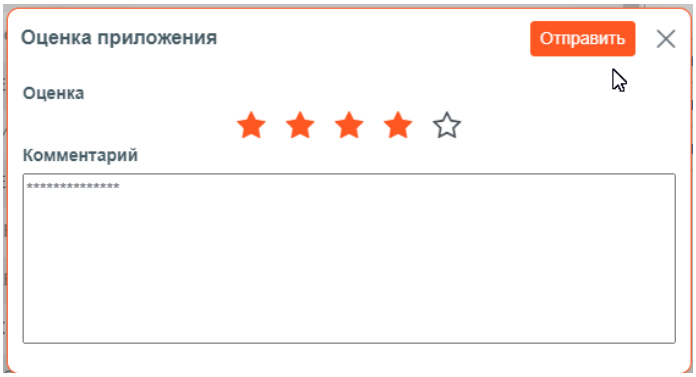


Рисунок 12 - Оценка приложения

Английская версия (Russian version) – переключает интерфейс *Системы* на английский (русский) язык (Рисунок 13).

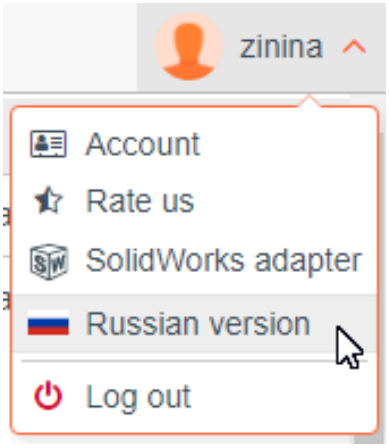


Рисунок 13 - Выбор языка интерфейса

Интегратор SolidWorks – загружает в компьютер пользователя интегратор, необходимый для совместной работы CAD-системы *SolidWorks* и справочников *Semantic MDM* (подробнее см. раздел 3.11)

Выход – выход из *Semantic MDM*. Обеспечивает завершение сеанса работы со справочниками. Если браузер был закрыт без завершения сеанса, то при повторном вызове не потребуется ввод регистрационных данных.

При работе с корпоративной версией *Системы* регистрацией пользователя и назначением роли, которую он выполняет, занимается системный администратор.

3.3 Дерево объектов НСИ и групп

Вся информация, хранящаяся в любом справочнике, имеет древовидную структуру. Пользователь может открыть нужный справочник для просмотра и поиска, нажав на его пиктограмму на стартовой странице. Организация информации во всех справочниках абсолютно идентична – каждый справочник имеет дерево справочных групп, таблицу позиций и карточки, общее описание которых дано в соответствующих разделах.

Дерево справочника представляет собой иерархию классов, объединяющих в группы объекты НСИ по сходству и подобию.

В Системе используются два вида группирования данных. Первый вид – справочные группы, которые содержат необходимую информацию в соответствии со своим назначением и могут объединяться в справочники на основе онтологический классификации, например, *Справочник стандарто*в содержит множество различных стандартов, применяемых для описания и выбора объектов – стандартных изделий, марок материалов и т.п. Справочные группы нужны для идентификации данных экспертом НСИ и имеют оптимальную структуру с точки зрения идентификации информации.

Второй вид группирования – иерархические классификаторы, которые позволяют объединять идентифицированные данные других справочников по любым основаниям. Например, *Классификатор материалов и сортаментов*

содержит сведения о марках материалов и видах сортаментов, изготавливаемых из них, с учётом возможных типоразмеров. Пользователями может быть создано любое количество необходимых классификаторов.

3.3.1 Закрепление справочной группы

Для удобства работы с деревом в Системе существует возможность скрыть неиспользуемые в работе группы справочника. Для чего необходимо установить курсор на группу справочника и, в контекстном меню (под правой кнопкой мыши), выполнить команду **Закрепить группу**. После выполнения команды в инструментальной панели дерева появляется кнопка ***Закреплённые группы***  , предназначенная для быстрого доступа к закреплённым группам из инструментальной панели *Дерева*. По нажатию данной кнопки в дереве скрываются незакреплённые группы и остаются доступными только те, которые были закреплены (Рисунок 14).

Для того, чтобы открепить группу необходимо вызвать контекстное меню и выбрать команду **Открепить группу**.

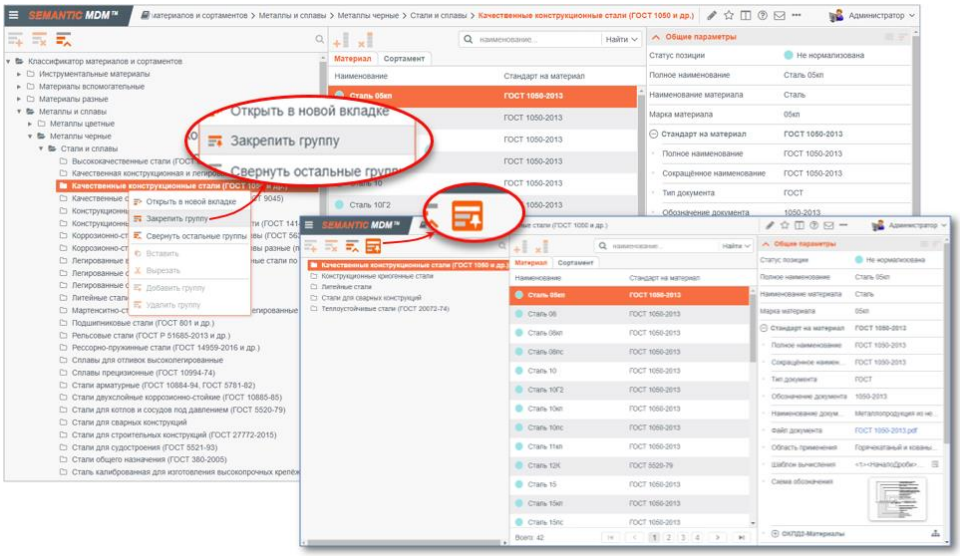


Рисунок 14 - Закрепление справочных групп

3.3.2 Взаимосвязи между группами

В *Semantic MDM* для обеспечения единства представления мастер-данных используются атрибуты связи (см. п. 3.16.5), которые обеспечивают использование позиций одних справочников в качестве значения атрибута других справочников. Например, сопроводительные документы (стандарты, ТУ и т.п.) хранятся в справочнике стандартов, при этом они применяются во множестве других справочных групп в качестве значения атрибута *Сопроводительный документ*. Для этого атрибут имеет настроенную связь (агрегацию) со справочником стандартов. При эксплуатации Системы все справочники «обрастают» множеством связей агрегации / композиции и для их просмотра применяется графический интерфейс.

Отображение взаимосвязей групп через агрегации и композиции и наследование выполнено в виде графической схемы. Назначением данной схемы, являются:

- Выявление и удаление лишних взаимосвязей групп;
- Определения наличия агрегаций / композиций с неопределенной областью.

Для вызова схемы следует установить курсор на нужную справочную группу и нажать кнопку  ***Взаимосвязи между группами***, которая располагается над деревом справочников.

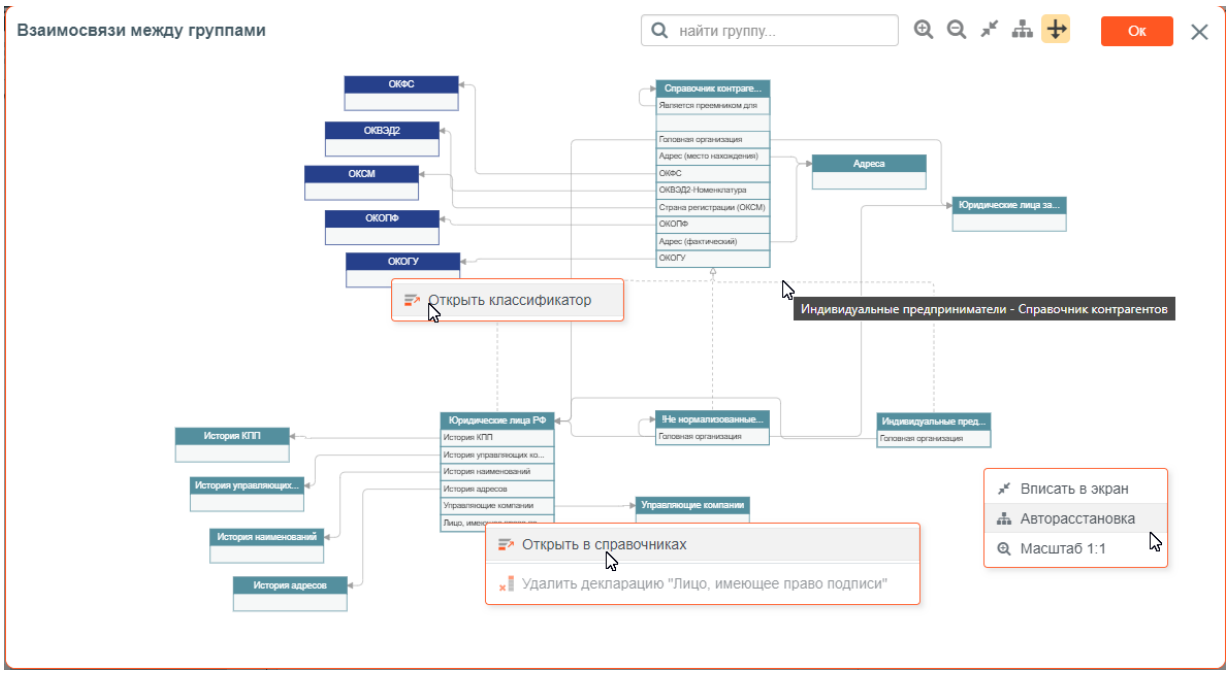


Рисунок 15 - Взаимосвязи между справочными группами

На графическую схему выводятся все группы (справочники и классификаторы), которые имеют хотя бы одну собственную декларацию, а также группы, на которые ссылаются на них через агрегацию / композицию (Рисунок 15). Для каждой из групп, отображается список собственных агрегаций / композиций, от которых в виде сплошной линии отображены связи к заголовку групп, которые указаны в области каждой из агрегаций / композиции. Если для атрибута агрегации / композиции в настройках не определена область, то такая агрегация / композиция будет подсвечена красным цветом.

Связи наследования одной группы от другой представлены на схеме в виде прерывистой линии со стрелкой на родительскую группу. При наведении курсора на линию связи всплывает ее наименование и связанные справочники (для агрегации / композиции) либо <Наименование подчиненной группы> - <Наименование родительской группы> (для связей наследования).

Для удобства просмотра связей в окне Взаимосвязи между группами используются кнопки и команды, представленные в Таблица 6:

Таблица 6. Функции кнопок

Кнопка / Команда		Функция
	**Увеличить масштаб**	Позволяет увеличить масштаб отображения схемы
	Масштаб 1:1	Команда доступна только из контекстного меню. Обеспечивает отображение всех совместимостей в масштабе 1:1, при котором графический элемент имеет размер 34x10 мм (для одного атрибута)
	**Уменьшить масштаб**	Позволяет уменьшить масштаб отображения схемы связей
	**Вписать в экран**	*Система* обеспечивает такой масштаб диаграммы, при котором в окне будут видны все настроенные стереотипы совместимости
	**Авторасстановка**	*Система* располагает объекты автоматически в плоскости экрана. Позволяет «распутать» хаотично расположенные связи. При большом числе связей команда значительно увеличивает область для просмотра и соответственно уменьшает масштаб.
	**Ортогональные линии связи**	Переключает режим отображения линий связи с произвольного на ортогональный и обратно

Классификаторы на схеме выделяются синим цветом, а справочники - серо-зеленым. Непосредственно из схемы можно перейти в интересующую справочную группу или классификатор. Для этого надо вызвать контекстное меню на нужном объекте и выбрать команду **Открыть в справочниках** / **Открыть классификатор**.

Окно быстрого поиска **найти группу…** позволяет найти справочную группу по наименованию и спозиционировать курсор на ней, при этом все линии связи этой группы будут подсвечены красным цветом (Рисунок 16). Линии связи также подсвечиваются при установке курсора на любой группе вручную.

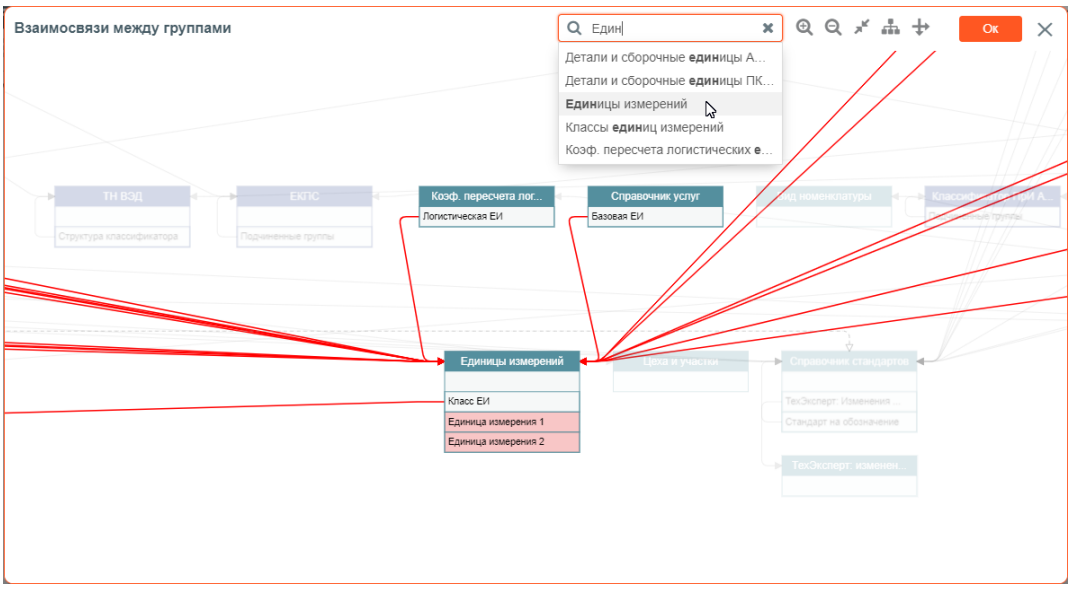


Рисунок 16 - Поиск справочной группы и просмотр её связей

Система позволяет выполнить удаление лишних агрегаций / композиций непосредственно на графической схеме. Для этого нужно:

- перейти в режим управления пакетами изменений (см. п. 3.13.1);
- установить курсор на интересующей справочной группе и нажать кнопку Взаимосвязи между группами;
- в открывшемся окне со схемой навести курсор на интересующую декларацию атрибута (см. раздел 3.13) и выбрать в контекстном меню команду Удалить декларацию <наименование декларации> (Рисунок 17).

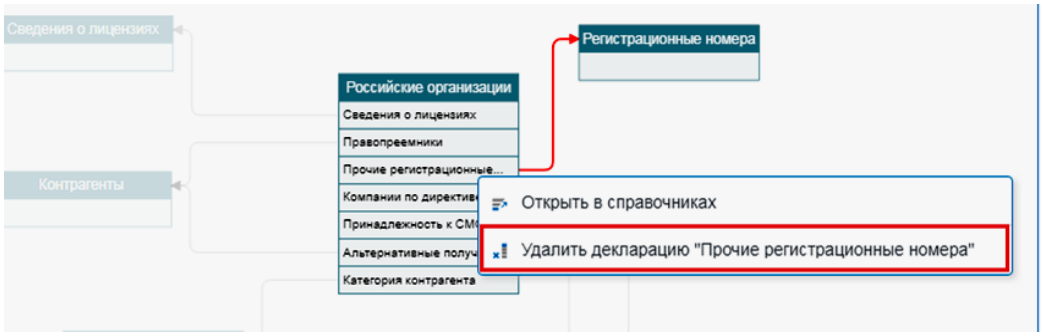


Рисунок 17 - Удаление декларации в окне Взаимосвязи между группами

Первичная расстановка объектов и линий связи на схеме выполняется автоматически. Можно выполнить ручную расстановку групп с нужным выравниванием графических связей и последующим сохранением

расстановки. Для этого нужно навести курсор на справочник или классификатор (появляется четырехнаправленная стрелка), зажать левую кнопку мыши и переместить объект в нужное место. Связи перемещаются и перестраиваются автоматически. Если затем выбрать команду **Авторасстановка**, то *Система* перестроит схему в автоматическом режиме. Кнопка ***Ортогональные линии связи*** изменяет схему таким образом, чтобы линии связи и наследования располагались относительно объектов под прямым углом.

3.4 Поиск в дереве справочных групп. «Худое» дерево

В *Semantic MDM* можно производить быстрый поиск необходимой справочной или классификационной группы. Для этого применяется ячейка быстрого поиска, расположенная над панелью дерева (Рисунок 18).

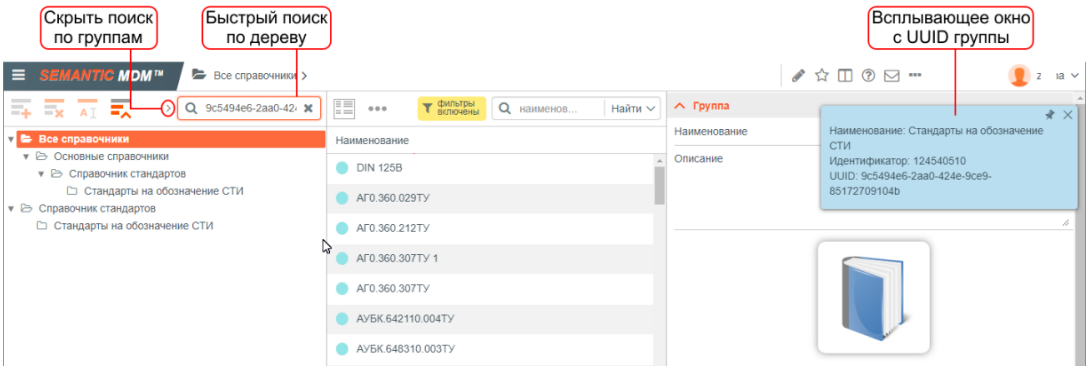


Рисунок 18 - Быстрый поиск по справочным группам

Поиск может производится по наименованию группы, по её уникальному универсальному идентификационному номеру (UUID), который присваивается автоматически любому справочному объекту или по описанию группы.

Для того, чтобы узнать идентификационный номер группы, на неё следует установить курсор в дереве и нажать на клавиатуре сочетание клавиш <*Ctrl*>+<*G*>. Номер появится во всплывающем окне. Кроме этого, идентификатор может быть получен при выгрузке структуры справочника в Excel или XML.

При выполнении быстрого (п. 3.7.4) или расширенного (п. 3.7.5) поиска по позициям дерево групп по умолчанию отображается полностью. При необходимости оно может быть отфильтровано с построением «худого» дерева. Для этого после поиска по позициям необходимо нажать кнопку ***Отфильтровать позициями***, расположенную под деревом групп. После построения "худого" дерева в нем останутся только те, справочные группы, которые содержат попавшие в фильтр позиции, а также их родительские группы (Рисунок 19).

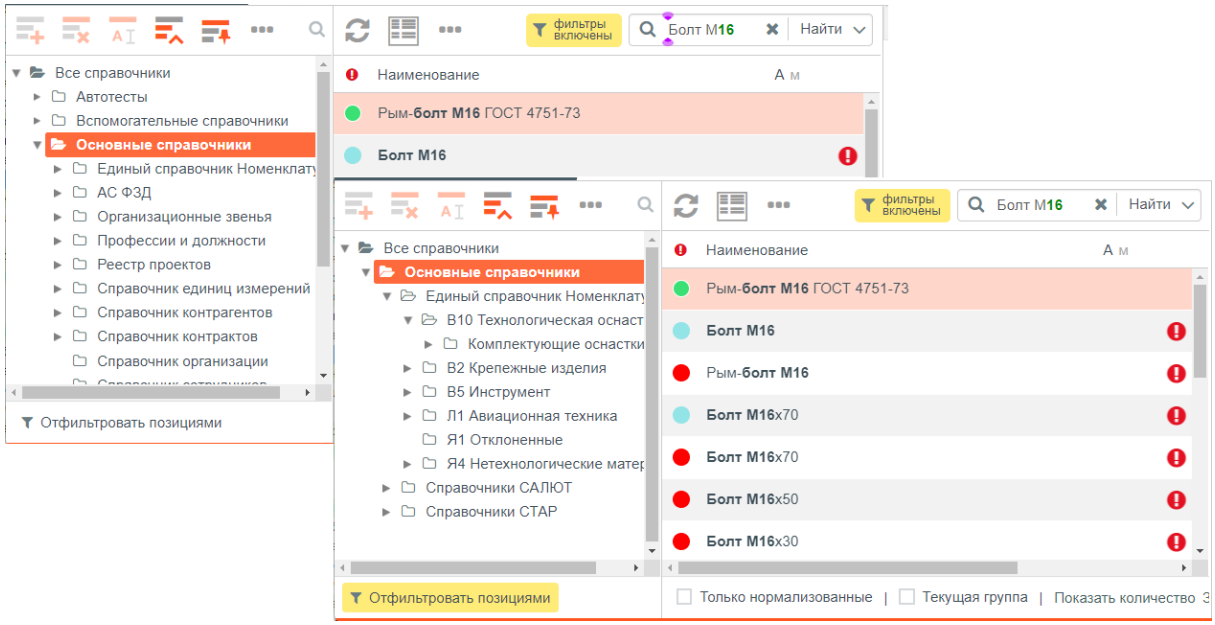


Рисунок 19 - Фильтрация дерева справочных групп по найденным позициям

3.5 Представление данных в таблице

При установке курсора на любой из групп в дереве в таблице отображаются позиции как текущей, так и всех подчинённых групп справочника или классификатора. Для просмотра позиций только выбранной справочной группы под таблицей имеется опция - **Текущая группа**. При установки маркера данной опции в таблице останутся только позиции текущей справочной группы.

По умолчанию эксперт НСИ (пользователь с привилегиями редактирования данных) видит позиции со всеми статусами (см. п. 3.14.20). Для просмотра нормализованных (эталонных) позиций нужно установить

маркер опции **Только нормализованные**, который расположен под таблицей (Рисунок 20).

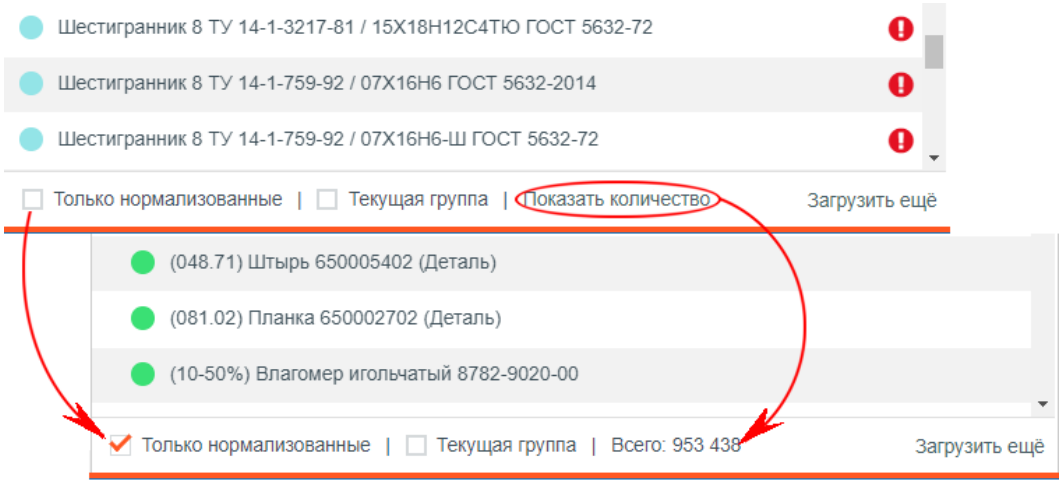


Рисунок 20 - Позиции в таблице

В иерархическом классификаторе над таблицей видны ярлыки вкладок, на которых расположены классифицируемые позиции. Для перехода с одной вкладки на другую достаточно щёлкнуть левой кнопкой мыши на ярлыке.

Для удобства пользователя строки таблицы данных попеременно окрашены разные цвета.

Вращая колесо мыши или перемещая бегунок в правой части списка позиций можно просматривать данные. Если таблица с данными очень широкая и столбцы не умещаются в выделенной области то её можно перемещать зажав левую кнопку мыши до появления курсора в виде .

По умолчанию в табличном представлении позиций отображается только первая страница с данными. Последующие порции данных, загружаются по мере перемещения данных вниз колесом мыши или клавиатурой (PageUp/PageDown), кроме того, данные могут подгружаться нажатием на кнопку ***Загрузить еще***. Максимальное количество однократно подгружаемых позиций равно 300 тысячам.

Для того, чтобы увидеть количество всех объектов в текущей ветке справочника нужно нажать кнопку ***Показать количество*** (Рисунок 20). При

переходе на другую справочную группу, режим отображения страниц сбрасывается. Режим отображения сохраняется отдельно для каждой группы индивидуально под каждого пользователя.

Установив курсор на какой-либо записи в таблице объекта НСИ, можно увидеть её атрибуты, располагающиеся справа от списка в карточке записи[1].

3.5.1 Инструментальная панель таблицы данных

Область таблицы данных имеет свою инструментальную панель, вид которой зависит от ширины сетки данных (Рисунок 21, Рисунок 22).

При минимальном размере таблицы часть кнопок группируются в выпадающий список под кнопкой  .

Размер и местоположение поля быстрого поиска всегда остаются неизменными. Количество доступных кнопок зависит от режима и справочной группы, в которой находится *Система*.

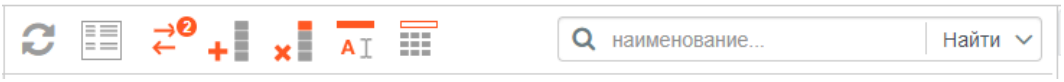


Рисунок 21 - Инструментальная панель таблицы без группировки кнопок

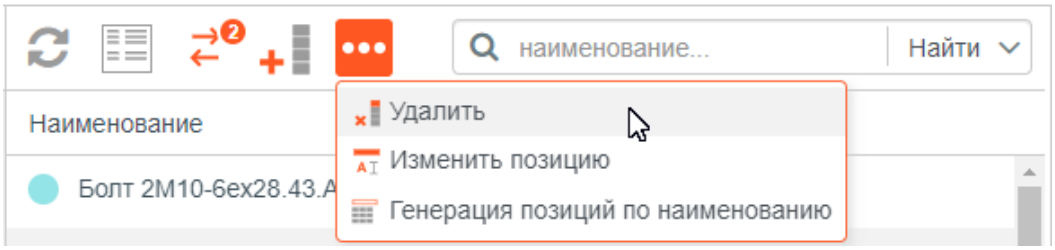


Рисунок 22 - Инструментальная панель таблицы с группировкой кнопок

Описание функций кнопок инструментальной таблицы данных представлено в Таблице 7:

[1] Если карточка справа отсутствует, то это значит, что выбран вид «Дерево и таблица» либо «Таблица». Чтобы увидеть карточку записи следует изменить способ отображения с помощью кнопки **Вид**

Таблица 7. Кнопки инструментальной таблицы данных

Кнопка		Функция
	**Обновить таблицу**	Кнопка обеспечивает обновление записей в таблице. Используется после редактирования данных
	**Заявки**	Открывает доступ к двум командам: **Создать заявку** и **Журнал заявок**. Команда **Создать заявку** позволяет создать заявку непосредственно из текущего объекта НСИ или из текущей записи. Команда **Журнал заявок** открывает вкладку с заявками.
	**Сравнить**	Кнопка открывает доступ к функции сравнения двух и более записей объекта НСИ по значениям их атрибутов. Подробнее см. п. 3.6.4
	**Добавить позицию**	Кнопка доступна только в режиме управления пакетами изменений. Добавляет новую запись к текущей ветке объекта НСИ
	**Удалить**	Кнопка доступна только в режиме управления пакетами изменений. Удаляет запись из текущей ветки объекта НСИ
	**Изменить позицию**	Кнопка доступна только в режиме управления пакетами изменений. Даёт возможность редактирования записи, созданной с использованием ограничительных таблиц.
	**Генерация позиций по наименованию**	Кнопка доступна в режиме управления пакетами изменений только для записей, генерирующихся по ограничительным таблицам. Вызывает Мастер создания записей для добавления новых записей по наименованию объектов

3.5.2 Добавление и удаление колонок на сетке данных

При первом обращении к любому из объектов НСИ в таблице объектов обычно будут присутствовать только колонки, настроенные для отображения по умолчанию, одна из которых, содержащая статусы записей, является системной, а содержание других зависит от объекта НСИ и настроек *Системы*. При необходимости можно увеличить число видимой на сетке информации[2]. Для этого следует:

- навести курсор на строку с заголовками колонок таблицы;
- щелчком правой кнопки мыши вызвать контекстное меню (Рисунок 23);

[2]Настроенные колонки сохраняются для текущего пользователя. При следующем обращении к разделу будет показана настроенная сетка данных

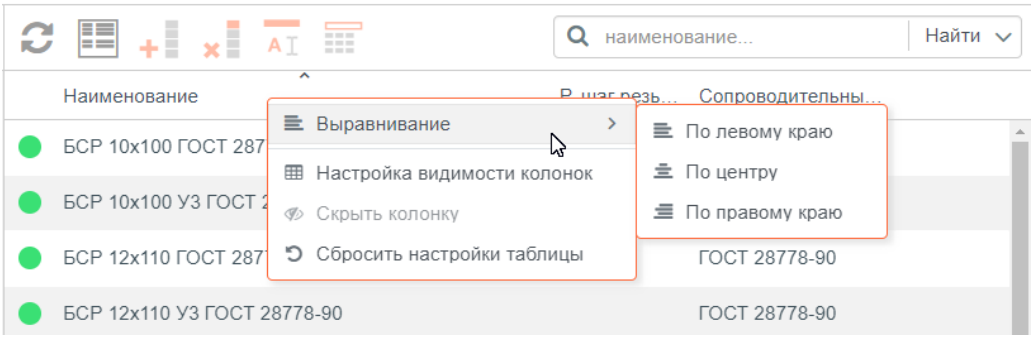


Рисунок 23 - Вызов настройки видимости колонок таблицы

- выполнить команду Настройка видимости колонок;
- в открывшемся окне Настройка видимости колонок поставить маркеры напротив тех атрибутов, которые требуется вывести на сетку данных и нажать кнопку ***ОК*** (Рисунок 24).

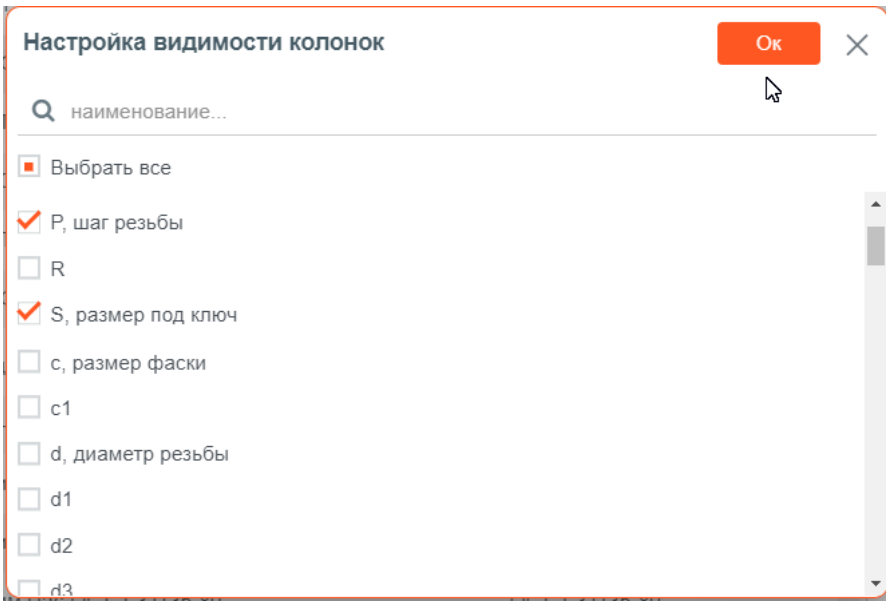


Рисунок 24 - Выбор атрибутов в колонки таблицы

Порядок расположения колонок в таблице может быть изменен. Для этого достаточно зажать левую кнопку мыши на нужном столбце и потянуть его (функция Drag&Drop) на разделительную линию ⋮, которая появляется на невидимой границе между столбцами при перемещении (Рисунок 25). Системный столбец со статусами не может быть перемещен. Если в *Системе* разрешено использование системной колонки (см. раздел 3.31) *Наименование*, то она также не может быть перемещена.

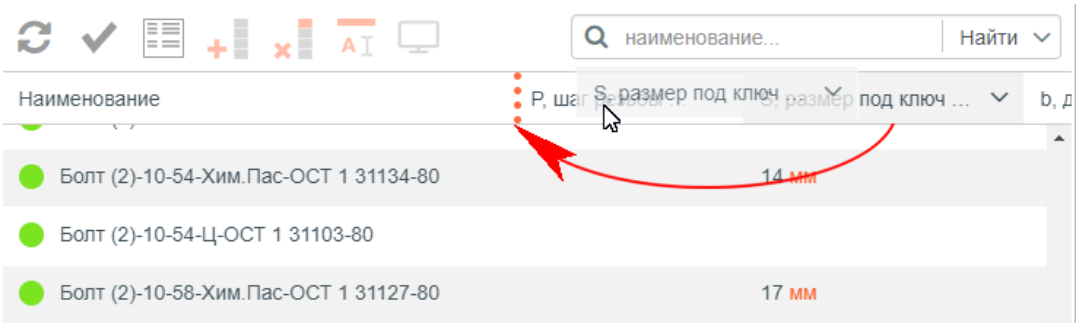


Рисунок 25 - Перемещение столбца на сетке данных

Для того, чтобы скрыть ненужную информацию (колонку), достаточно навести курсор на её заголовок и в контекстном меню выполнить команду **Скрыть колонку** либо вновь выполнить команду **Настройка видимости колонок** и снять маркеры напротив ненужных атрибутов.

Если для атрибута, выведенного в таблицу, настроен постобработчик на изменение (см. п. 3.14.8) с возможностью ручного запуска, то в настройках колонки будет доступен его запуск (Рисунок 26). Команда доступна только в режиме редактирования.

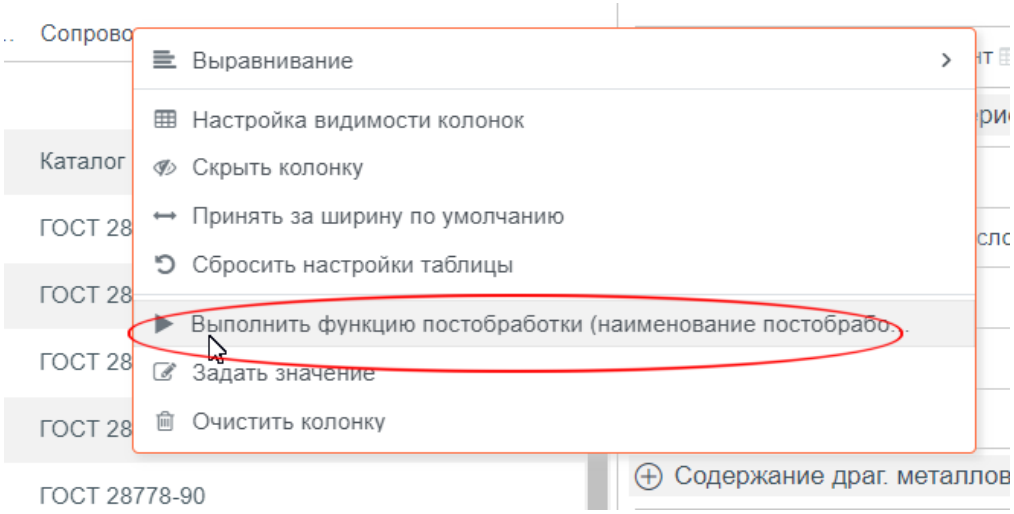


Рисунок 26 - Запуск постобработчика на изменение из таблицы

Чтобы вернуться к настройкам таблицы по умолчанию, нужно выполнить в контекстном меню команду **Сбросить настройки таблицы**. После этого таблица приобретёт вид «по умолчанию».

Команда **Выравнивание**[3] обеспечивает доступ к способам расположения данных в ячейках таблицы (Рисунок 27). Доступны три способа расположения:

- **По левому краю** (используется во всех таблицах по умолчанию);
- **По центру**;
- **По правому краю**.

Команда **Принять за выравнивание по умолчанию** обеспечивает установку принятого способа расположения для всех пользователей таблицы, которые не установят иное выравнивание. Команда доступна только в режиме редактирования (см. раздел 3.13).

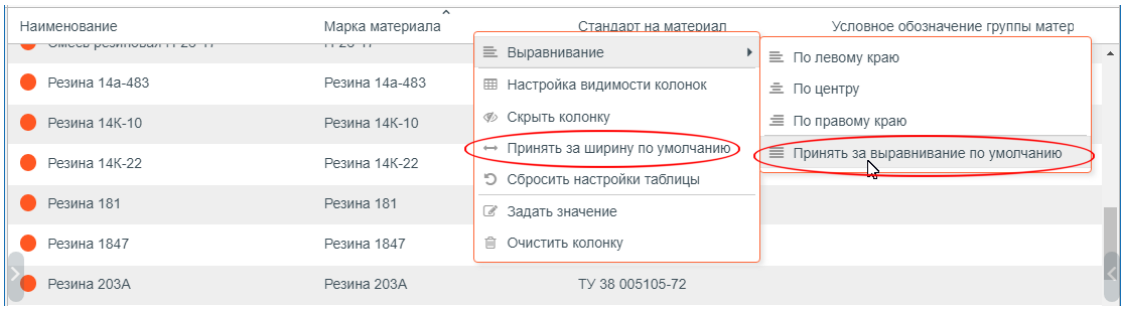


Рисунок 27 - Выравнивание записей в колонках таблицы данных

Ширина любых колонок подбирается автоматически в зависимости от типа данных. Команда **Принять за ширину по умолчанию** позволяет установить текущую ширину колонки таблицы в качестве ширины для всех пользователей, которые не установят иную ширину (Рисунок 27). Команда доступна только в режиме редактирования.

Если столбцов много, и они не умещаются в табличной области, то таблицу можно перемещать влево-вправо, а также вверх-вниз, зажав левую кнопку мыши до появления курсора в виде .

[3]*Команды **Принять за выравнивание по умолчанию** и **Принять за ширину по умолчанию** доступны только для пользователей с административными правами.*

3.6 Работа с информацией в объектах НСИ

Работа с информацией во всех подсистемах и модулях Semantic MDM ведётся с использованием карточек. При установке курсора на наименовании справочной группы в дереве, справа будет открыта карточка группы, содержащая атрибуты, продекларированные для нее (декларации атрибутов). При нахождении в иерархическом классификаторе – карточка классификатора.

Выбор позиции в таблице данных позволяет обратится к карточке позиции, содержащей значения атрибутов, подробно описывающих объект.

3.6.1 Карточка группы

Любой справочный объект имеет собственное описание, располагающееся в правой части рабочего окна и называющееся карточкой объекта. Содержание карточки зависит от того, в каком объекте *Системы* находится пользователь и где в текущий момент располагается курсор. Кроме того, имеют значения права доступа, имеющиеся у текущего пользователя.

На уровне группы (курсор установлен в дереве) в карточке отображаются следующие компоненты (Рисунок 28):

- *Наименование* – наименование справочной группы;
- *Описание* - текстовое описание типа объектов, находящихся в группе (не обязательное свойство);
- *Идентификатор OTD* - идентификатор объекта в соответствии с открытым техническим словарем по ГОСТ Р ИСО 22745-1-2013 «Системы промышленной автоматизации и интеграции»;
- *Связь с метаданными* – область для подключения системного объекта НСИ, который выбирается из выпадающего списка;
- *Постобработчик на создание* - скрипт Groovy, обеспечивающий обогащение и / или проверку атрибутивного состава записи после ее создания в справочной группе;

- *Постобработчик на изменение статуса* - скрипт Groovy, срабатывающий на изменение статуса записи;
- *Изображение* - область присоединения рисунков, отражающих сущность позиций группы. Если изображение не добавлено, то в карточке данный компонент не отображается.

При нажатии на клавиатуре комбинации <Ctrl>+<G> во всплывающем окне отображается GUID (уникальный идентификатор) текущей группы объекта НСИ (Рисунок 28).

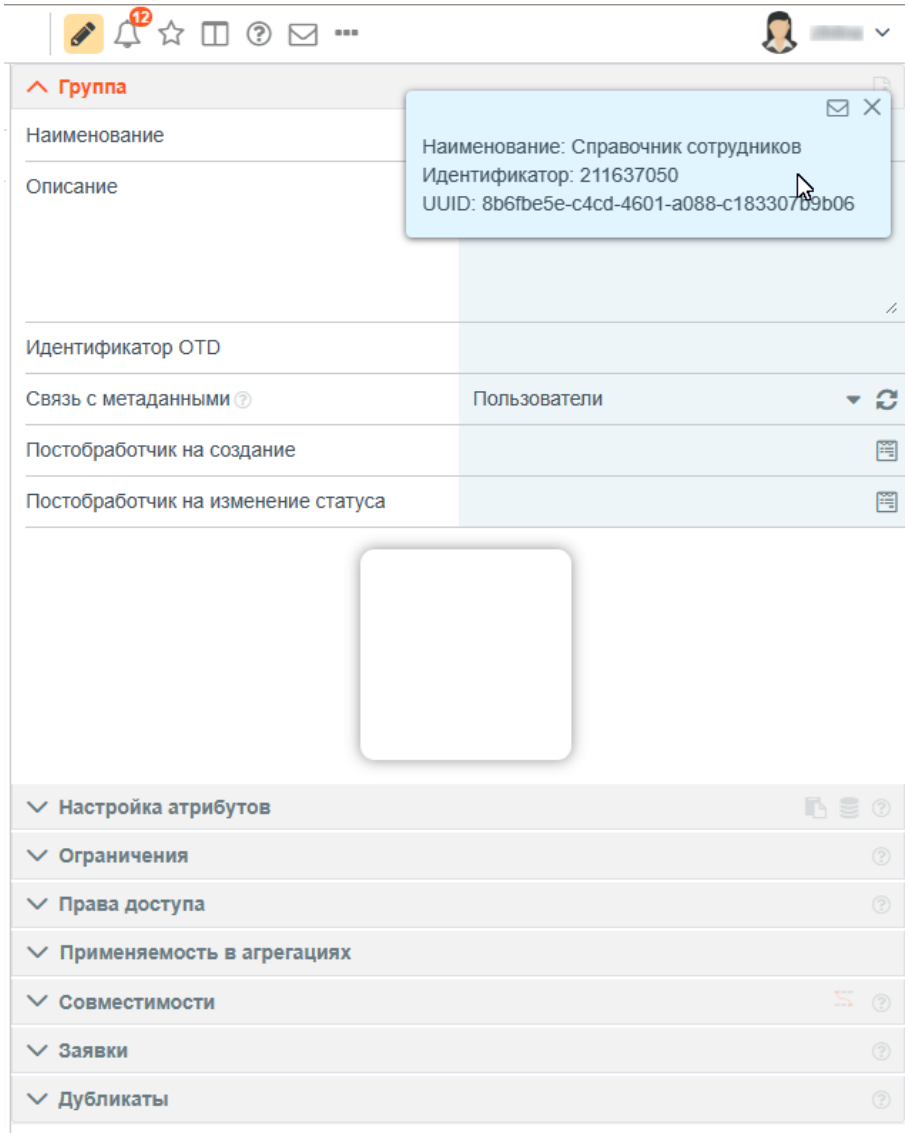


Рисунок 28 - Карточка справочной группы

Карточка группы также содержит несколько разворачиваемых панелей (разделов):

Атрибуты – раздел содержит декларации атрибутов, применяемых для описания позиций данной справочной группы и элементы их настройки. В режиме **Управление пакетами изменений** существует возможность настройки подсветки категории и множественных атрибутов связи. Цвета подсветки по умолчанию назначаются системным администратором. При необходимости цвет подсветки может быть настроен в каждой справочной группе для категории и множественных атрибутов связи в секции **Настройка атрибутов** карточки группы (Рисунок 29);

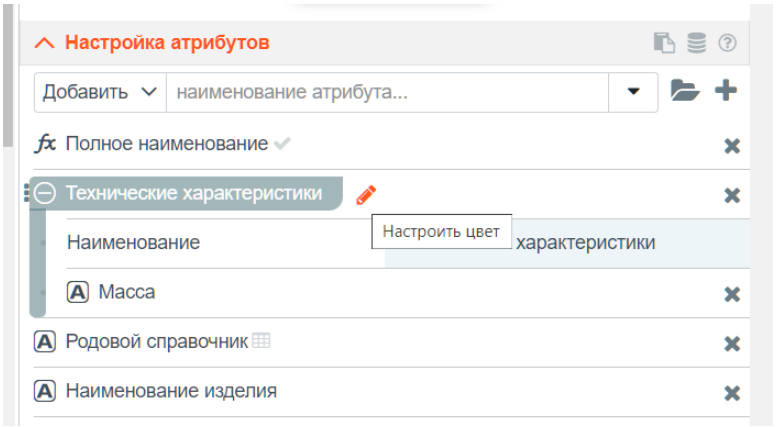


Рисунок 29 - Настройка подсветки категории и множественных атрибутов связи

- **Ограничения** – раздел описывает четыре обязательных элемента, которые определяют особенности создания записей в данной группе:
  - *Ограничительные таблицы* – список ограничительных таблиц, по которым проверяются и создаются записи текущей группы (см. раздел 3.18);
  - *Обязательные атрибуты* – список атрибутов из раздела **Атрибуты**, заполнение которых является обязательным[4];

[4]Если обязательный атрибут не заполнен, то запись не может быть опубликована со статусом ***Нормализована***

- *Уникальные атрибуты* – список атрибутов и сочетаний атрибутов из раздела **Атрибуты**, значения которых будут проверяться при вводе среди всех имеющихся записей в *Системе*, а при присвоении статуса *Нормализована* в пределах группы среди записей с аналогичным статусом;
- *Функции проверки данных* – позволяет создать скрипт Groovy для проверки вводимых данных. Например, проверка ИНН на контрольное число, проверка на наличие в записи латиницы или двойных пробелов;

- **Права доступа** – раздел виден только пользователям, имеющим административные привилегии. В нем можно настраивать особенности работы с выбранным объектом НСИ для различных ролей и отдельных пользователей;
- **Применяемость в агрегациях** – содержит список других объектов или групп, с которыми данная справочная группа находится в агрегации / композиции, а также их атрибуты, значения которых применяются в объектах текущей группы;
- **Совместимости** – в разделе указываются объекты, семантически совместимые с текущим, например, аналоги;
- **Заявки** – в разделе представлены настройки маршрутов согласования при работе с записями группы, а также сами заявки, связанные с изменением, созданием или удалением объекта.
- **Дубликаты** – в разделе возможен поиск дубликатов, а также содержится список ранее найденных дубликатов текущего объекта. Раздел доступен только пользователям с административными правами. Подробнее о работе с дубликатами изложено в разделе 3.15.

В карточке группы можно выполнить быстрый поиск атрибута по имени. Для этого необходимо выполнить следующие действия (Рисунок 30):

- ввести в поле быстрого поиска, расположенного над списком деклараций атрибутов, символы, которые должны содержаться в

наименовании атрибута;

- установить курсор в поле со значением атрибута или в поле свойства декларации;
- сбросить поиск через кнопку ×, после чего система сохранит в поле видимости декларацию атрибута, в которую ранее пользователь установил курсор.

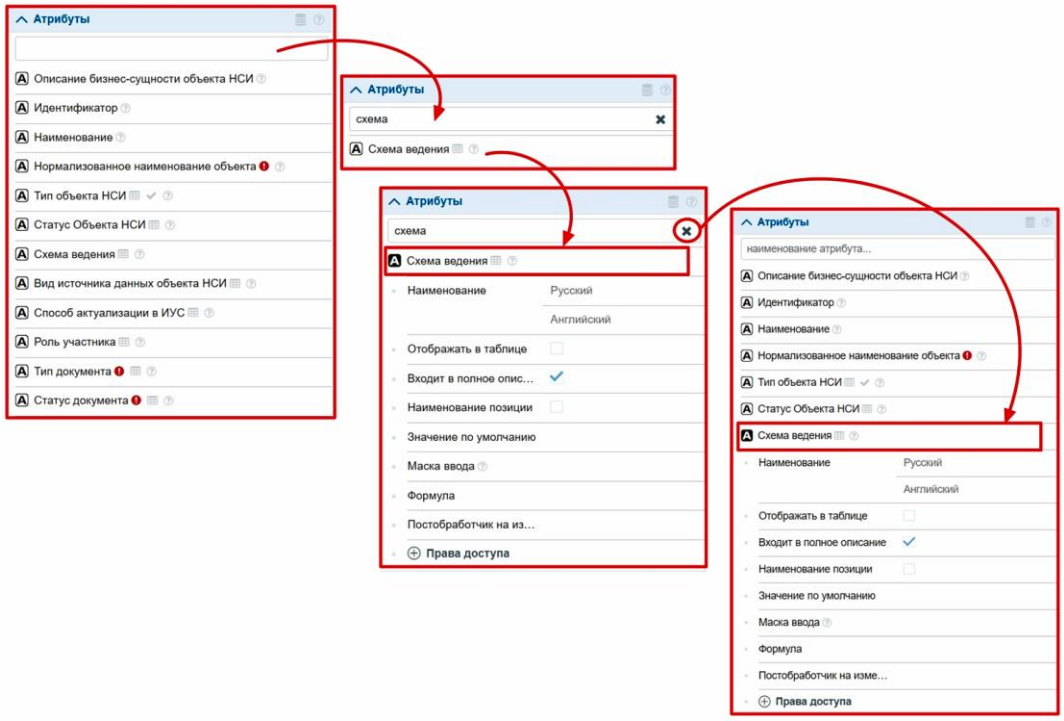


Рисунок 30 - Быстрый поиск атрибутов в карточке группы

3.6.2 Карточка иерархического классификатора

Карточка классификатора содержит те же категории, которые были описаны для карточки группы в п. 3.6.1, кроме категорий **Совместимости**, **Применяемость в агрегациях**, которые используются только в обычных справочных группах. Также в карточке присутствует дополнительный раздел **Связи**, который содержит сведения об объектах текущей группы и связях между ними.

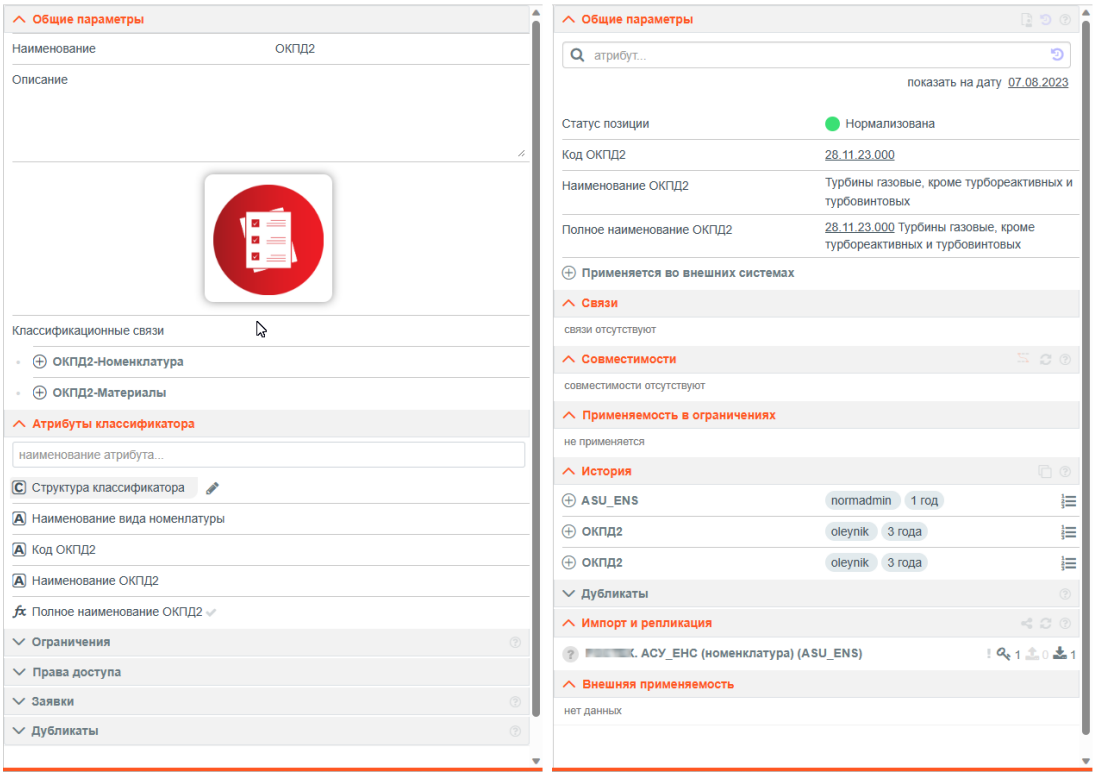


Рисунок 31 - Карточка классификатора и классификационного уровня

Карточка подчиненной группы иерархического классификатора содержит несколько разворачиваемых панелей аналогичных тем, которые были описаны для карточки группы в п. 3.6.1, кроме разделов **Атрибуты, Ограничения, Права доступа, Применяемость в агрегациях**, **Заявки.**

- Кроме перечисленных выше, в карточке подчиненной группы классификатора присутствуют:раздел **Общие параметры**, содержащий сведения о статусе, наименовании, свойствах и других характеристиках группы по аналогии с карточкой записи (см. п. 3.6.3);
- раздел **Связи**, который содержит сведения об объектах, классифицируемых текщей группой и связях между ними;
- раздел **Импорт и репликация,** который содержит список внешних информационных систем (ИС), которые используют данные текущего классификатора. Подробнее о процедурах импорта и репликации см. раздел 3.30;

- раздел **Внешняя применяемость**, содержащий список внешних информационных систем, в которых применяется текущий объект (в зависимости от системных настроек (см. раздел 3.31) раздел может отсутствовать в карточке объекта), а также сведения о нем из этих внешних систем.

3.6.3 Карточка записи

Для объектов на сетке данных (курсор установлен в таблице) карточка содержит полное описание объекта в разворачиваемых панелях, представленных на рисунке ниже:

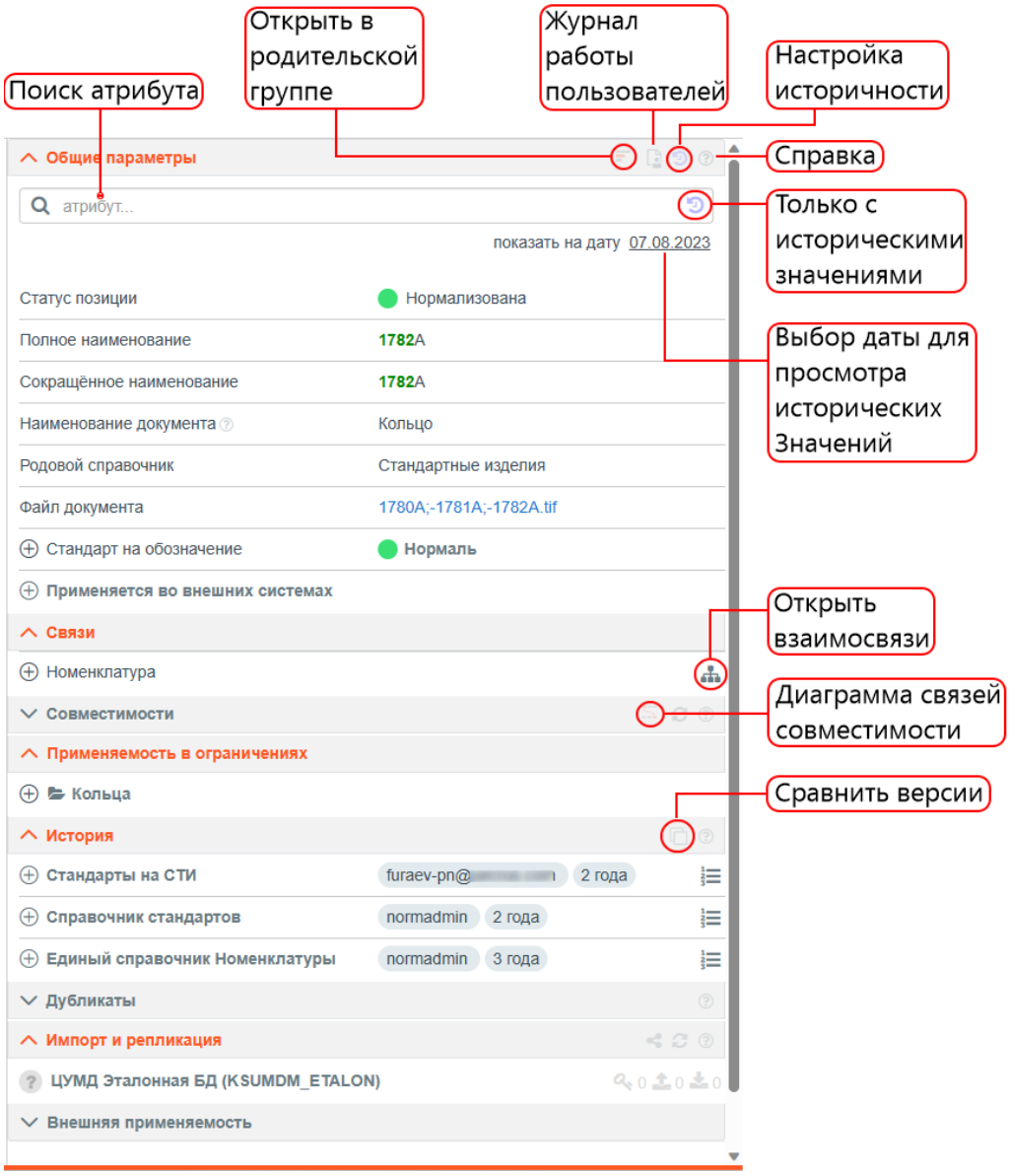


Рисунок 32 - Внешний вид карточки позиции

- **Общие параметры** – здесь располагаются сведения о статусе позиции, наименовании, свойствах и других характеристиках объекта. Если в карточке справочной группы добавлено изображение, то его можно видеть в системном компоненте *Изображение из группы*. Если изображения в группе нет, то данный компонент в карточке позиции отсутствует.

В заголовке панели располагаются четыре пиктограммы - кнопки, которые отвечают за:

- ***Открыть в родительской группе*** – курсор будет споцизионирован в дереве справочников на родительской группе выбранной позиции;
- ***Журнал работы пользователей*** – доступ к журналу работы пользователей с текущей позицией. Кнопка доступна только для пользователей с привилегией на чтение журнала работы пользователей;
- ***Настройка историчности*** – кнопка активна только в режиме работы с пакетами изменений и доступна только пользователям с привилегией *Управление историчностью данных*. включает режим задания исторических значений атрибутов и периода действия позиции;
- ***Справка*** – открывает справку по Системе по соответствующему разделу.

Все сведения о позиции указываются в атрибутах объекта. Перечень доступных атрибутов формируется в разделе **Настройка атрибутов** справочной группы, в которую входит позиция. Атрибуты могут группироваться в категории, которые по умолчании располагаются в карточке в свернутом виде. Категории, а также множественные атрибуты связи имеют подсветку (Рисунок 33), цвет которой настраивается при редактировании / создании справочной группы или может быть задан по умолчанию в системных настройках.

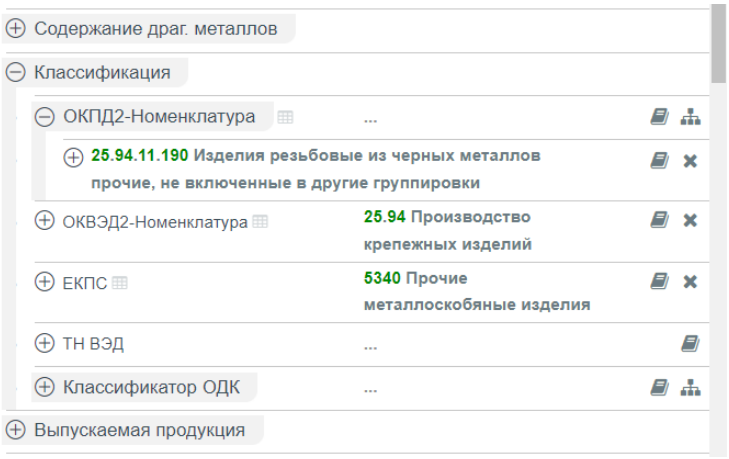


Рисунок 33 - Подсветка категорий и множественных атрибутов связи в карточке позиции

При раскрытии атрибутов вложенной (агрегированной) позиции выполняется затенение атрибутов родительской позиции, что повышает удобство просмотра (Рисунок 34).

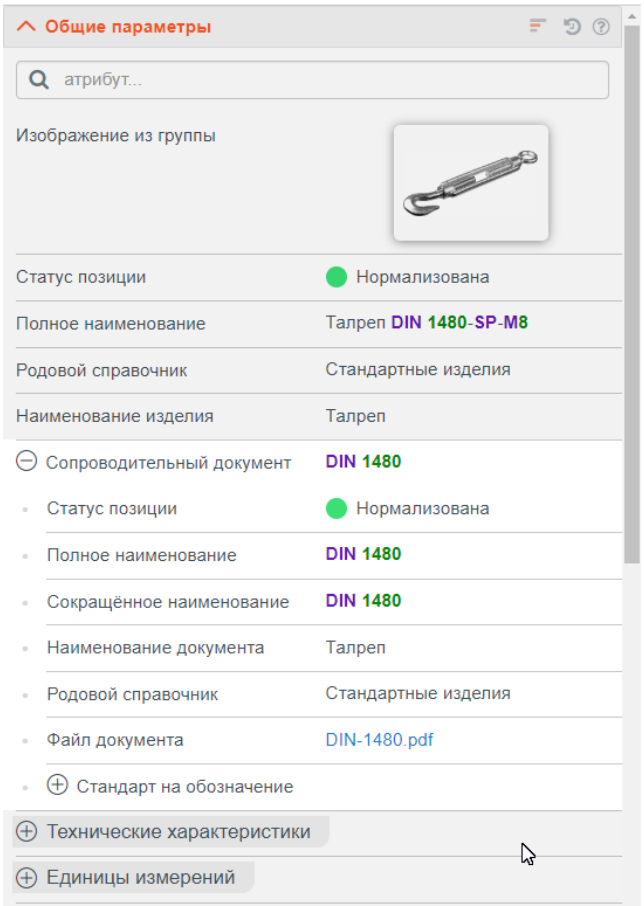


Рисунок 34 - Затенение атрибутов родительской позиции при просмотре значений вложенной

В карточке позиции можно выполнить быстрый поиск атрибута по имени, в том числе с учетом исторических значений. Поле для быстрого поиска расположено над атрибутами. После ввода наименования, система отобразит в карточке позиции только те атрибуты, которые содержат введенную часть наименования (Рисунок 35).

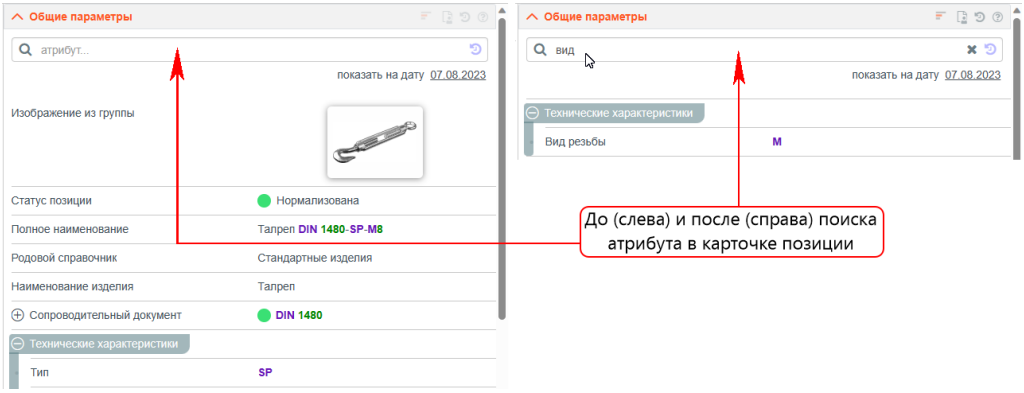


Рисунок 35 - Выполнение поиска атрибута в карточке позиции

В общих параметрах имеется три системных элемента - статус позиции, комментарий к статусу (доступен только в режиме управления пакетами изменений) и раздел **Применяется во внешних системах**. Процедура присвоения статуса описана в п. 3.14.20. Раздел **Применяется во внешних системах** содержит список наименований систем, с которыми имеются переходные ключи (подробнее в разделе 3.30).

- **Связи** – здесь указываются объекты, которые ссылаются через атрибут связи на данный объект;

Если в разделе **Связи** возникает значок  , то это означает, что в списке раздела представлена только часть объектов (обычно не более 250). Для того, чтобы просмотреть все связанные объекты настроить новые связи или удалить существующие необходимо нажать кнопку  ***Открыть взаимосвязи***, после чего будет открыто отдельное окно **Редактор связей**, где можно ознакомится с полным списком связанных позиций. Редактирование данных возможно только в режиме управления пакетами изменений (подробнее см. п. 3.13.1).

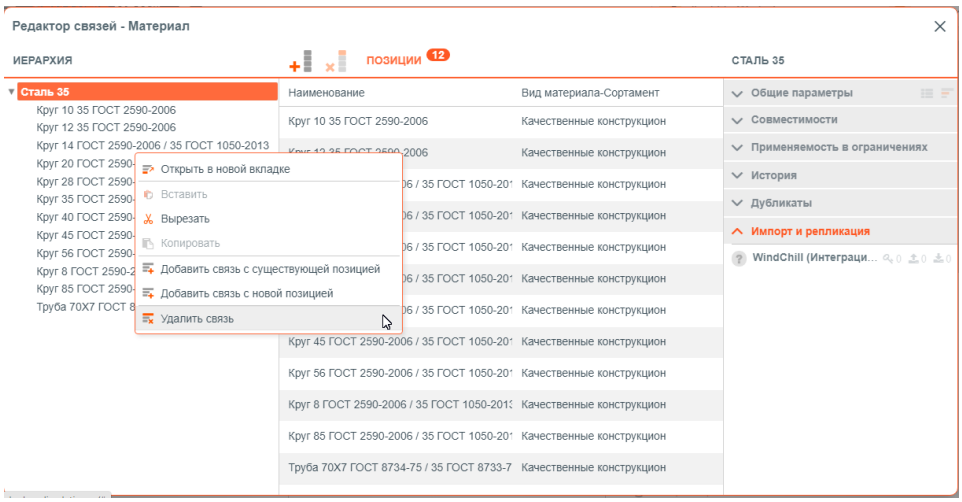


Рисунок 36 - Окно Редактор связей

- **Совместимости** – в разделе присутствуют сведения об объектах справочника семантически совместимых с текущей позицией. Здесь же выполняется настройка совместимости (см. п. 3.13.8);
- **Применяемость в ограничениях** содержит наименование и ссылку на ограничительную таблицу, в которой данный объект используется;
- **История** – журнал событий, содержащий сведения о всех пакетах изменений, применявшихся к текущему объекту, а также доступ к просмотру предыдущих версий объекта через пиктограмму - кнопку ***Сравнить версии*** (см. п. 3.13.11). Раздел видим только пользователям с привилегиями работы с историей;
- **Дубликаты** – здесь возможен поиск дубликатов, а также содержится список ранее найденных дубликатов, текущего объекта. Раздел доступен только пользователям с привилегией *Поиск дубликатов*. Подробнее о работе с дубликатами изложено в разделе 3.15;
- **Импорт и репликация** – содержит список внешних информационных систем в / из которых производится репликация данной позиции и сведения о переходных ключах, сообщениях, которые были отправлены во внешнюю систему, а также сообщениях, на основе которых была сформирована (импортирована) данная позиция;

- **Внешняя применяемость**[5] – список внешних информационных систем, в которых применяется текущий объект (в зависимости от системных настроек раздел может отсутствовать в карточке объекта), а также сведения о нем из них. Например, остатки в складской системе.

Для просмотра исторических данных необходимо щелкнуть левой кнопкой мыши по дате в правом верхнем углу карточки в строке **показать на дату** и, в открывшемся календаре, выбрать (установить) дату, для просмотра позиции (Рисунок 37). Выбранная дата будет подсвечена желтым цветом, а рядом будет показана пиктограмма фильтра. Для возврата к текущему состоянию позиции достаточно однократно щелкнуть по пиктограмме фильтра.

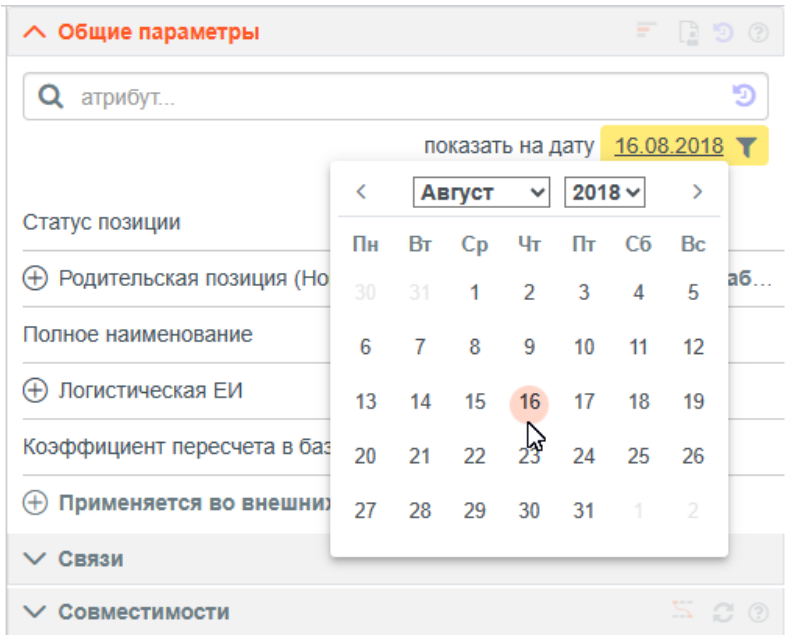


Рисунок 37 - Просмотр исторических значений позиции

3.6.4 Сравнение информации об объектах

Для справочных объектов, располагающихся в таблице данных, доступна функция сравнения.

Чтобы сравнить два или более объектов, необходимо:

[5] Внешняя применяемость объекта доступна, если для объекта НСИ настроен web-сервис, с помощью которого передаётся информация из внешней информационной системы

- найти интересующий объект на сетке данных, используя любые варианты поиска, см. раздел 3.7;
- в строке найденного объекта нажать всплывающую кнопку (появляется при наведении курсора на область кнопки) ***Добавить / Убрать из сравнения*** , которая расположена в правой части строки данных (Рисунок 38);

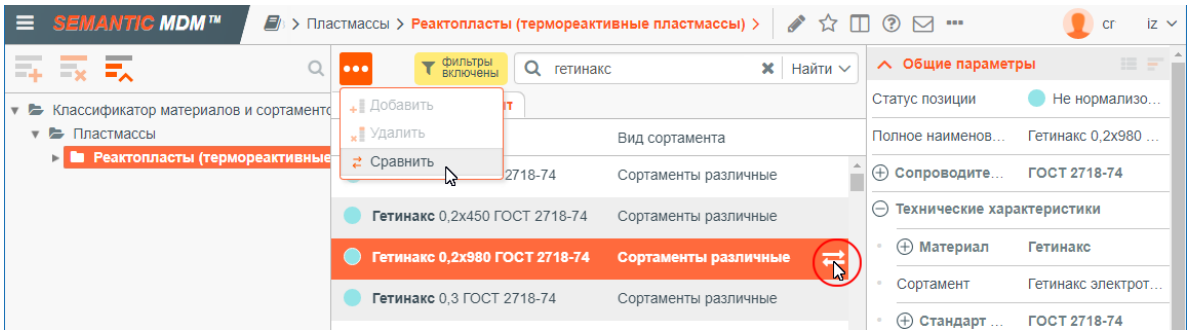


Рисунок 38 - Добавление объекта справочника к сравнению

- повторить добавление для требуемого числа объектов;
- после выбора всех интересующих объектов нажать кнопку ***Сравнить*** (Рисунок 39) для предварительного просмотра списка сравнения (выдающий список под кнопкой ***Сравнить*** инструментальной панели таблицы данных);
- нажать кнопку ***Сравнить*** в окне предварительного просмотра списка сравнения. Расположение кнопки ***Сравнить*** зависит от вида инструментальной панели таблицы.

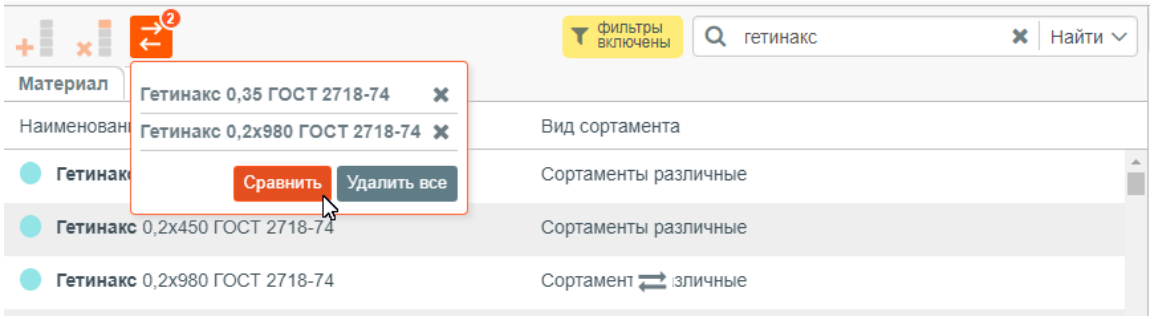


Рисунок 39 - Окно предварительного просмотра списка сравнения

В окне предварительного просмотра списка сравнения можно просмотреть список объектов и удалить некоторые или все объекты, выбранные для сравнения.

Для удаления одного объекта из списка сравнения используется кнопка «×» справа от наименования объекта. Для очистки списка сравнения используется кнопка ***Удалить все***.

В открывшемся окне **Сравнение позиций** (Рисунок 40) будут даны характеристики выбранных объектов. Для ускорения анализа в окне можно установить маркер **только основные**, в этом случае в таблице останутся только основные характеристики объектов. При установке маркера **только различающиеся** в таблице останутся только те атрибуты, значения которых различаются у выбранных объектов. Маркер **только заполненные** позволяет оставить только те атрибуты записей, которые имеют заполненные значения.

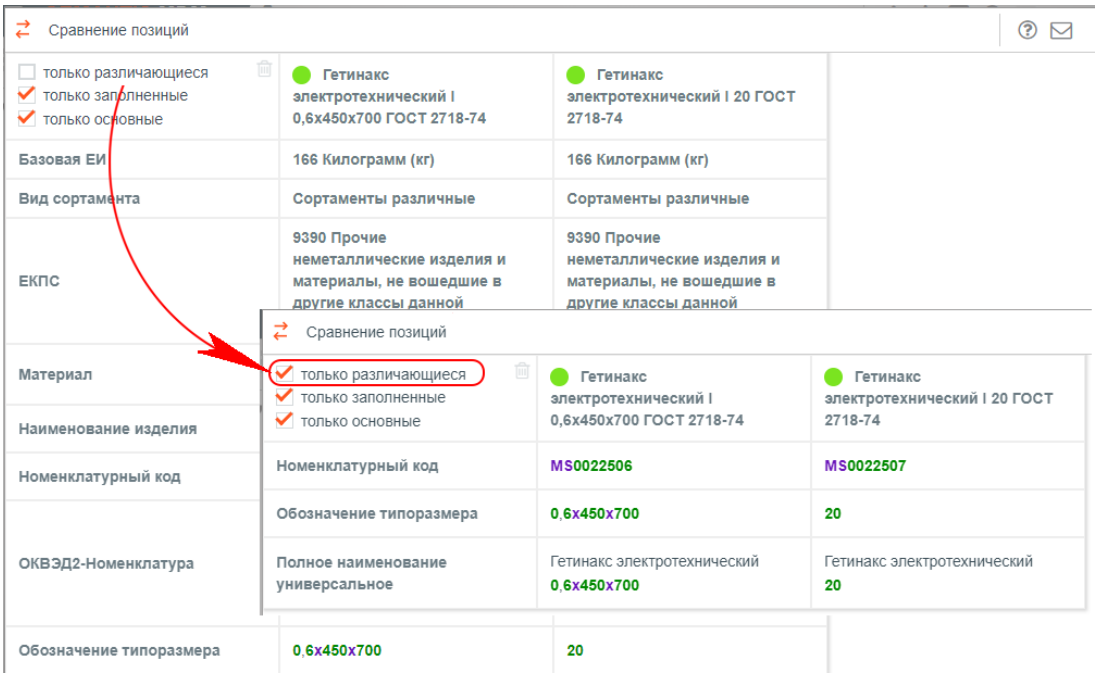


Рисунок 40 - Сравнение записей (объектов НСИ)

Для удаления объектов из списка сравнения в окне **Сравнение позиций** применяются кнопки , расположенные в правых верхних углах столбцов таблицы в списке сравнения. Кнопки для удаления отдельных объектов являются всплывающими, т.е. становятся видимыми при наведении на них

курсора мыши. Чтобы удалить все объекты из списка сравнения нужно нажать кнопку ***Удалить список сравнения***, располагающуюся в первом столбце таблицы в окне **Сравнение позиций**. После нажатия этой кнопки список сравнения очищается и окно **Сравнение позиций** закрывается.

3.6.5 Полное описание справочного объекта

Для каждого справочного объекта, расположенного на сетке данных, возможен просмотр полного описания. Для этого следует установить курсор на интересующем объекте и вызвать из контекстного меню команду **Полное описание позиции**. Аналогичная команда находится в меню кнопки ***Вид***. После чего в нижней части рабочего окна под таблицей с данными появится область с полным описанием выбранной записи (Рисунок 41).

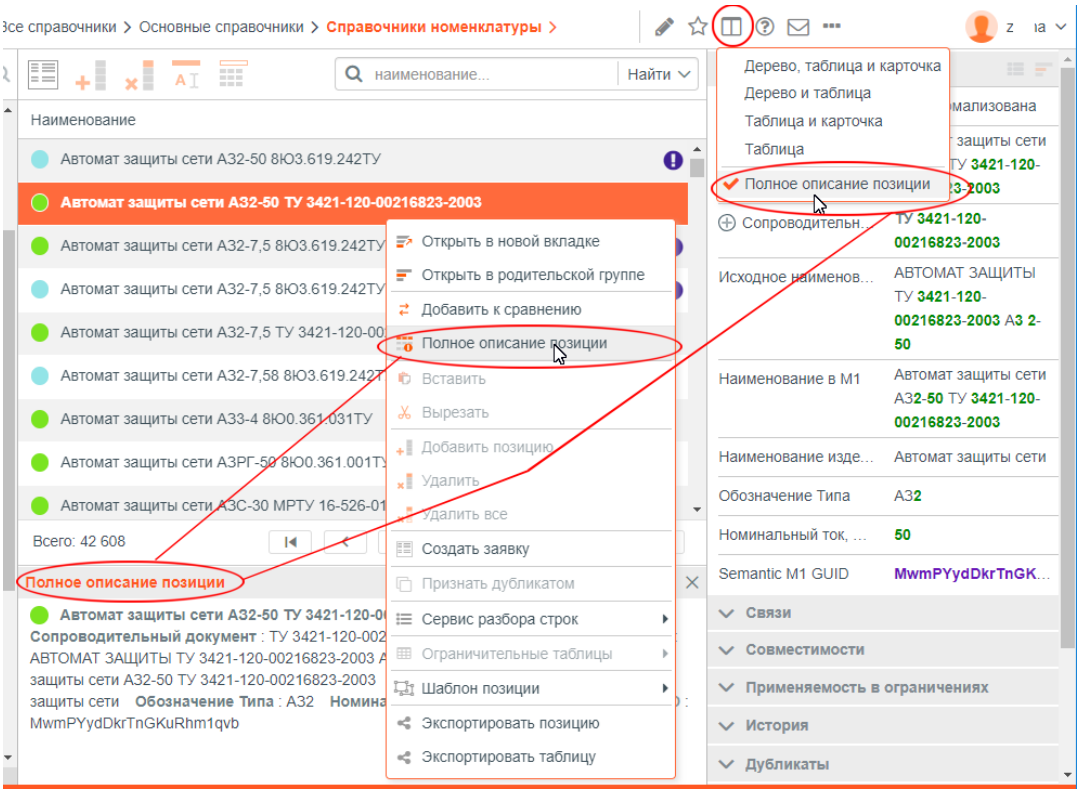


Рисунок 41 - Просмотр полного описания записи

Чтобы закрыть полное описание, достаточно нажать на кнопку в правом верхнем углу области с описанием позиции.

3.6.6 Открытие документов ассоциированными приложениями

Система предоставляет возможность подключать файлы любого типа к справочным объектам. Для этого атрибуту, в котором будет храниться файл, необходимо назначить тип данных (см. п. 3.16.4) *Файл*. Настройку атрибутов выполняет *Эксперт НСИ*.

В атрибуте с типом данных *Файл* в качестве значения может быть использован файл любого типа. Прикреплённый к объекту файл можно открыть в ассоциированном приложении, кликнув на ссылку файла в таблице данных или в карточке (Рисунок 42). Если с данным типом файла не ассоциировано ни одно приложение на ПК пользователя, то файл будет автоматически загружен в локальную папку (настроенную на ПК пользователя для всех загружаемых документов) для дальнейшей работы с ним[6].

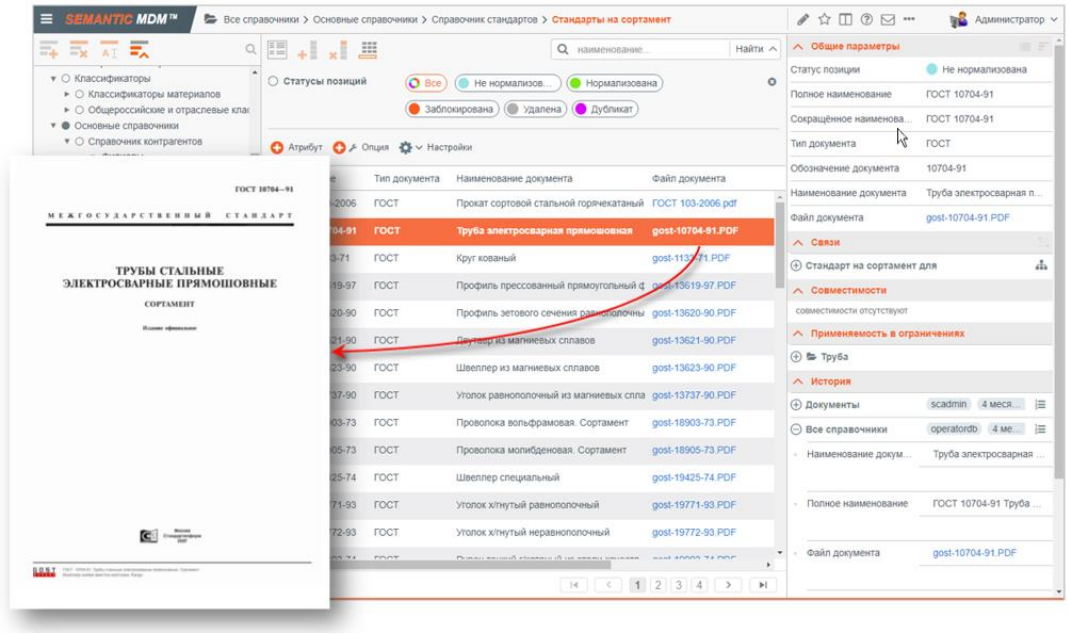


Рисунок 42 - Открытие файла документа ассоциированным приложением

[6]В Системе реализовано версионное хранение файлов существующих записей. Загрузка нового файла или его удаление не приводит к удалению предыдущей версии файла. Удаление позиции приводит к удалению всех файлов, на которые она когда-либо ссылалась, если на эти файлы кроме нее более никто не ссылается

3.7 Поиск данных

В больших информационных массивах процедуры быстрого поиска требуемых данных приобретают особую актуальность. *Система* позволяет с высокой скоростью осуществлять поиск данных в массиве справочной информации. Поиск может производиться по всей базе данных, по определённому объекту НСИ или в пределах указанной группы. В качестве критериев поиска используются значения атрибутов записей.

Система предоставляет пользователям различные способы поиска требуемых данных. Применение того или иного способа зависит от характера и количества информации об искомом объекте.

Для выполнения поиска доступны:

- поиск по всей базе данных. Выполняется со стартовой страницы *Системы* (см. п. 3.7.1);
- сортировка записей на сетке данных (см. п. 3.7.2);
- фасетный поиск с использованием выбранных значений атрибутов записей (см. п. 3.7.3);
- быстрый поиск. Проводится в пределах текущей справочной группы (см. п. 3.7.4);
- расширенный поиск по выбранным критериям (атрибутам) и их значениям (см. п. 3.7.5);
- поиск с учетом исторических значений.

3.7.1 Поиск по всем справочникам *Системы*

На стартовой странице представлена строка для быстрого поиска по всем справочникам (Рисунок 43). Поиск осуществляется по всем атрибутам объектов, присутствующим в объектах НСИ. Результаты поиска выдаются по мере обнаружения записей, содержащих запрошенные данные.

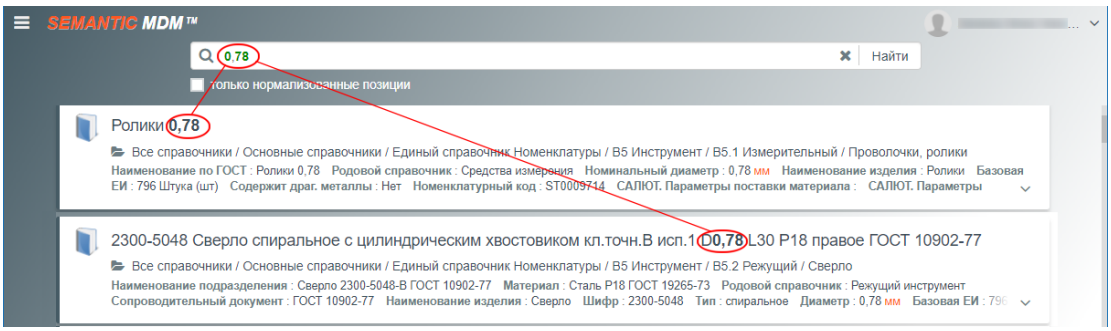


Рисунок 43 - Результаты поиска по всем справочникам

3.7.2 Сортировка записей на сетке данных

Сортировка данных применяется как простейшая процедура, облегчающая поиск требуемых данных в списке объектов НСИ. Для сортировки необходимо щёлкнуть левой кнопкой мыши по заголовку поля таблицы, где следует отсортировать данные. В результате все записи таблицы будут отсортированы по возрастанию значений в выбранном поле. Повторный щелчок мыши на заголовке приводит к сортировке значений по убыванию.

Для столбцов с типом данных «Дата» (см. п. 3.16.4) при включении сортировки, сначала выполняется сортировка по убыванию, а затем по возрастанию.

Для сортировки по нескольким колонкам необходимо держать нажатой клавишу <Ctrl> совместно со щелчком левой кнопкой мыши по заголовку таблицы.

3.7.3 Использование фасетного поиска

Для отбора значений в таблице данных по некоторым заранее определённым условиям можно воспользоваться функцией фасетного поиска в таблице данных.

Для вызова выпадающего меню фасетного поиска следует щёлкнуть мышью на значке «» рядом с заголовком того столбца таблицы, в котором располагается требуемая характеристика объекта (Рисунок 44).

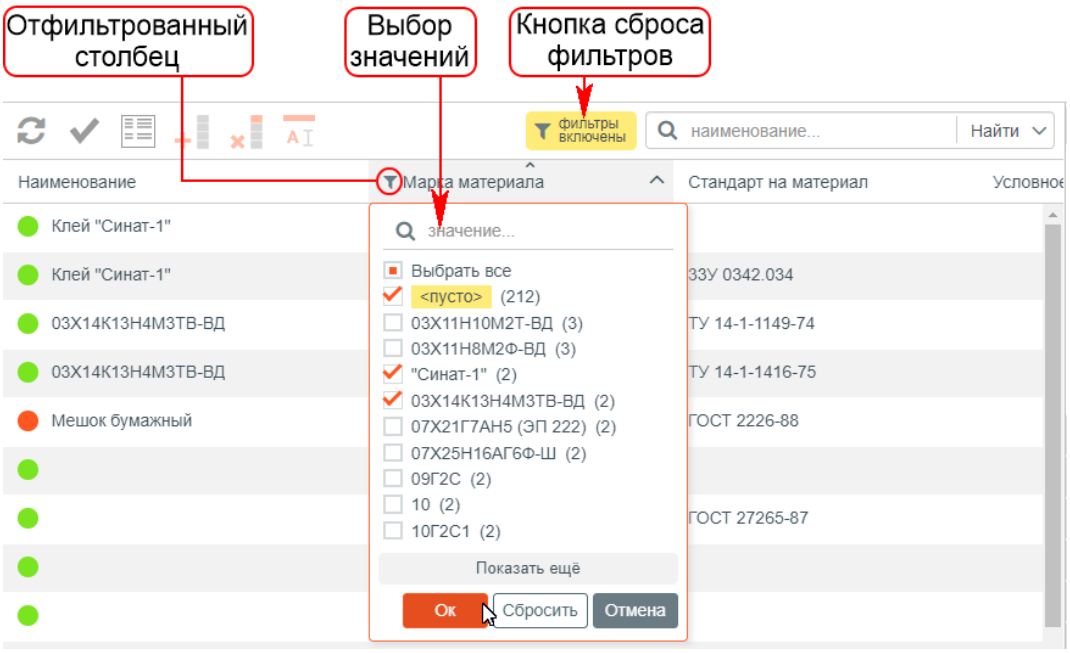


Рисунок 44 - Фасетный поиск данных по значениям

В выпадающем меню рядом с каждым значением в круглых скобках расположено число записей, которые ему соответствуют. Чтобы выбрать значение, нужно установить соответствующие маркеры (галочки) и затем нажать кнопку ***ОК***.

Значение <пусто> позволит включить в выбор те записи, у которых значение в фильтруемом столбце отсутствует.

По умолчанию в списке фасетов присутствует только 10 первых значений. Для загрузки всех возможных значений нужно нажать кнопку ***Показать еще***, расположенную в конце списка.

Если значение, попавшее в фасетный поиск, не умещается в окне полностью (больше ширины окна), то при наведении на него курсора оно отображается во всплывающем окне (hint).

Фасетный поиск может быть последовательно произведён по любому числу столбцов таблицы данных[7]. Результатом будут являться те записи,

[7]Фильтрация не работает в системном столбце «Наименование». Необходимо воспользоваться сортировкой.

которые соответствуют всем отмеченным значениям. Столбцы, по которым выполнен поиск, будут отмечены специальным значком .

Чтобы изменить или отменить результаты проведённого поиска по одному из столбцов, нужно повторно нажать на значок «» рядом с заголовком столбца и снять либо установить маркеры напротив нужных значений. Для снятия всех установленных маркеров следует нажать кнопку ***Сбросить***.

Для того, чтобы вернуть таблицу к первоначальному состоянию, необходимо нажать желтую кнопку ***Фильтры включены***, расположенную над таблицей данных рядом с полем быстрого поиска.

Фасетный поиск может проводиться совместно с другими видами поиска. Например, записи могут быть предварительно отфильтрованы быстрым поиском, а затем конкретизированы при помощи фасетного.

Фильтрация ведётся только в рамках выделенной группы. В случае изменения группы фасетные фильтры отключаются. Для сохранения фильтрации при переключении между группами требуется использовать расширенный поиск (см. п. 3.7.5).

3.7.4 Быстрый поиск

Самым оперативным способом поиска объектов НСИ является быстрый (нечёткий) поиск. Для его выполнения необходимо в поле быстрого поиска набрать искомую строку[8]. Поиск осуществляется в полнотекстовом режиме.

Для наиболее быстрого поиска следует указывать в любом порядке ключевые слова, например, марку, номер ГОСТа, типоразмер и другие возможные значения атрибутов объекта. Для всех позиций доступен поиск по уникальному идентификатору (GUID) Semantic MDM, а также по переходному ключу.

[8]При выполнении поиска буквы «е» и «ё» учитываются как одинаковые

Результат поиска будет отображаться в списке позиций НСИ, в таблице данных скроются объекты, не удовлетворяющие условию поиска (Рисунок 45). Для отображения в дереве только тех справочных группы, Которые содержат в себе результаты поиска («худое» дерево) следует воспользоваться кнопкой Отфильтровать позициями, расположенной в нижней части панели дерева.

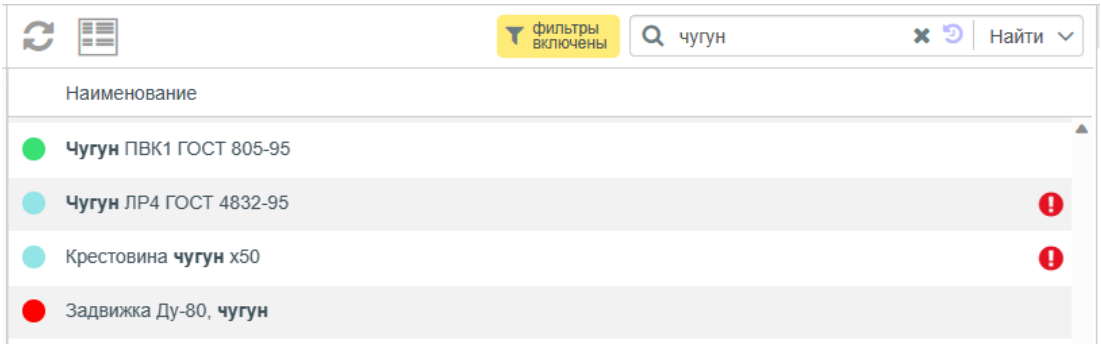


Рисунок 45 - Результаты быстрого поиска

Построение «худого дерева» справочных групп после выполнения быстрого или расширенного поиска (см. п. 3.7.5) выполняется по отдельной кнопке **Отфильтровать по найденным позициями**, которая расположена под деревом[9].

Для того, чтобы выйти из режима показа результатов поиска, необходимо нажать кнопку **Фильтры включены** для их выключения.

Строка запроса для быстрого поиска может использовать специальные операторы уточнения поиска (Рисунок 46). Чтобы просмотреть список операторов достаточно нажать на кнопку в строке быстрого поиска.

[9] Результаты поиска в виде «худого дерева» будут отображаться только в том случае, если администратором *Системы* выполнена соответствующая настройка. В противном случае, поиск будет осуществляться только в пределах выбранной справочной ветки, а результаты поиска будут показаны только на сетке данных

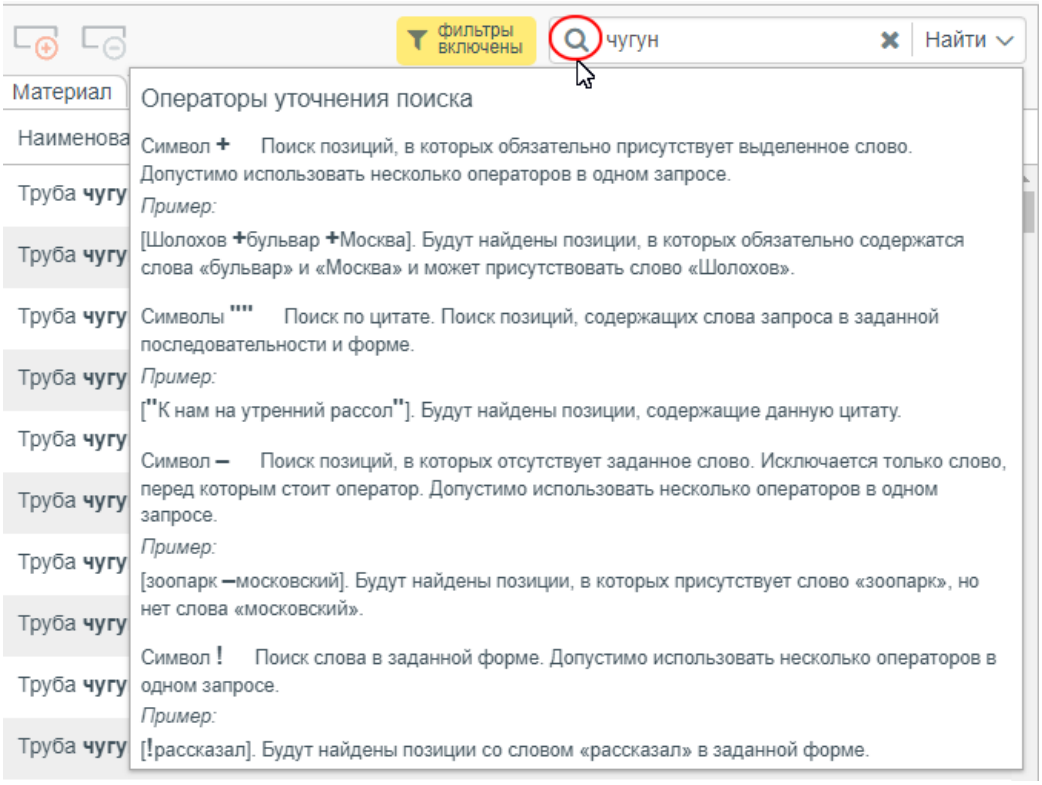


Рисунок 46 - Операторы уточнения поиска

3.7.5 Расширенный поиск

Если необходимо искать по значениям атрибутов объектов, документов или файлов документов, то следует воспользоваться расширенным поиском. Для запуска режима настройки расширенного поиска следует щёлкнуть мышью на значке «˅» кнопки ***Найти***, расположенной рядом с полем быстрого поиска (Рисунок 47).

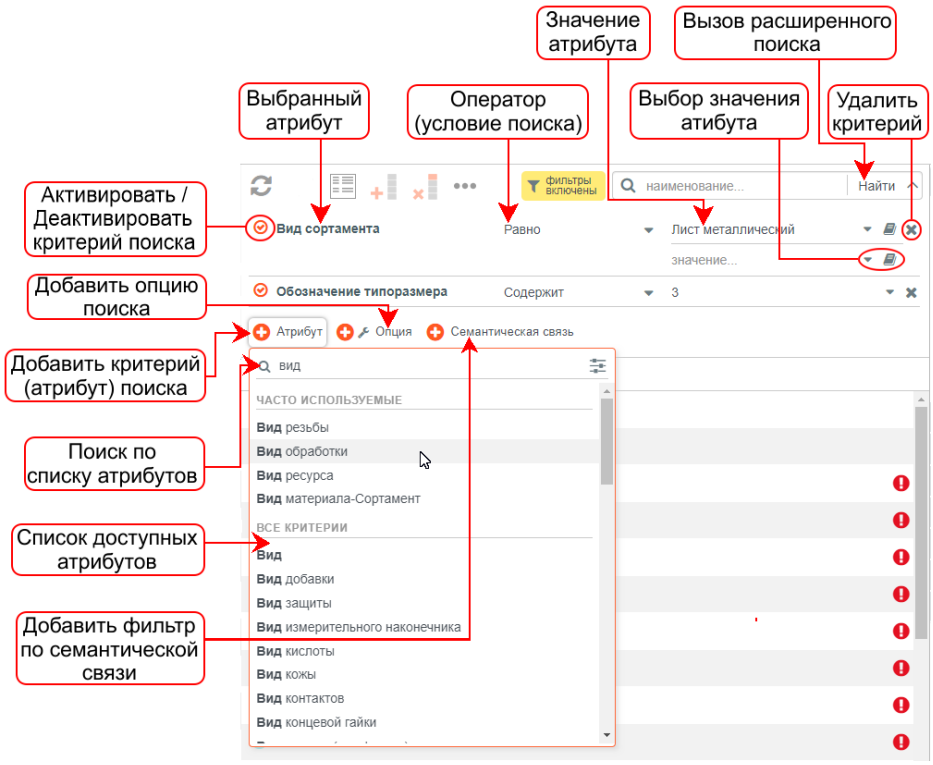


Рисунок 47 - Выбор критериев расширенного поиска

Затем нужно нажать кнопку ***Атрибут*** и в открывшемся списке атрибутов выбрать необходимый[10]. Повторить выбор критериев требуемое количество раз. Выбранные критерии будут записаны над таблицей данных.

В списке критериев поиска в верхней части расположены пять наиболее часто используемых критериев для быстрого доступа к ним.

При выборе критериев поиска у пользователя есть возможность их поиска по наименованию. Для этого в верхней части списка параметров расположена ячейка для ввода названия атрибута. Поиск производится по мере ввода символов. В строке поиска также расположена кнопка ***Настроить критерии*** (Рисунок 48), позволяющая запустить выбор и настройку списка критериев, которые постоянно используются для поиска в текущем объекте НСИ.

[10]*Для выбора доступны только те критерии, которые продекларированы для объектов в текущем объекте НСИ. Если какой-либо атрибут в объекте НСИ не имеет декларации, то он не может выступать в качестве критерия поиска*

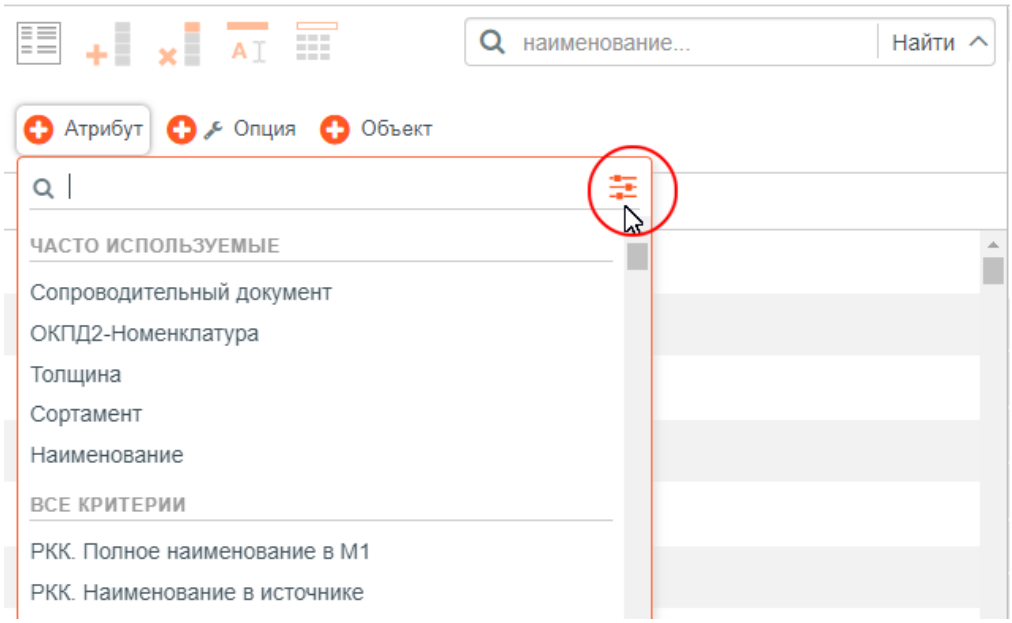


Рисунок 48 - Кнопка **Настроить критерии** для расширенного поиска

В окне **Критерии поиска** будут представлены все атрибуты, используемые для описания объектов текущего объекта НСИ (Рисунок 49). Для добавления атрибута в качестве критерия поиска необходимо найти его в списке слева (можно воспользоваться ячейкой поиска над ним) и перетащить в правую часть окна (Drag&Drop) в область «*перетащите атрибут*». При необходимости здесь же можно настроить условия поиска, а также возможные значения. Критерии поиска, настроенные подобным образом, будут автоматически показываться при вызове расширенного поиска в данном объекте НСИ.

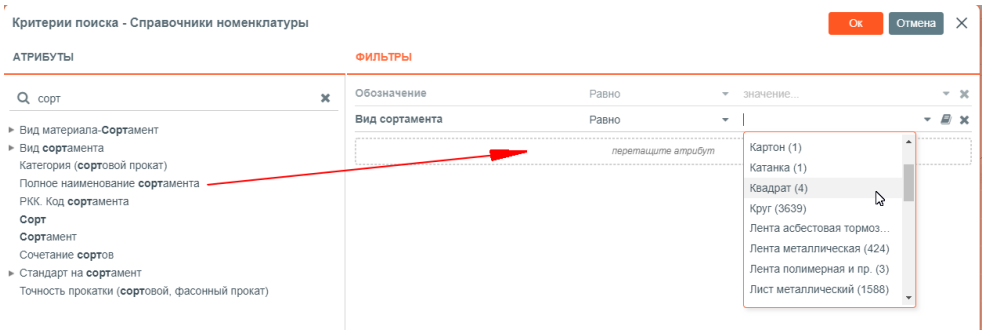


Рисунок 49 - Окно **Критерии поиска** для настройки постоянных критериев поиска в текущем объекте НСИ

Для каждого выбранного критерия поиска нужно установить оператор (условие поиска). Для этого следует щёлкнуть мышью в поле выбора условия и из выпадающего списка выбрать необходимое условие (Рисунок 50).

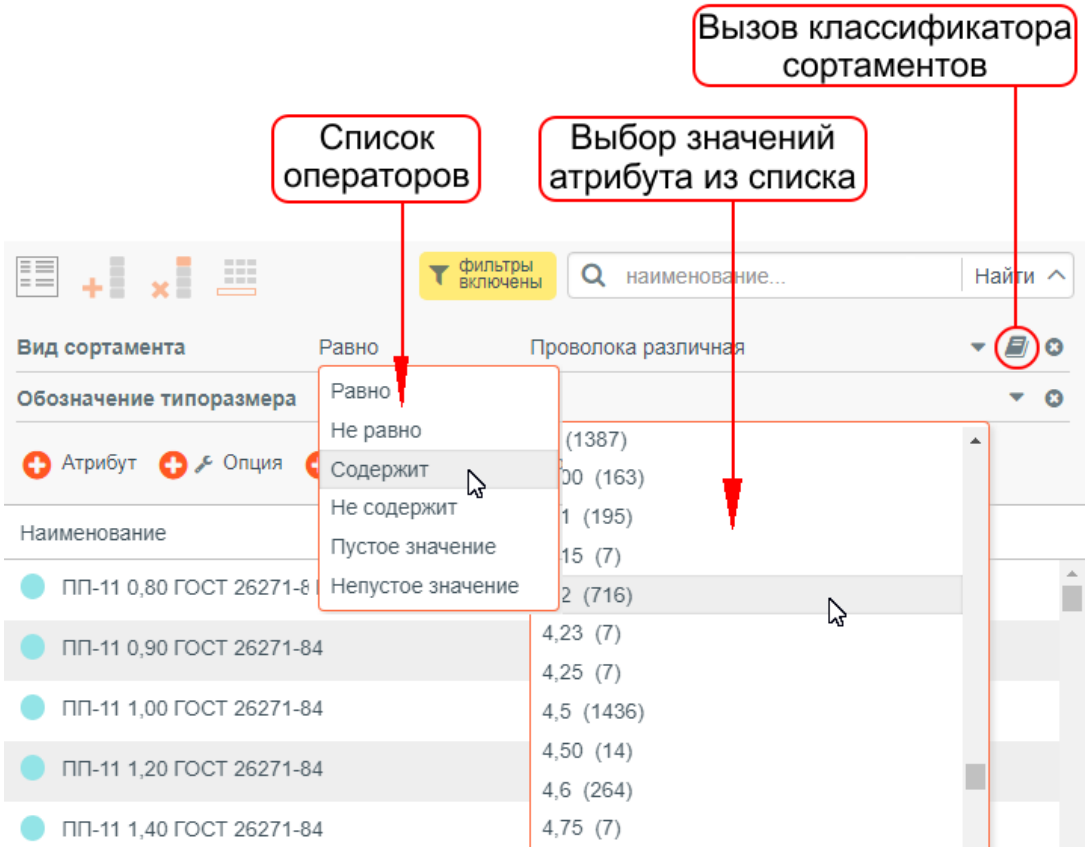


Рисунок 50 - Выбор операторов и значений атрибутов для расширенного поиска

Для числовых атрибутов применяются условие, представленные в Таблице 8.

Таблица 8. Операторы числовых атрибутов

Оператор	Комментарий
Между	Значение должно находиться в указанном диапазоне (a – b). Чтобы установить границы нужно нажать на <Пробел> после ввода первого значения диапазона. Допускается ввод нескольких диапазонов
Равно	Полное соответствие атрибута искомому значению, указанному в поле значений
Не равно	Искомые значения атрибута не должны быть равны значению, указанному в поле значений
Больше	Искомые значения атрибута должны быть больше, чем значение, указанное в поле значений
Больше или равно	Искомые значения атрибута должны быть не меньше, чем значение, указанное в поле значений

Меньше	Искомые значения атрибута должны быть меньше, чем значение, указанное в поле значений
Меньше или равно	Искомые значения атрибута должны быть не больше, чем искомое значение, указанное в поле значений
Пустое значение	Будут отфильтрованы все записи, для которых не заполнено значение атрибута
Непустое значение	Будут показаны все записи, для которых заполнено (непустое) значение искомого атрибута

Для числовых атрибутов поиск выполняется по точному представлению числа без учета единицы измерения, то есть, если введено значение 1000 мм, то будут найдены как записи со значением атрибута 1000 мм, так и записи со значением 1 м. При этом поиск учитывает тип числа (см. п. 3.16.4): единичное (номинальное) значение, диапазон или число с отклонениями.

Для текстовых (строковых) атрибутов используются сравнения, представленные в 9.

Таблица 9. Операторы текстовых атрибутов

Оператор	Комментарий
Равно	Полное соответствие атрибута искомому значению, указанному в поле значений
Не равно	Искомые значения атрибута должны начинаться с символов, указанных в поле значений
Заканчивается на	Искомые значения атрибута должны заканчиваться символами, указанными в поле значений
Содержит	Искомые значения атрибута должны содержать в своём составе символы (цифры, буквы, слова), указанные в поле значений. Возможен ввод списка значений. Поиск по списку значений будет выполняться через «ИЛИ».
Не содержит	Искомые значения атрибута не должны содержать символов (цифр, букв, слов), указанных в поле значений. Возможен ввод списка значений. Поиск по списку значений будет выполняться через «И»
Пустое значение	Будут отфильтрованы все записи, для которых не заполнено значение атрибута
Непустое значение	Будут показаны все записи, для которых заполнено (непустое) значение искомого атрибута
По шаблону	В поисковом запросе в качестве значения могут быть использованы подстановочные знаки “?” (заменяет любой один символ) и “*” (заменяет любое количество символов)
Длина значения больше	Будут показаны все позиции, значение длины (по количеству символов) строкового атрибута которых превышает указанное значение

Длина значения меньше	Будут показаны все позиции, значение длины (по количеству символов) строкового атрибута которых меньше указанного значения
Длина значения равна	Будут показаны все позиции, значение длины (по количеству символов) строкового атрибута которых соответствует указанному значению

Для строковых и числовых атрибутов с типом «множественный» дополнительно доступны условия поиска, представленные в Таблица 10.

Таблица 10. Операторы строковых и числовых атрибутов с типом «множественный»

Оператор	Комментарий
Кол-во значений меньше	Будут показаны те позиции, у которых количество значений множественного атрибута меньше указанного в поиске
Кол-во значений больше	Будут показаны те позиции, у которых количество значений множественного атрибута больше указанного в поиске
Кол-во значений равно	Будут показаны те позиции, у которых количество значений множественного атрибута равно указанному в поиске

Для каждого критерия поиска нужно заполнить поле значений в соответствии с выбранным условием сравнения. Значение может быть введено с клавиатуры, с помощью команды **Вставить** из контекстного меню браузера или выбрано из выпадающего списка значений. Можно также вставить список значений из буфера обмена в качестве критериев расширенного поиска. Например, можно скопировать в буфер из текстового или табличного файла 100 строк с GUID записей и вставить их в критерий поиска записей по идентификаторам.

Поиск выполняется по <ИЛИ> в рамках одного атрибута. Таким образом, если перечислить значения для атрибута (в том числе с использованием фасета), то *Система* будет искать все записи, которые содержат хотя бы одно из них. Для числа с плавающей точкой поиск выполняется по точному совпадению значения. При поиске учитывается тип числа с плавающей точкой: число, диапазон, число с отклонениями. Поиск выполняется по мере заполнения значений критериев. Результаты поиска остаются в таблице данных.

Если расширенным поиском ничего не найдено, то «худое дерево» строится до текущей выделенной группы.

Перечень критериев поиска сохраняется для справочной группы. При необходимости любой выбранный критерий может быть деактивирован без удаления из списка. Кнопка активации / деактивации располагается слева от наименования критерия поиска.

Чтобы удалить один из критериев поиска, нужно нажать кнопку, расположенную в правой части строки с параметром поиска, или удалить значение, по которому проводился поиск. Для удаления значения можно воспользоваться всплывающей кнопкой ***Очистить*** (Рисунок 51).

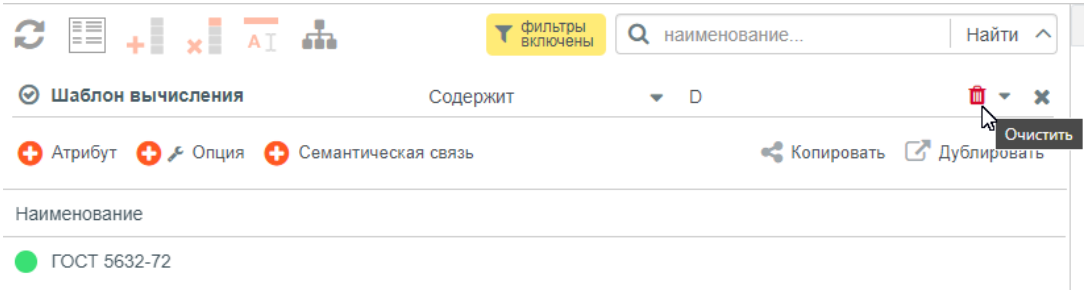


Рисунок 51 - Удаление значения критерия расширенного поиска

В качестве опций поиска пользователю доступны несколько вариантов (Рисунок 52, Рисунок 53).

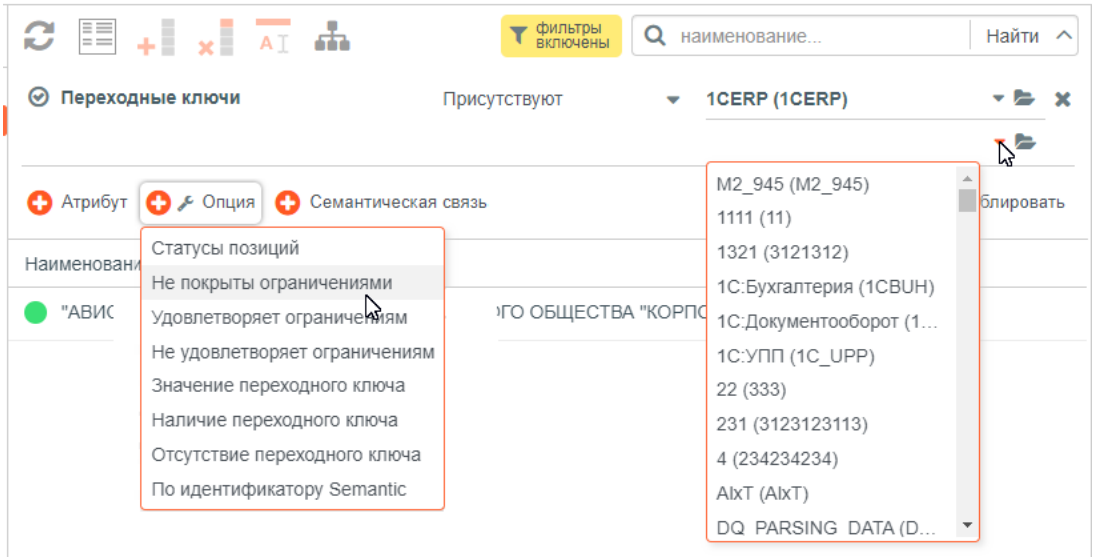


Рисунок 52 - Опции расширенного поиска

- **Статусы позиций** - статус выбирается щелчком левой кнопки мыши на соответствующем статусе (см. п. 3.14.20). Можно выбрать один,

несколько либо все статусы;

- **Не покрыты ограничениями** - опция позволяет найти:
  - *Любое ограничение* – все записи, которые не покрываются ни одним ограничением;
  - *Обязательные ограничения* – все записи, которые не покрываются ни одним обязательным ограничением;
  - *Необязательные ограничения* – все записи, которые не покрываются ни одним необязательным ограничением;
  - *Уникальные атрибуты* – все записи, для которых отсутствуют уникальные атрибуты;
  - *Обязательные атрибуты* – все записи, для которых отсутствуют обязательные атрибуты;
  - *Ограничительные таблицы* – все записи, которые не покрываются ограничительными таблицами;
  - *Функции проверки данных* – все записи, которые не покрываются ни одной функцией проверки данных;
- **Удовлетворяют ограничениям:**
  - *Все ограничения* – все записи, которые удовлетворяют все имеющимся ограничениям;
  - *Обязательные ограничения* – все записи, которые удовлетворяют обязательным ограничениям;
  - *Необязательные ограничения* – все записи, которые удовлетворяют всем необязательным ограничениям;
  - *Уникальные атрибуты* – все записи, которые имеют уникальные значения в уникальных атрибутах;
  - *Обязательные атрибуты* – все записи, для которых заполнены обязательные атрибуты;
  - *Ограничительные таблицы* – все записи, которые удовлетворяют всем ограничительным таблицам;

- *Функции проверки данных* – все записи, которые удовлетворяют всем функциям проверки данных;

- **Не удовлетворяет ограничениям -** записи, несоответствующие уникальности или обязательности атрибутов, функциям проверки данных, ограничительным таблицам. При этом в таблице данных останутся только те записи, которые не удовлетворяют выбранным типам ограничений, в том числе:
  - *Обязательные ограничения* – позволяет найти все записи, которые не удовлетворяют обязательным ограничением;
  - *Необязательные ограничения* – позволяет найти все записи, которые не удовлетворяют необязательным ограничением;
- **Значение переходного ключа** – поиск записи по значению переходного ключа с внешней информационной системой. При активизации данной опции в дополнительном окне нужно выбрать внешнюю ИС и указать возможное значение ключа;
- **Наличие переходного ключа** – в качестве значения указывается наименование внешней ИС, с которой у объектов поиска может быть переходной ключ. Выбор условия «*С любой системой*» позволит выполнить поиск всех записей, которые имеют хотя бы один переходной ключ. В качестве значений может быть скопирован список систем или ключей из буфера обмена;
- **Отсутствие переходного ключа** – в качестве значения указывается наименование внешней ИС, с которой у объектов поиска отсутствует переходной ключ;
- **По идентификатору Semantic** – поиск объекта по его GUID;
- **Только актуальные позиции** – поиск позиций, находившихся в статусе «Нормализована» на указанную дату;

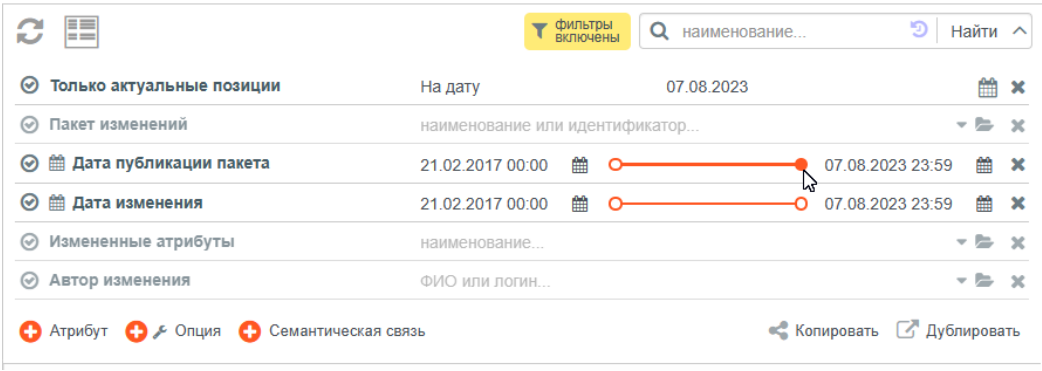


Рисунок 53 – Дополнительные опции расширенного поиска

- **Пакет изменений** – поиск позиций, входивших в указанный пакет изменений. Поиск выполняется по наименованию или идентификатору пакета;
- **Дата публикации пакета** – поиск позиций, которые находились в пакетах изменений, опубликованных в указанный в поиске период времени. Период указывается в формате <ДД.ММ.ГГГГ> + <ЧЧ:МН>;
- **Дата изменения** – поиск позиций по дате изменения. Дата задается как период времени в формате <ДД.ММ.ГГГГ> + <ЧЧ:МН>;
- **Измененные атрибуты** – поиск позиций, для которых были изменены значения перечисленных атрибутов. атрибуты указываются по наименованию. Эта опция не доступна при выполнении расширенного поиска в режиме работы с пакетом изменений;
- **Автор изменения** – поиск позиций, которые были изменены указанным в поиске пользователем. Для пользователя указывается логин в Системе или Ф.И.О. Эта опция не доступна при выполнении расширенного поиска в режиме работы с пакетом изменений;
- **Изменённые позиции** – опция доступна только в режиме редактирования (управления пакетами изменений). Позволяет найти все измененные позиции в рамках активного пакета изменений;

Для всех опций добавлена возможность задавать сразу несколько значений, для выполнения поиска по «ИЛИ». Например, можно найти все позиции, которые удовлетворяют обязательным «ИЛИ» необязательным ограничениям.

Опция **Семантическая связь** позволяет вести расширенный поиск по семантическим связям объекта. Для данного вида расширенного поиска выбор условий поиска не требуется.

При включении опции **Семантическая связь**, *Система* по умолчанию предлагает поиск любой связи для выбранной записи (Рисунок 54). В этом случае будут отфильтрованы все записи, связанные с указанной в качестве критерия, по любой семантической связи. Можно вставить список значений из буфера. Допускается вставлять наименования позиций или их GUID. При вставке выполняется поиск подходящих позиций по наименованию или их GUID.

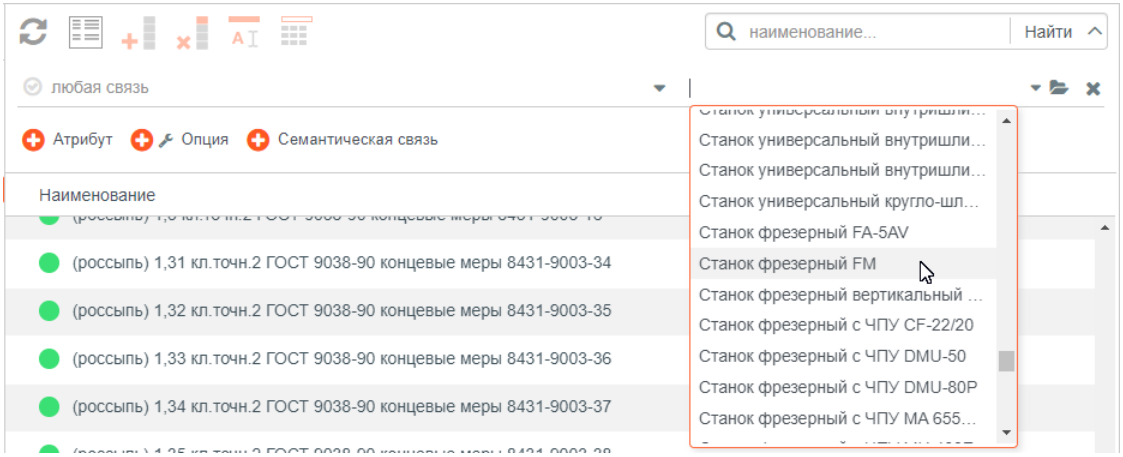


Рисунок 54 - Выбор записи для просмотра всех семантических связей

Если в качестве критерия выбран определенный тип семантической связи для указанной записи, то в фильтр попадут только те записи, которые соответствуют данному типу связи и совместимы с текущей записью (Рисунок 55). Направление стрелок в наименовании связи указывает на результат поиска. Например, для связи «Номенклатура --> является аналогом --> Номенклатура» будут найдены все записи, для которых указанная в

расширенном поиске в качестве значения, является аналогом. И наоборот, для связи «Номенклатура --> имеет аналоги --> Номенклатура», будут отфильтрованы все записи, являющиеся аналогами для выбранной.

Количество одновременно используемых в расширенном поиске семантических связей не ограничено. Для добавления следующей нужно повторно нажать кнопку ***Семантическая связь***.

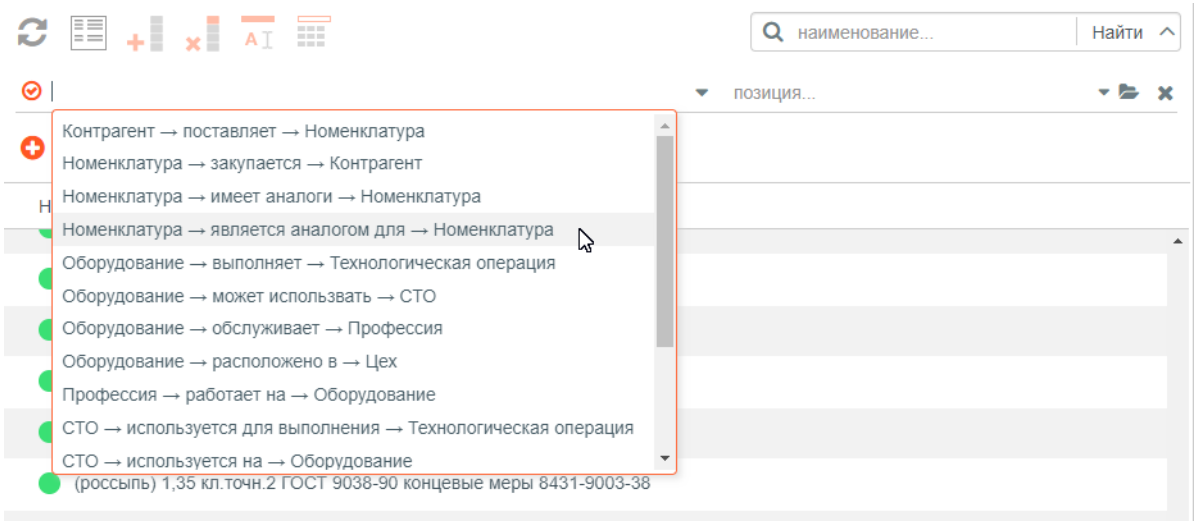


Рисунок 55 - Поиск записи по определенной семантической связи

После выбора любого атрибута и указания хотя бы одного значения пользователю становятся доступны кнопки (Рисунок 56):

- ***Копировать*** – формирует и сохраняет в буфер специальную адресную строку, в которой сохраняются текущие критерии и значения расширенного и быстрого поиска, что позволяет передать её другому пользователю в письме или мессенджере, чтобы тот смог по ссылке запустить идентичный расширенный поиск;
- ***Дублировать*** – по данной кнопке открывается новая вкладка, полностью повторяющая текущую, включая запущенный расширенный поиск.

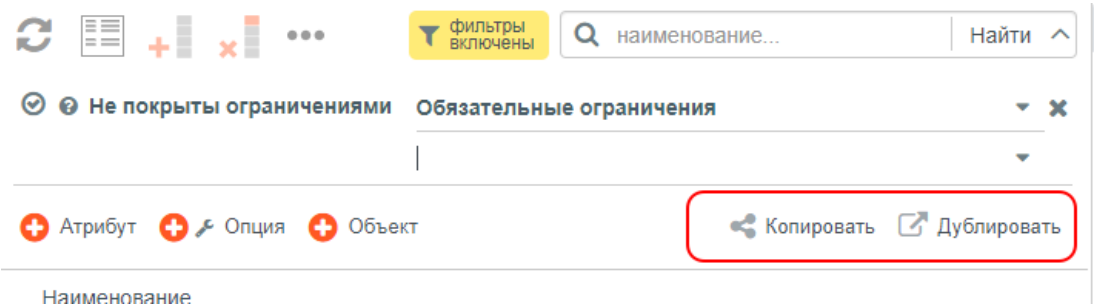


Рисунок 56 - Кнопки для копирования и дублирования вариантов расширенного поиска

Чтобы отменить результаты поиска следует нажать кнопку ***Фильтры включены*** для их выключения.

3.8 Использование Избранного

Избранное служит для оперативного доступа к наиболее часто используемым объектам НСИ и является аналогом личной библиотеки специалиста. Пользователь может добавлять в **Избранное** любые объекты из *Системы*. **Избранное** уникально для каждого пользователя, зарегистрированного в *Системе*.

3.8.1 Добавление информации в Избранное

Чтобы добавить справочник, ветку справочника или объект из таблицы в **Избранное** нужно установить на него курсор в дереве классификатора или на сетке данных, нажать кнопку ***Избранное*** и выбрать команду ***Добавить в избранное*** (Рисунок 57).

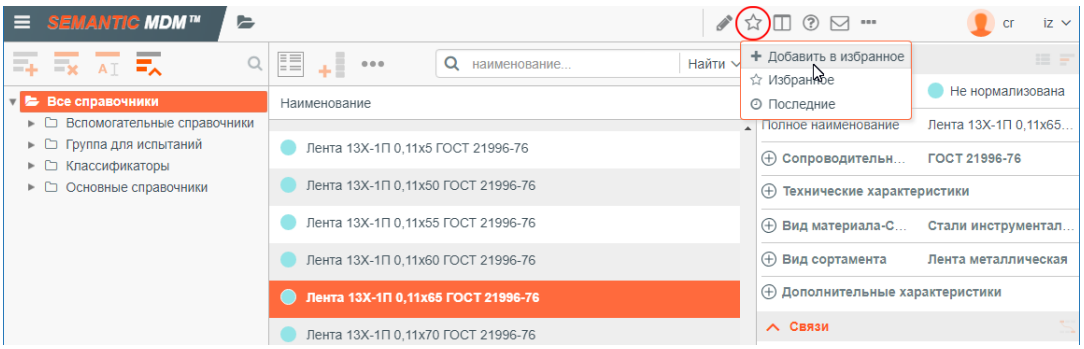


Рисунок 57 - Добавление объекта в избранное

После добавления в **Избранное** пользователь может выбрать показывать ли объект на стартовой странице и в какую именно папку нужно разместить объект (Рисунок 58).

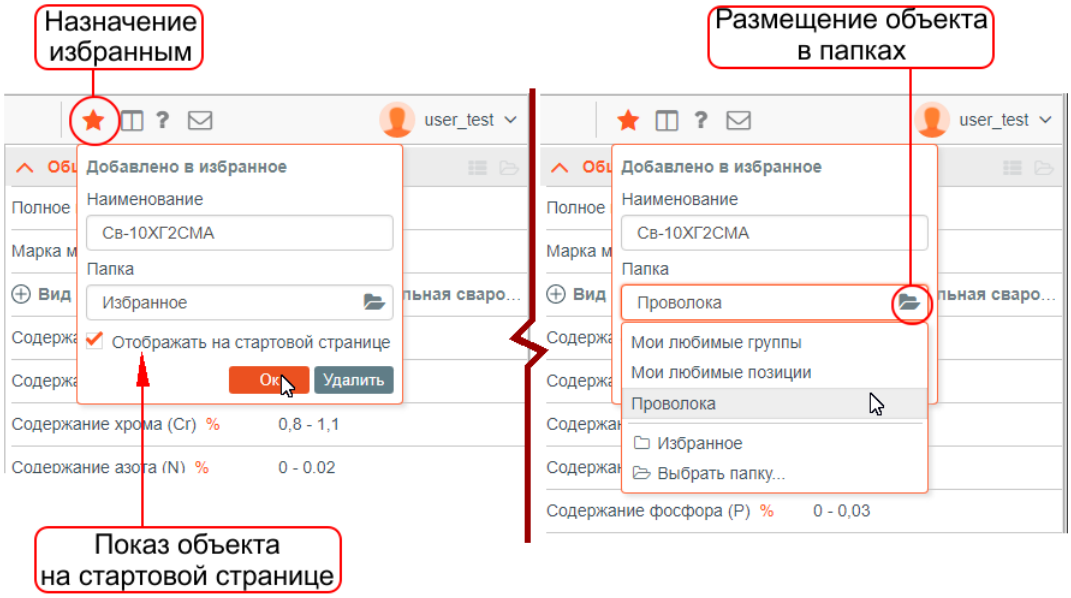


Рисунок 58 - Выбор размещения объекта в избранном

При выборе папки для размещения объекта пользователю доступно размещение в папке **Избранное** (по умолчанию), в ранее созданной папке (разделе избранного) или в новой папке. Объект также можно разместить в папках, доступ в которые возможен со стартовой страницы *Системы*.

3.8.2 Создание новой папки в разделе «Избранное»

Для создания новой папки необходимо в окне добавления избранного использовать команду **Выбрать папку**.

В открывшемся окне **Избранное (**Рисунок 59**)** нажать кнопку ***Создать папку***. При этом будет создана новая папка, для которой следует указать название и затем нажать кнопку ***Выбрать***, при этом объект будет размещён во вновь созданной папке.

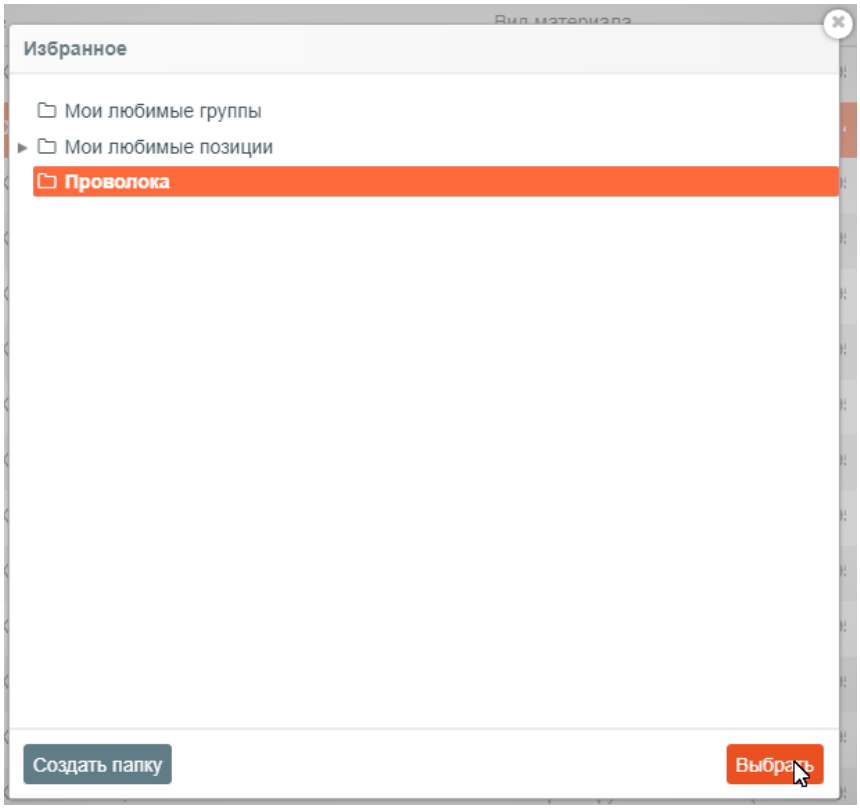


Рисунок 59 - Создание и выбор новой папки в избранном

3.8.3 Редактирование избранного

Для объекта, размещённого в избранном, доступно редактирование места размещения (папки), а также видимости на стартовой странице.

Чтобы отредактировать избранное, нужно установить на требуемый объект курсор, нажать кнопку ***Избранное*** и выбрать команду ***Изменить в избранном*** (Рисунок 60).

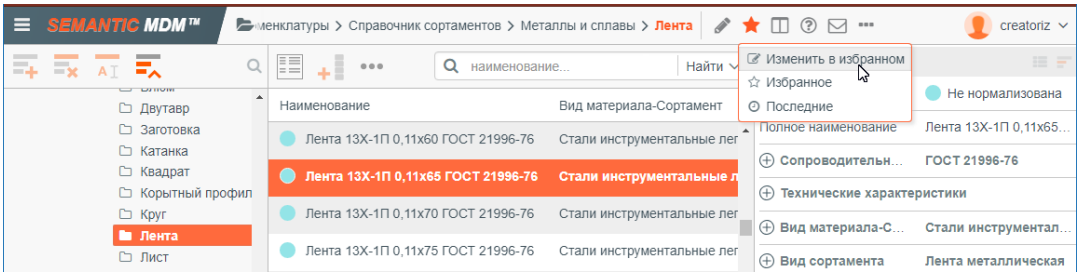


Рисунок 60 - Вызов редактирования избранного объекта

Для редактирования доступно размещение объекта и его видимость на стартовой странице.

3.8.4 Раздел «Избранное»

Для работы с объектами в избранном следует нажать кнопку ***Избранное*** и выбрать команду **Избранное**. Кроме того, к избранному можно обратиться со стартовой страницы *Системы*, нажав на кнопку  рядом с заголовком *Избранное*.

В открывшемся окне (Рисунок 61) доступны следующие операции[11]:

- **Переименовать** – даёт возможность переименовать любую папку в избранном;
- **Добавить корневую папку** – создаёт новую папку в корневой директории;

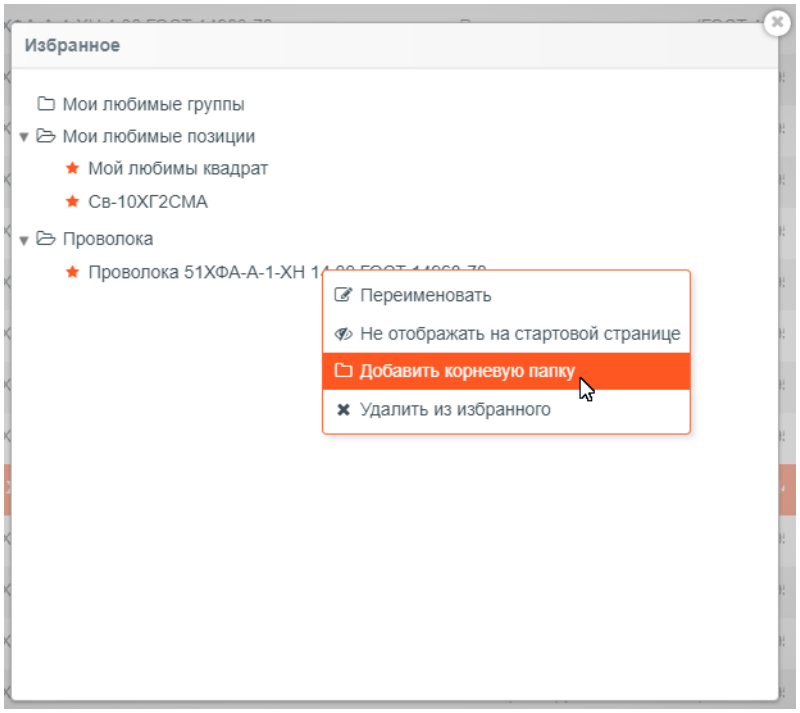


Рисунок 61 - Работа в окне «Избранное»

- **Не отображать на стартовой странице** – объект не будет показан на стартовой странице *Системы*;
- **Удалить из избранного** – удаляет объект из категории избранного.

[11]Объекты в окне **Избранное** можно перемещать из одной папки в другую путём их «перетаскивания» (функция Drag&Drop)

3.8.5 Работа с избранным на стартовой странице

Добавленные в избранное объекты представлены в виде ссылок в разделе *Избранное* на стартовой странице *Системы*. Для обращения к ним достаточно нажать на соответствующую кнопку в этом разделе.

Если на избранном объекте вызвать контекстное меню (под правой кнопкой мыши), то для него будут доступны две команды (Рисунок 62) – **Не отображать на стартовой странице** и **Удалить из избранного**[12].

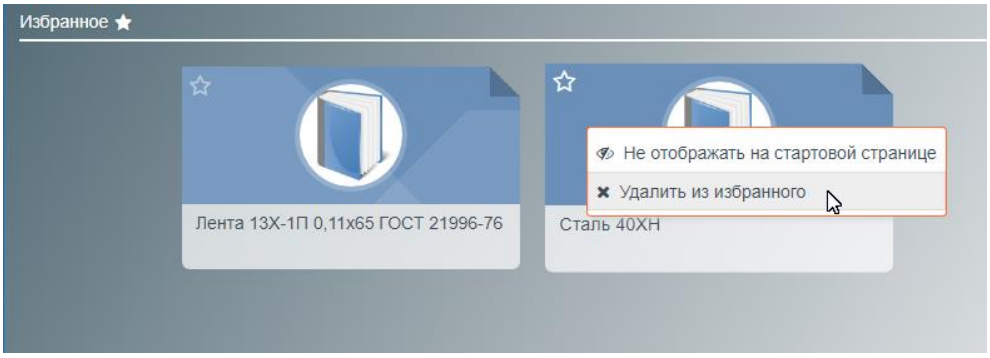


Рисунок 62 - Представление избранного на стартовой странице

Кроме настроек для избранного объекта, со стартовой страницы можно обратиться к разделу **Избранное** в целом. Для этого нужно нажать на кнопку ★, находящуюся радом с заголовком *Избранное*.

3.9 Создание пользовательского объекта НСИ

Для удобства работы с наиболее часто применяемыми справочными объектами пользователь может создать свой собственный справочник и разместить в нем нужную информацию. Данный функционал доступен только для пользователей с привилегией Просмотр всех справочников.

Пользовательский справочник представляет собой раздел, содержащий ссылки на группы, размещённые в основных справочниках, оперативный доступ к которым интересует пользователя.

[12]Удаление объекта из избранного не приводит к его удалению из справочной системы

Для создания такого объекта НСИ необходимо:

- на стартовой странице в области объектов НСИ нажать кнопку ***Создать пользовательский справочник*** (Рисунок 63);



Рисунок 63 - Создание пользовательского справочника

- в открывшемся окне (Рисунок 64) Пользовательский справочник ввести в разделе **Наименование** название справочника на русском и (если требуется) на английском языке;

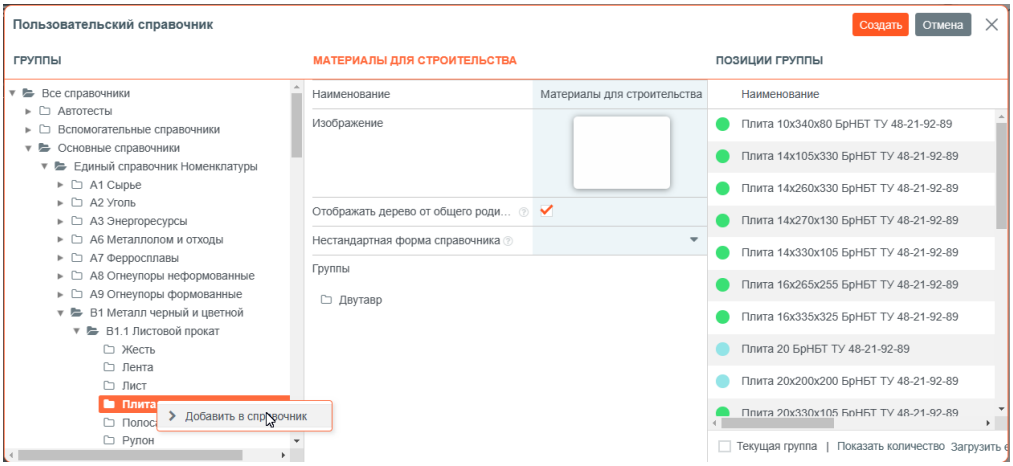


Рисунок 64 - Редактирование пользовательского справочника

- в ячейке **Изображение** можно присоединить рисунок, который будет отображаться для этого объекта НСИ на стартовой странице. Для этого используется кнопка ***Загрузить изображение*** или двойной щелчок левой кнопкой мыши на области изображения;

- **Отображать дерево от общего родителя** – при включении опции дерево будет построено, начиная с общего родителя выбранных справочных групп. При выключенной опции дерево будет содержать только выбранные и их дочерние группы;
- **Нестандартная форма справочника** – позволяет указать альтернативную реализацию представления справочника, загруженную из расширения. Если нестандартная форма справочника не указана или расширение не найдено, используется представление по умолчанию;
- установить курсор на интересующей группе объектов и вызвать из контекстного меню команду **Добавить в справочник**. Можно также воспользоваться функцией Drag&Drop. Добавленные группы списком представлены в разделе **Группы**.

Чтобы удалить группу из состава пользовательского справочника, достаточно щёлкнуть на её наименовании в разделе **Группы** и нажать кнопку X, появляющуюся рядом с названием.

Для удаления пользовательского объекта НСИ можно воспользоваться кнопкой ***Удалить***[13] на пиктограмме объекта НСИ на стартовой странице.

На стартовой странице также можно вызвать пользовательский объект НСИ для редактирования. Для этого нужно нажать кнопку ***Изменить*** на пиктограмме.

При наведении курсора на кнопку ***Переместить*** (Рисунок 65), он меняет вид на четырёхнаправленную стрелку. Зажав левую кнопку мыши, можно перенести пиктограмму (функция Drag&Drop) пользовательского объекта НСИ на новое место.

[13] *Удаление пользовательского объекта НСИ не приводит к удалению информации из основных объектов НСИ*



Рисунок 65 - Пиктограмма пользовательского объекта НСИ на стартовой странице

3.10 Группирование объектов НСИ

Для удобства пользователей с целью сокращения места, занимаемого пиктограммами справочников на стартовой странице, объекты могут быть объединены в группы (Рисунок 66).

Чтобы объединить несколько справочников в одну группу, достаточно на пиктограмме добавляемого объекта НСИ зажать левую кнопку мыши и перетащить её на пиктограмму другого объекта НСИ. При этом объекты НСИ объединятся в группу (папку). При добавлении объекта НСИ в уже существующую группу процедура будет аналогичной, только в этом случае пиктограмму справочника перетаскивают на пиктограмму группы.

Новая группа справочников по умолчанию получает наименование «Новая папка». Чтобы изменить название, нужно вызвать на пиктограмме группы контекстное меню и выбрать команду **Переименовать**, после чего ввести новое название группы непосредственно на пиктограмме.

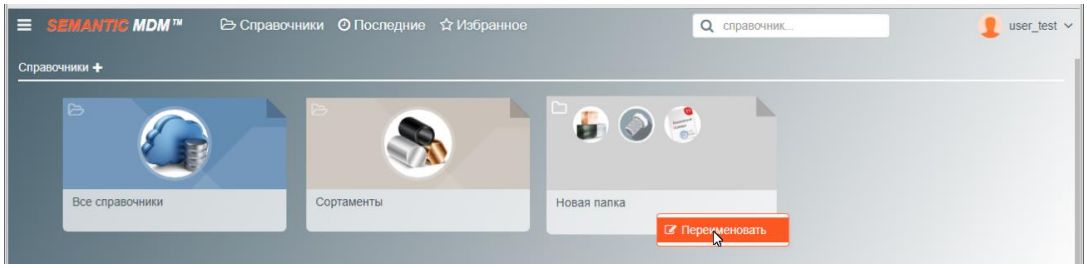


Рисунок 66 - Объединение справочников в группу

Для того, чтобы извлечь справочник из группы, нужно зажать на пиктограмме справочника левую кнопку мыши и перетащить пиктограмму на невидимую границу[14] между остальными пиктограммами (Рисунок 67).



Рисунок 67 - Извлечение объекта НСИ из группы

3.11 Интеграция справочников в SolidWorks

Semantic MDM интегрируется с CAD-системой SolidWorks©, что позволяет использовать материалы и сортаменты при конструкторском проектировании, обеспечивая запись информации в конструкторский документ в соответствии с российскими стандартами.

3.11.1Указание материала в чертеже изделия

В результате интеграции на панели инструментов SolidWorks© в режиме работы с чертежом пользователю доступна отдельная вкладка Semantic MDM, на которой расположены три кнопки – Указать материал, Разместить материал детали и Удалить материал (Рисунок 68).

[14]*Невидимая граница появляется только при перемещении пиктограмм (Drag&Drop). Если перемещение не происходит, нажмите на клавиатуре клавиши <Ctrl>+<R> и повторите процедуру перетаскивания*

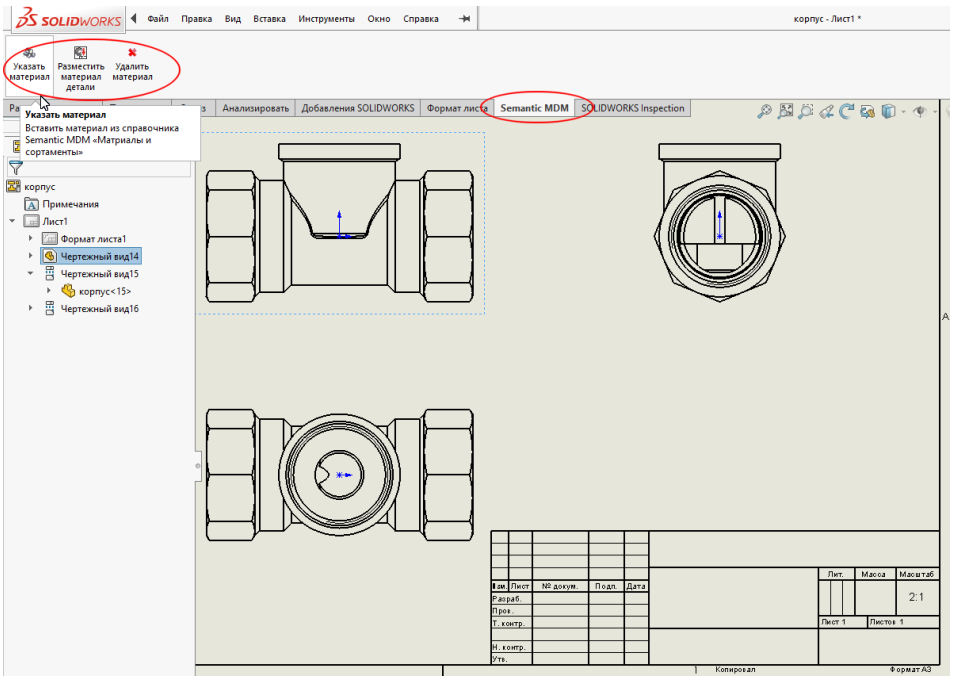


Рисунок 68 - Панель Semantic MDM в рабочем окне Solid Works для чертежа

При нажатии на кнопку **Указать материал** будет запущена Система, причём для работы будут доступны только два справочника – Материалы и Сортаменты (Рисунок 69).

В нужном справочнике следует найти (см. раздел 3.7) нужную позицию и, установив на неё курсор, нажать кнопку **Выбрать**, которая расположена в правом нижнем углу окна справочника.

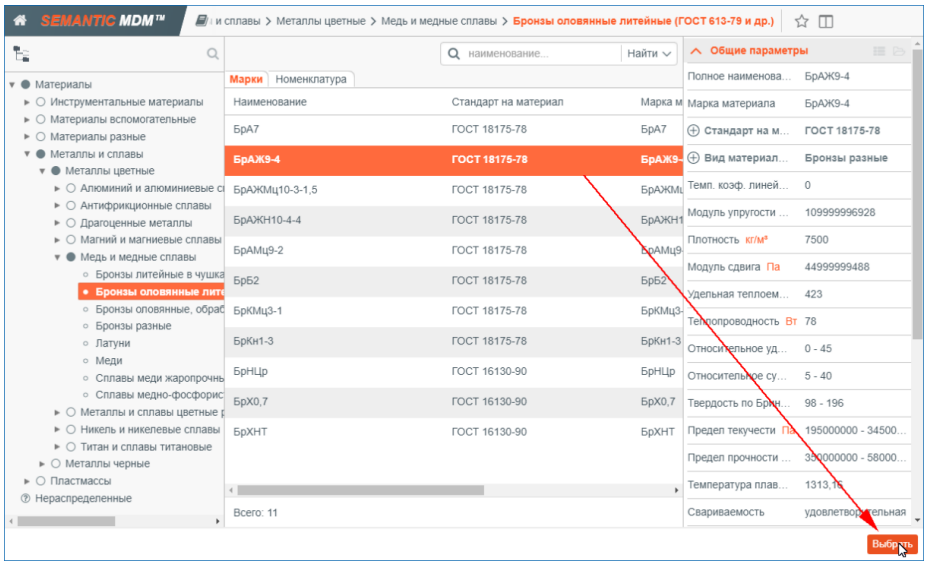


Рисунок 69 - Выбор материала из справочника

Выбранный материал будет присвоен чертежу изделия и появится в соответствующей графе основной надписи.

Чтобы удалить материал достаточно нажать кнопку Удалить материал на панели инструментов Semantic MDM.

Кнопка Разместить материал предназначена для указания в чертеже материала, присвоенного модели изделия.

3.11.2 Добавление материала к 3D-модели

При работе с 3D-моделью на панели инструментов CAD-системы пользователю также доступна отдельная вкладка Semantic MDM, на которой расположены две кнопки – Указать материал и Удалить материал (Рисунок 70).

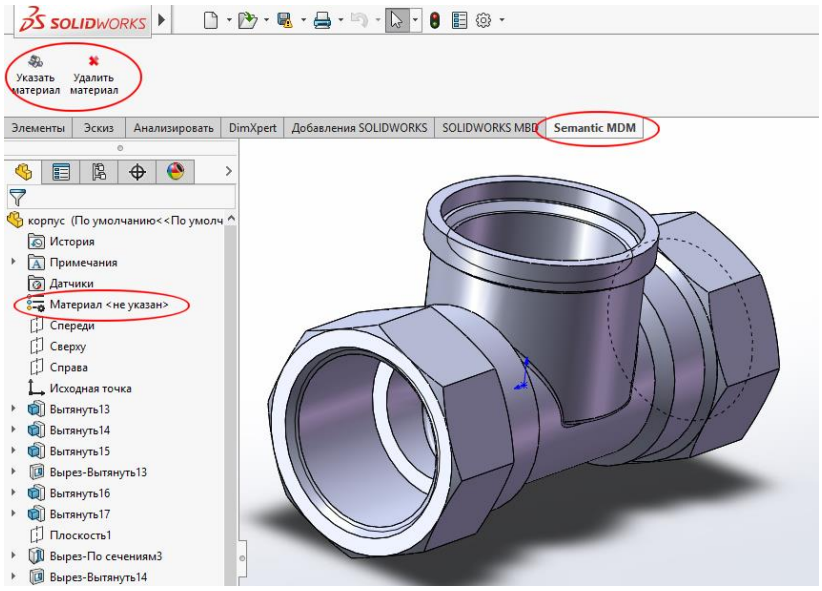


Рисунок 70 - Панель Semantic MDM в рабочем окне Solid Works

При нажатии на кнопку Указать материал будет запущен WEB-портал, причём для работы будут доступны только два справочника – Материалы и Сортаменты.

В нужном справочнике следует найти (см. раздел 3.7) нужную позицию и, установив на неё курсор, нажать кнопку Выбрать, которая расположена в правом нижнем углу окна справочника. После этого выбранный материал или

сортамент будут записаны в дереве построения (Рисунок 71).

Если материал выбирался для модели, то, по умолчанию, ей будет присвоен цвет, который зарегистрирован в справочнике для этого материала.

Кнопка Удалить материал позволяет удалить сведения о материале, присвоенном модели.

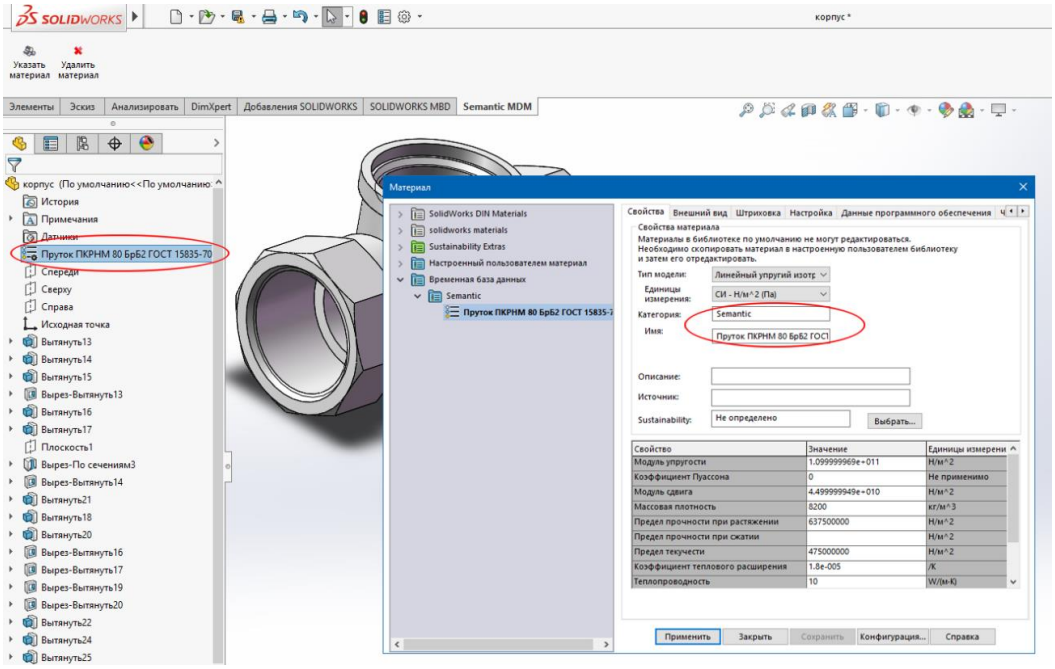


Рисунок 71 - Материал 3D-модели в дереве и окне редактирования

При создании чертежа из модели необходимо вписать данные о материале в основную надпись. Для этого в режиме чертежа используется кнопка Разместить материал. При нажатии на эту кнопку материал, присвоенный ранее 3D-модели, будет размещён в соответствующей графе (Рисунок 72).

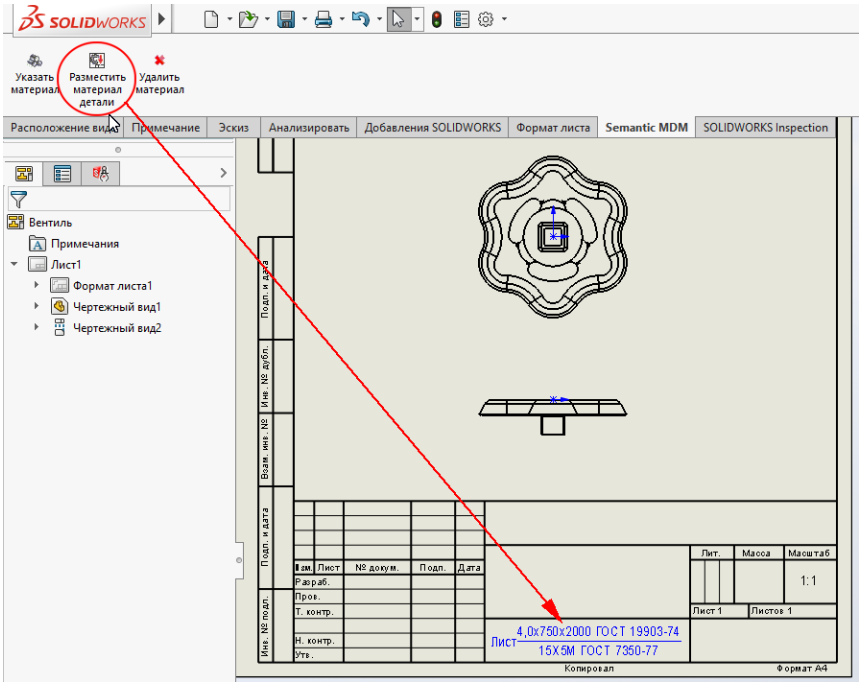


Рисунок 72 - Размещение материала 3D-модели в основной надписи

3.12 Работа с заявками

Для внесения любых изменений в нормативно-справочную информацию, выступающую в качестве мастер-данных для различных информационных систем, необходимо соблюдение определённой последовательности действий, которые обеспечат правильность и качество вводимых данных.

Обеспечения единообразия подходов для работы с информацией можно добиться использованием заявительного способа работы, при котором любое изменение данных производится на основании заявки пользователя.

Для работы с заявками в Системе присутствует **Журнал заявок**, который может быть вызван с помощью кнопки **Навигация** (Рисунок 73).

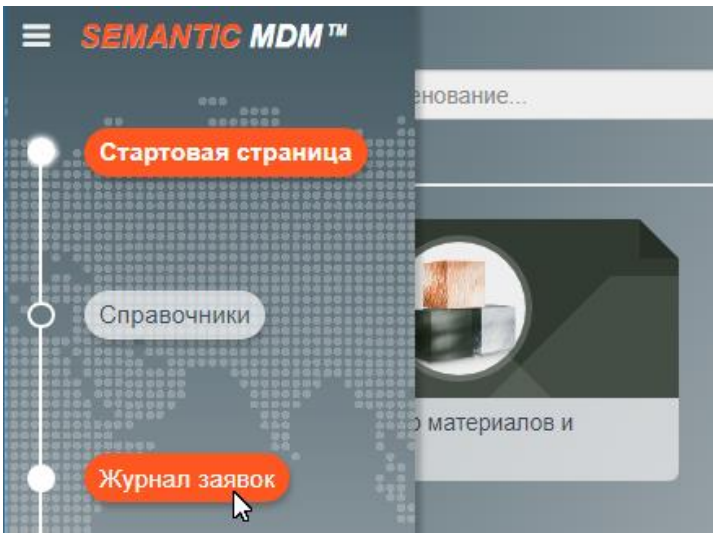


Рисунок 73 - Вызов Журнала заявок

Внешний вид Журнала заявок представлен на рисунке ниже. Журнал заявок содержит набор вкладок, в центре каждой вкладки располагается таблица с заявками, слева от таблицы, на панели отделенной сплиттером, располагаются фильтры, а справа от таблицы - карточка заявки, имеющая свои собственные вкладки.

Заявки, созданные пользователем (Заявителем), располагаются на вкладке Мои заявки. Для заявок, находящихся у пользователя в обработке как у Эксперта НСИ, используется вкладка Текущие. На вкладке Обработанные находятся заявки, по которым приняты решения на любом из этапов обработки заявки. Вкладка Просроченные заявки содержит список заявок, которые не были обработаны в установленные сроки. Вкладка Диспетчеризация предназначена для заявок, требующих ручной или автоматической диспетчеризации для передачи в обработку. На вкладке Все заявки находится перечень всех заявок, зарегистрированных в Системе. Вкладка Отчёты предназначена для формирования и просмотра отчётов по заявкам, поданным пользователями, её описание приведено в разделе «Отчеты» (см. раздел 3.25). Вкладка Отчеты SLA содержит соответствующие отчеты с оценками эффективности работы экспертов НСИ по выполнению заявок. Вкладки Отчеты и Отчеты SLA доступны для просмотра только пользователям с привилегией Менеджер заявок.

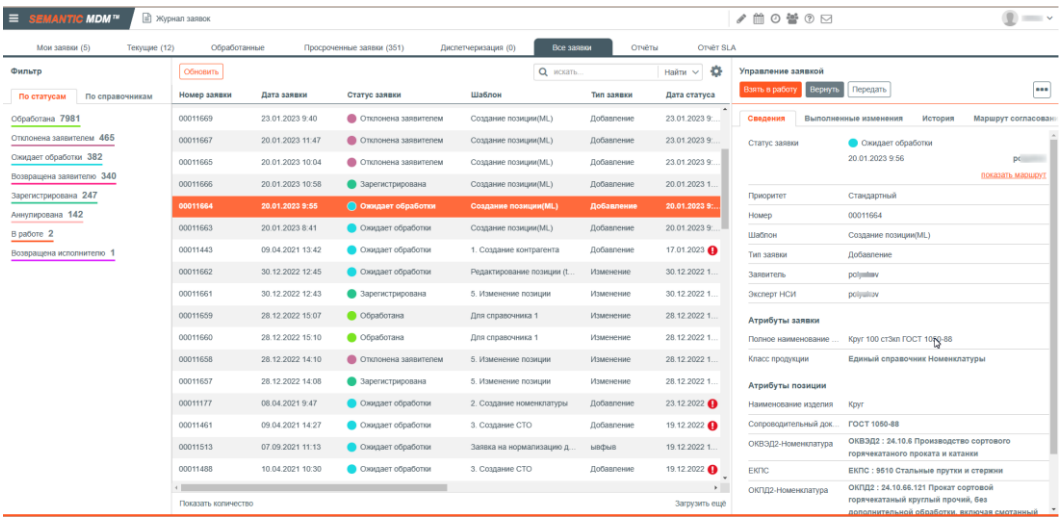


Рисунок 74 - Вид подсистемы «Журнал заявок»

В таблице каждой заявке присваивается:

- **Номер заявки** – порядковый номер заявки в *Системе*;
- **Номер группы** - порядковый номер для группы заявок;
- **Дата заявки** – дата и время создания заявки *Заявителем*;
- **Шаблон** – наименование шаблона, по которому была создана заявка[15];
- **Тип заявки** – описание типа заявки (добавление, удаление и т.д.);
- **Дата статуса** – дата и время присвоения текущего статуса заявки. При создании совпадает с датой заявки, затем изменяется одновременно со статусом;
- **Статус заявки** – состояние заявки в текущий момент времени. Статус также отображается в строке заголовка в виде цветной полоски (подчёркивания);
- **Срок обработки** – дата и время, до наступления которых данная заявка должна быть обработана. При наступлении указанного момента *Системой* будет отправлено уведомление о завершении срока исполнения / согласования заявки;

[15]*Добавление (создание) новых объектов и изменение существующих возможно только с использованием шаблонов*

- **Срок завершения** – отображается количество времени, оставшегося до завершения срока обработки заявки;
- **Срок уведомления** – дата и время предварительного уведомления о необходимости выполнения заявки;
- **Этап** – текущий этап при использовании маршрутов согласования;
- **Справочник** – наименование справочника, к которому относится заявка;
- **Приоритет** – отображается приоритет, установленный для заявки и определяющий сроки ее обработки;
- **Заявитель** – имя пользователя, создавшего заявку;
- **Эксперт НСИ** – имя пользователя (пользователей), которые создают или изменяют данные по заявке;
- **Срок уведомления о заявке с открытым пакетом изменений** – дата, когда *Специалист по ведению НСИ* будет уведомляться Системой о наличии открытых, но не обработанных пакетов изменений по заявкам. Настраивается в шаблоне заявки;
- **Срок аннулирования заявки с открытым пакетом изменений** – дата, когда будет удален автоматически созданный пакет изменений по заявке, если он не был обработан специалистом по ведению НСИ. Настраивается в шаблоне заявок.
- Столбцы со значениями атрибутов заявок и атрибутов позиций, соответствующие указанным в использованных шаблонах

При необходимости количество видимых колонок таблицы с заявками может быть изменено. Для этого следует нажать кнопку ***Настройки таблицы*** ⚙, расположенную над сеткой данных. В открывшемся окне ***Настройка видимости колонок*** (Рисунок 75) следует снять или установить маркеры напротив требуемых полей и нажать кнопку ***ОК***.

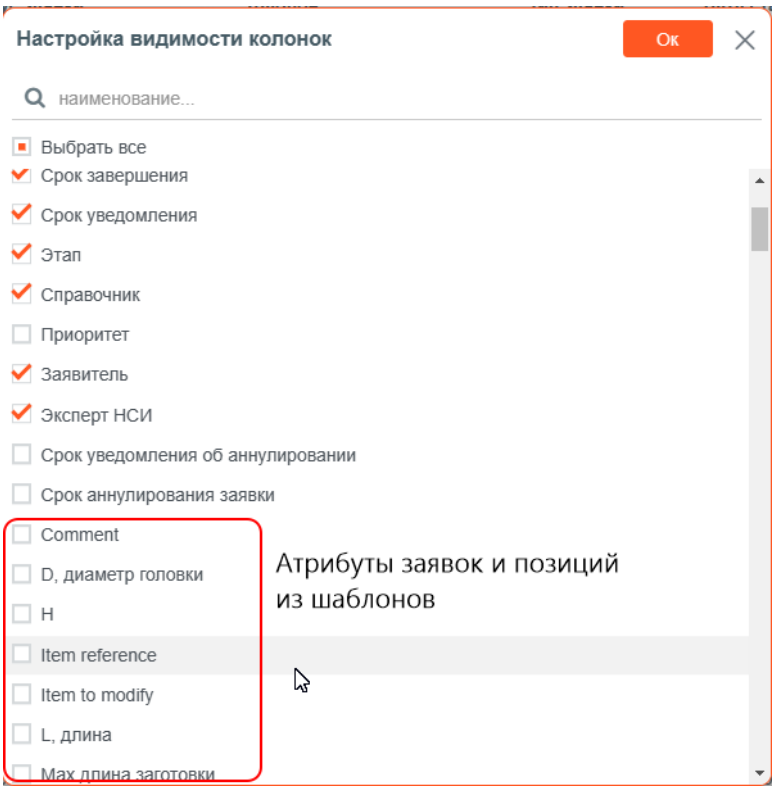


Рисунок 75 - Настройка колонок таблицы Журнала заявок

При необходимости последовательность расположения столбцов таблицы может быть изменена[16]. Для этого необходимо зажать левую кнопку мыши на заголовке столбца и, не отпуская её, переместить его в нужное место до появления специального разделителя ⋮,после чего отпустить кнопку мыши.

3.12.1 Поиск по заявкам

В Журнале заявок доступен быстрый и расширенный поиск по списку заявок.

Ячейка быстрого поиска располагается над таблицей с заявками справа и позволяет производить поиск по:

- **номеру заявки** – поиск может проводится как по номеру с лидирующими нулями, так и без них;
- **наименованию заявки**;

[16]Системная колонка со статусами записей и колонка «Наименование», если она отображается, не могут быть перемещены!

- **значениям атрибута заявки и позиции**;
- **комментариям пользователя** при переводе заявки на новый статус;
- **по пользователю**, указанному в заявке.

При щелчке по заголовку любого из столбцов таблицы с заявками происходит их сортировка. Первый щелчок – сортировка по возрастанию, второй – по убыванию. По умолчанию заявки в таблице располагаются по убыванию даты создания. Выше расположены последние по времени создания заявки.

Для быстрой фильтрации по статусу заявки, дате подачи и пр. можно воспользоваться фильтрами, располагающимися слева от таблицы на отдельной панели (Рисунок 76). Щелчок по наименованию в фильтре оставит в таблице только те заявки, которые соответствуют выбранному. Фильтрация выполняется:

- для вкладок **Мои заявки, Обработанные, Все заявки:**
  - по статусам;
  - по справочникам;
- для вкладок **Текущие** и **Просроченные** заявки:
  - по статусам;
  - по датам;
- для вкладки **Диспетчеризация**:
  - Эксперты НСИ;
  - по приоритетам;
  - по датам.

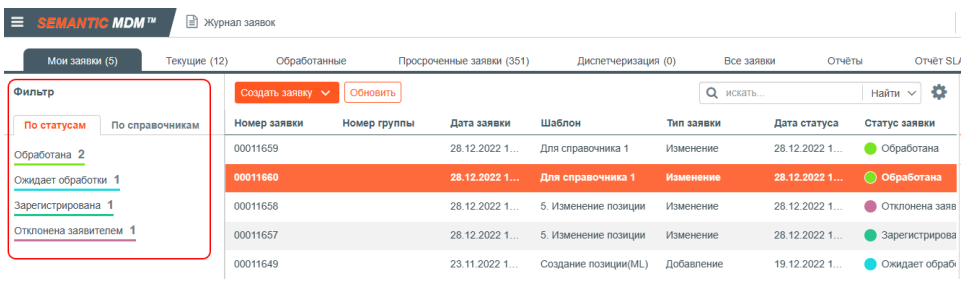


Рисунок 76 - Фильтры для Журнала заявок

Кнопка Обновить, расположенная над таблицей, предназначена для актуализации данных, представленных в таблице.

Фасетный поиск может осуществляется по значениям любого из столбцов таблицы, кроме колонок Номер заявки и Этап.

Расширенный поиск в Журнале заявок доступен по кнопке ⌄, расположенной рядом с ячейкой быстрого поиска (Рисунок 77). Расширенный поиск доступен по следующим критериям:

- по атрибутам, включая атрибуты заявки и позиции;
- по срокам, включая:
  - срок обработки;
  - дата заявки;
  - дата статуса;
  - срок уведомления;
  - срок уведомления об аннулировании;
  - срок аннулирования заявки.

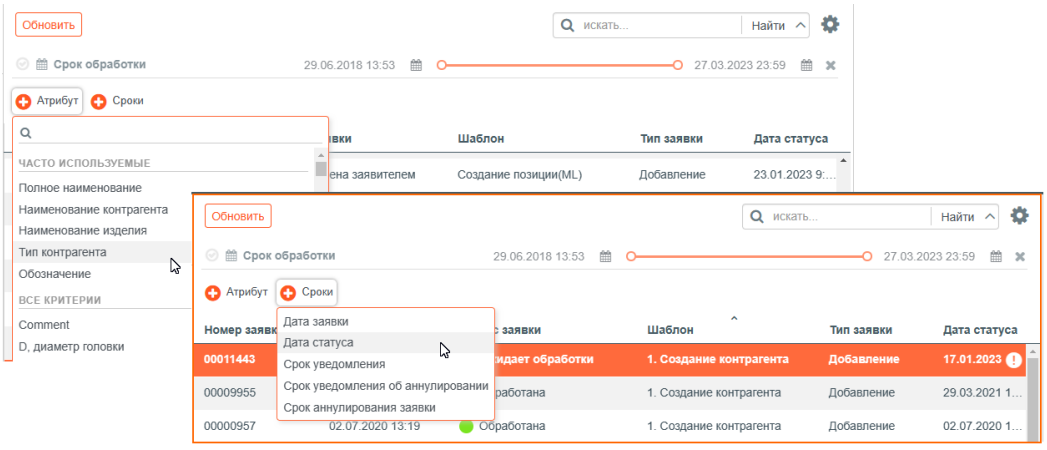


Рисунок 77 - Расширенный поиск по заявкам

Для того, чтобы выйти из режима показа результатов поиска, необходимо нажать кнопку Фильтры включены для их выключения.

3.12.2 Создание одиночной заявки

Для создания заявки пользователю необходимо:

находясь в справочной группе или на сетке данных нажать на панели инструментов таблицы кнопку **Заявки** и выбрать команду **Создать заявку** (Рисунок 78) либо перейти в **Журнал заявок** (также можно использовать кнопку Навигация);

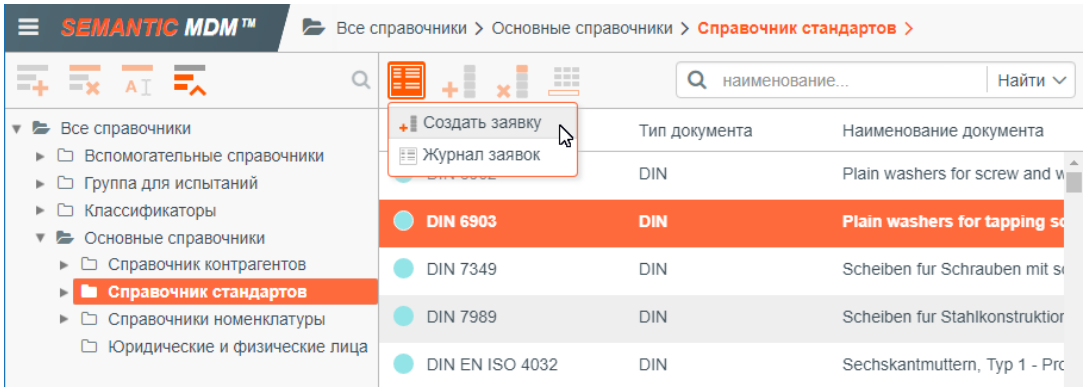


Рисунок 78 - Запрос на создание заявки на позицию из справочника

- если заявка создаётся в **Журнале заявок**, то нужно нажать кнопку Создать заявку, расположенную над таблицей заявок слева на вкладке Мои заявки;
- в открывшемся окне **Выберите шаблон** заявки щёлкнуть левой кнопкой мыши на требуемом шаблоне и нажать кнопку Ок (доступность шаблона заявки для выбора определяется его настройками). Если для текущего пользователя в справочнике отсутствуют шаблоны заявок, то ему будет выдано соответствующее предупреждение (Рисунок 79). Если перечень шаблонов велик, можно воспользоваться строкой поиска по названию, располагающейся в верхней строке окна;

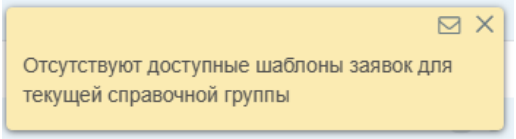


Рисунок 79 - Предупреждение Системы об отсутствии доступных пользователю шаблонов заявок в справочнике

При создании заявки пользователю отображаются не более 100 шаблонов (Рисунок 80).

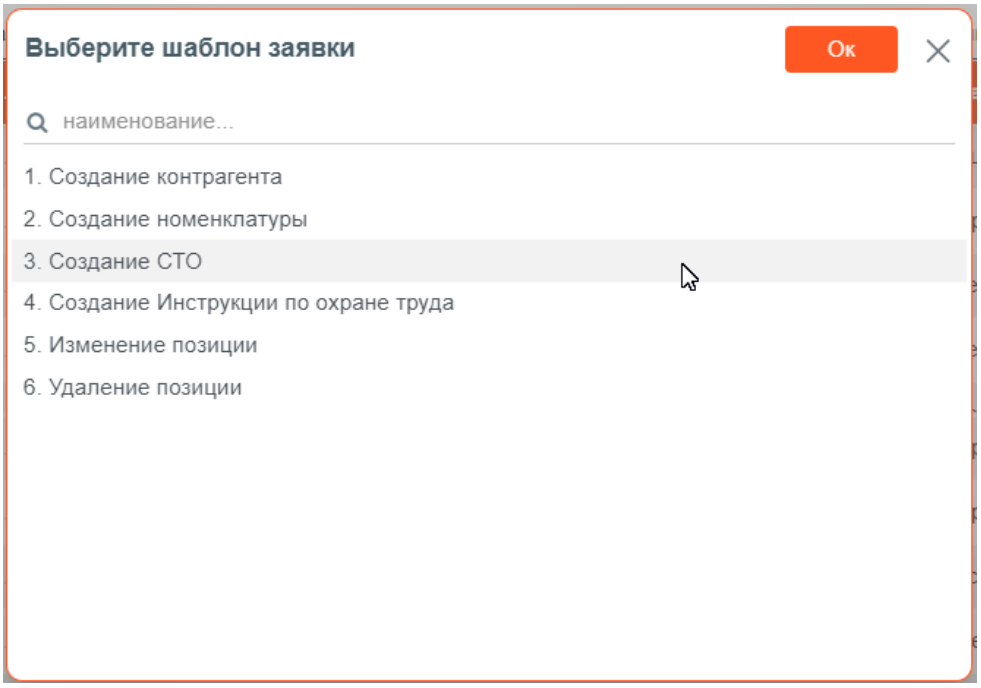


Рисунок 80 - Вид окна выбора шаблона заявки

Для справочной группы доступны только те шаблоны заявок, в которых есть атрибут заявки с включённой опцией «Инициализировать заявку группой/позицией» в область которого входит текущая справочная группа, а также те шаблоны, для которых область не определена или отсутствует атрибут заявки с включённой опцией «Инициализировать заявку *группой/позицией*».

- в открытом окне заявки на вкладке **Сведения** (Рисунок 81) следует заполнить предлагаемые атрибуты заявки вручную, с использованием выпадающих списков или используя обращение к соответствующим справочникам и классификаторам. Атрибут, отмеченный *, должен быть заполнен обязательно, если он пустой, то заявка не будет отправлена;

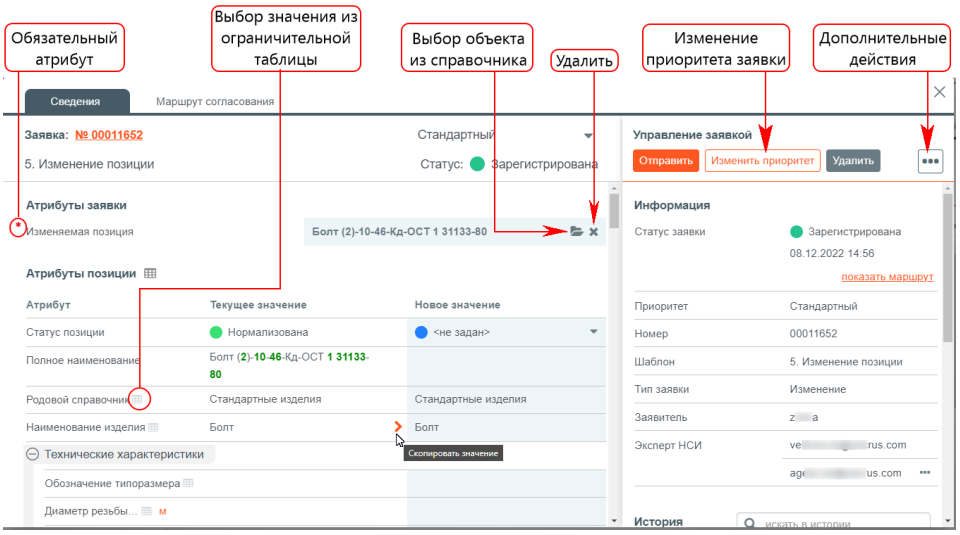


Рисунок 81 - Вкладка **Сведения** с атрибутами заявки на изменение позиции

Состав атрибутов заявки и позиции, которые необходимо заполнять пользователю, определяется в шаблоне заявки и не может быть изменен на стадии ее заполнения.

- если атрибуты позиции могут быть заполнены с использованием ограничительных таблиц (см. п. 3.13.6), то рядом с таким атрибутом будет расположена пиктограмма Ограничительные таблицы, при нажатии на которую можно выбрать значение атрибута из ограничительной таблицы. При этом сохраняется возможность указать значение вручную;
- в заявках можно указать исторические значения атрибутов позиции, а также даты начала и окончания действия позиции (Рисунок 82) (только в заявках на изменение). Для указания исторических значений необходимо нажать кнопку Настройка историчности, расположенную рядом с заголовком Атрибуты позиции. В этом случае станут доступными поля для ввода / изменения дат действия позиции, а также кнопки для вызова окна указания исторических значений, всплывающие в правой части ячейки со значением атрибута. Исторические значения атрибутов указываются во всплывающем окне История значений для каждого атрибута по отдельности;

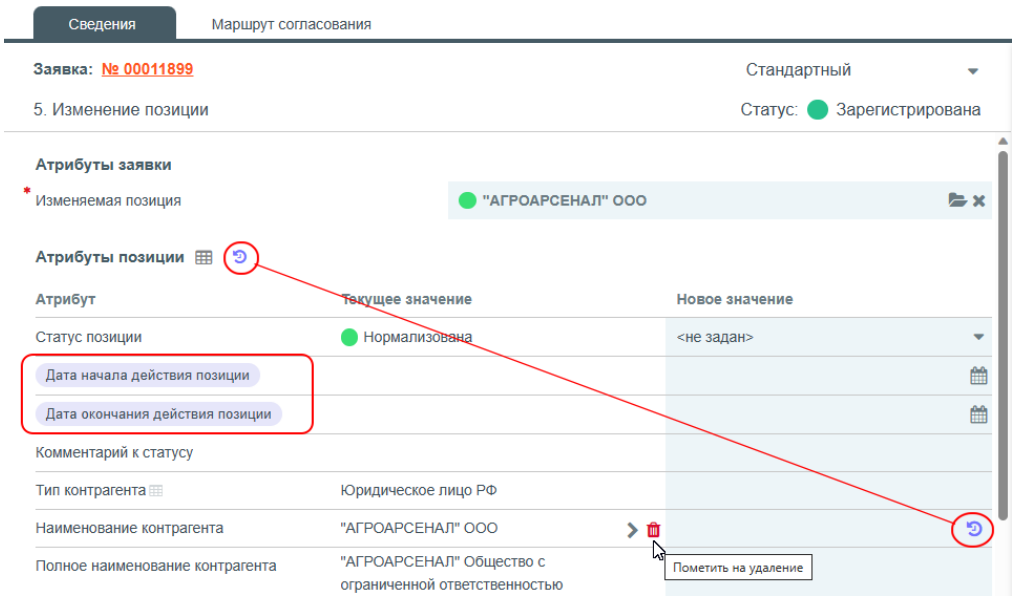


Рисунок 82 - Даты начала и окончания действия позиции

- в заявках на изменение (в шаблонах заявок, где доступны текущие и новые значения атрибутов) можно установить маркер удаления значения (всплывающая пиктограмма **Пометить на удаление** справа от значения в столбце **Текущее значение**) как для атрибута связи, так и для простого атрибута;
- если в заявке присутствует атрибут связи, отвечающий за композируемые позиции, то можно выполнить добавление новой композируемой позиции или изменение существующих (Рисунок 83) (только в заявках на изменение). При добавлении или редактировании композируемой позиции её атрибуты отображаются на белом фоне, остальные атрибуты заявки закрашиваются серым. Если в заявке на изменение редактируются сразу несколько композируемых значений, то они выбираются из композируемого справочника, который поднимается по кнопке **Добавить для изменения**. Добавляемые, удаляемые или изменяемые композируемые позиции отмечаются соответствующей пиктограммой. Удаляемая позиция также отмечается перечеркнутым шрифтом. Изменяемые позиции имеют в заявке два столбца - **Текущее значение** и **Новое значение**. Работа с ними аналогична работе с любыми изменяемыми позициями. Если

композируемая позиция была добавлена по ошибке, то для ее удаления из списка в заявке (не в справочнике) используется кнопка **Удалить**.

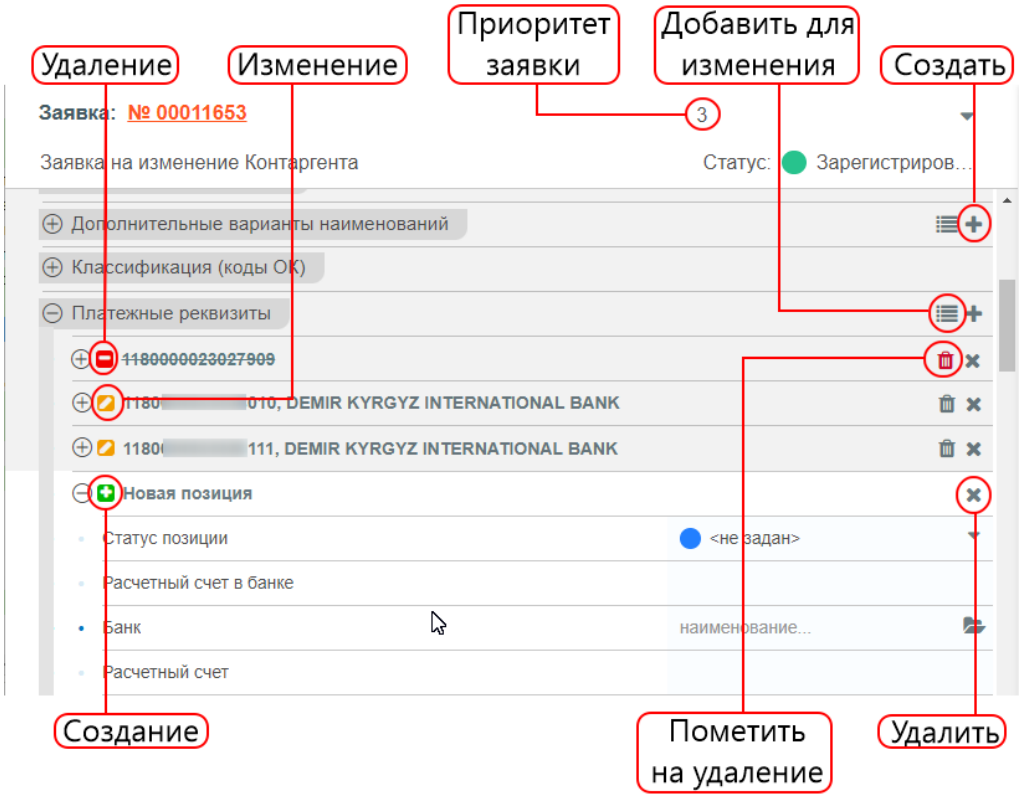


Рисунок 83 - Композируемые позиции в заявке на новую позицию

- в процессе заполнения заявки пользователю доступен поиск дубликатов создаваемой позиции. Для этого в нижней части формы заявки расположена кнопка **Поиск подходящих позиций**. Поиск выполняется в рамках указанной в заявке справочной группы по совпадению введенных значений. Для работы поиска подходящих позиций необходимо, что бы в справочной группе, куда подается заявка, был настроен поиск дубликатов (см. п. 3.15). Если поиск дубликатов не настроен, то поиск подходящих позиций из заявки не работает;

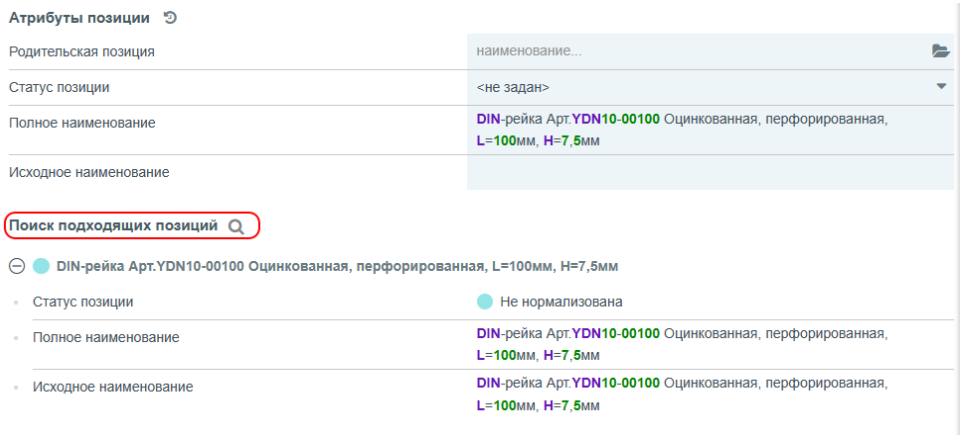


Рисунок 84 - Выполнение поиска подходящих позиций при создании заявки

- при необходимости нажать на кнопку **Изменить приоритет**, выбрать в списке доступный приоритет и нажать кнопку **Ок**. Приоритет заявки по умолчанию настраивается в шаблоне. Возможность повышения приоритета определяется соответствующей настройкой (см. п. 3.12.11) (Рисунок 85).

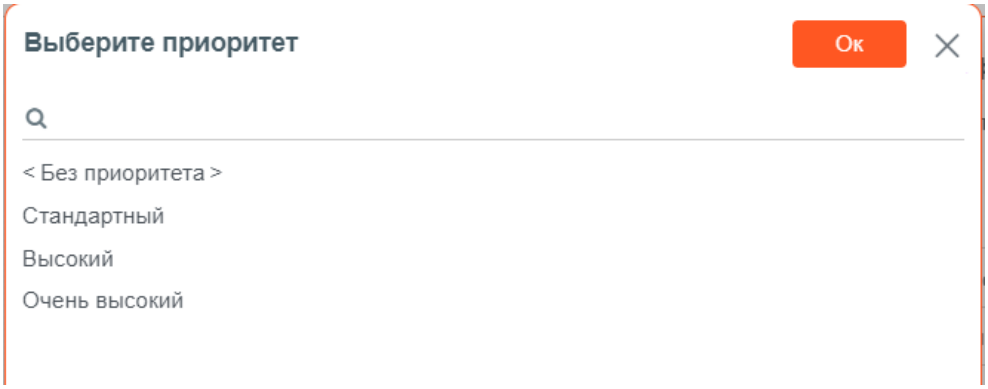


Рисунок 85 - Выбор приоритета, устанавливаемого заявке заявителем

- после заполнения всех необходимых атрибутов позиции, нажать кнопку **Отправить**. После этого появится дополнительное окно **Подтверждение** (Рисунок 86), в котором Заявитель может дать необходимые пояснения (можно оставить окно пустым), а также прикрепить файл, затем нажать на кнопку **Ок**, в результате чего заявка будет создана и отправлена на обработку. Комментарий и файл, прикрепленный в окне подтверждения доступны для просмотра и скачивания (файл) в карточке заявки на вкладке **История**.

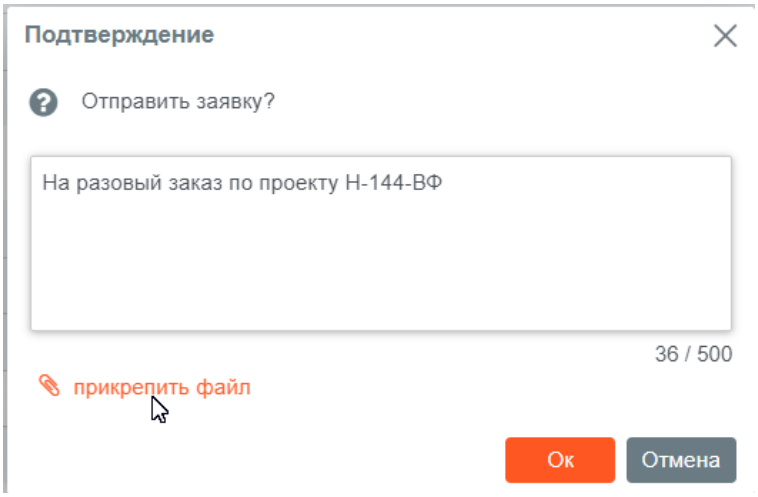


Рисунок 86 - Окно Подтверждение создания заявки

Перед отправкой на исполнение, система выполнит проверку заполненных атрибутов заявки и атрибутов позиции на уникальность среди всех ранее отправленных заявок этого же шаблона и, если такая заявка найдется, то пользователю будет выдано предупреждение с указанием номеров найденных заявок с точно такими же значениями атрибутов (Рисунок 87). Если пользователь все же отправит заявку на исполнение, то в историю заявки будет сделана соответствующая запись, о том что, не смотря на предупреждение, пользователь отправил заявку на исполнение.

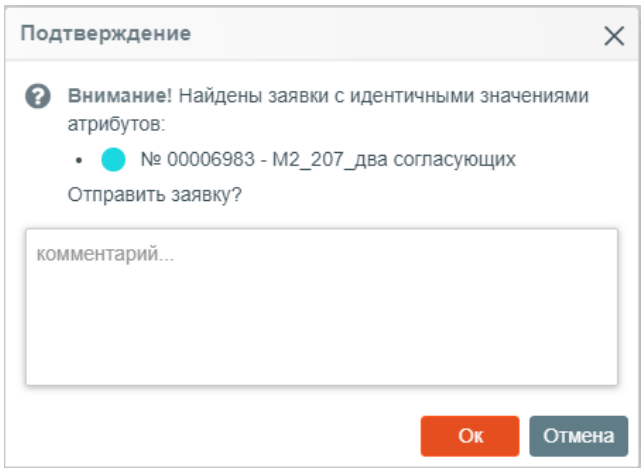


Рисунок 87 - Проверка заявки на уникальность

В панели **Управление заявкой** (Рисунок 88) располагаются кнопки **Отправить**, **Изменить приоритет** и **Удалить** . В качестве дополнительных опций при работе с заявкой можно открыть заявку в новой вкладке браузера,

а также создать копию текущей заявки. Создание копии позволяет создавать однотипные заявки для одной и той же группы справочника.

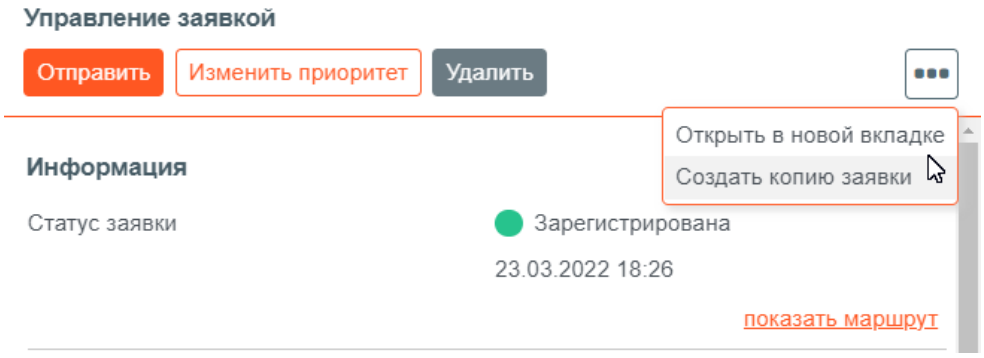


Рисунок 88 - Дополнительные опции при работе с заявкой

Вкладка **Маршрут согласования** (Рисунок 89), а также ссылка показать маршрут на вкладке **Сведения** в карточке заявки позволяет посмотреть этапы согласования заявки по выбранному шаблону. Если этапы согласования отсутствуют, то решение по заявке принимается Экспертом НСИ. При просмотре маршрута доступны сведения о наименовании этапа согласования, количестве и именах согласующих лиц, а также о разрешении на редактирование данных в заявке на каждом из этапов.

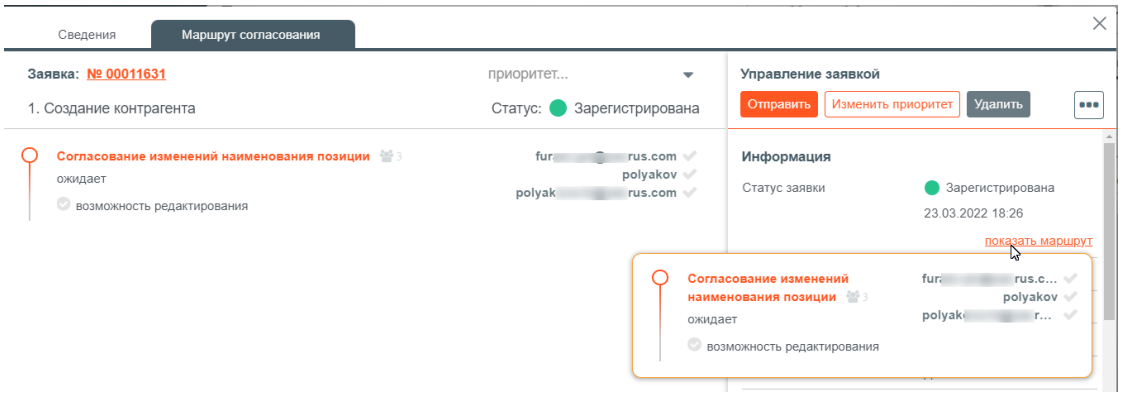


Рисунок 89 - Просмотр маршрута согласования в заявке

Если заявка создана, но не была отправлена, то она получает статус – *Зарегистрирована*. Чтобы вернуться к работе с такой заявкой достаточно выполнить на ней двойной щелчок левой кнопкой мыши.

На вкладке **Мои заявки** (Рисунок 90) пользователю (*Заявителю*) доступны действия с заявками, вызываемые из контекстного меню. Помимо общих команд – **Создать заявку**, **Открыть заявку**, **Открыть в новой вкладке** и **Создать копию заявки**, есть ряд дополнительных действий (Таблица 11), которые зависят от текущего статуса заявки.

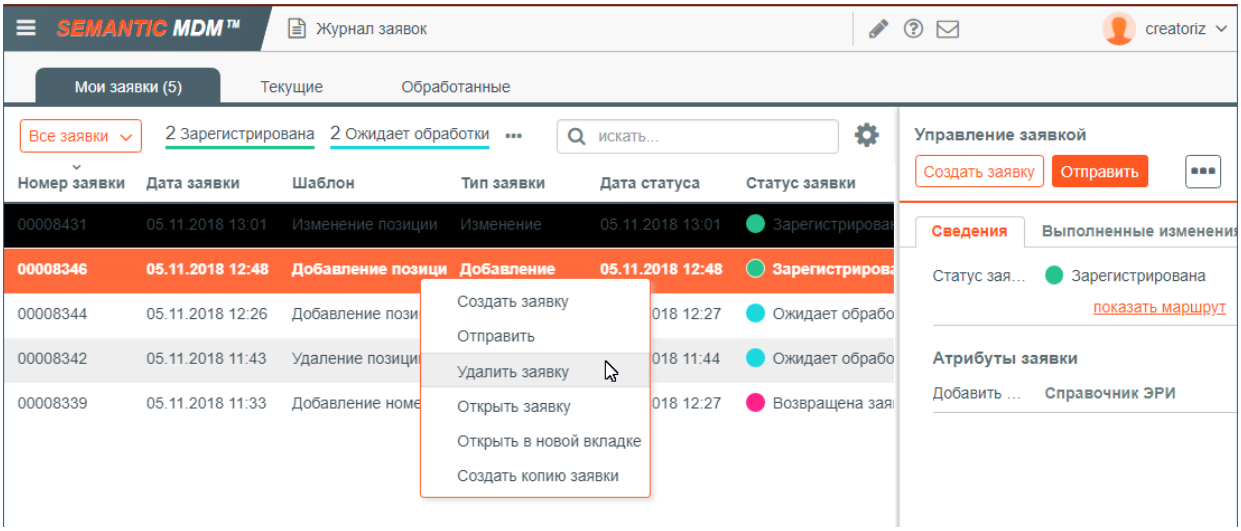


Рисунок 90 - Контекстное меню для работы с заявками

Таблица 11. Действия со статусами заявок

Статус	Команда	Комментарий
Зарегистрирована	**Отправить**	Заявка будет создана и отправлена *Эксперту НСИ*
	Удалить	Заявка будет удалена
Ожидает обработки	**Вернуть**	Заявка будет возвращена заявителю с присвоением соответствующего статуса. При возвращении требуется указать причину в окне **Подтверждение.** Отзыв заявки разрешен только в том случае, если с заявкой еще никто не работал**!**
Ожидает диспетчеризации		
Возвращена заявителю	**Уточнить**	Заявка будет открыта для редактирования, после чего её можно вновь отправить *Эксперту НСИ*
	Отменить	Заявка получит статус **Отменена заявителем**. При отмене требуется указать причину в окне **Подтверждение**

Изменение атрибутов заявки и атрибутов позиции в заявке *Заявителем* допускается только, если статус заявки – **Зарегистрирована**, **Возвращена заявителю**. Заявка, которая была принята и выполнена получит статус l **Обработана**.

На вкладке **Просроченные заявки** располагаются те заявки, по которым превышены сроки исполнения (подробнее о сроках см. п. 3.12.25).

По всем событиям, происходящим с заявкой, *Заявитель* получает уведомления от браузера. Уведомления могут дублироваться на электронную почту пользователя, если это настроено в его профиле. Кроме этого, процесс выполнения заявки отображается на дополнительной вкладке **История** в карточке заявки.

3.12.3 Создание групповой заявки

Групповая заявка позволяет быстро создать несколько однотипных заявок с разными значениями атрибутов и отправить их на исполнение. Создание групповой заявки возможно только из подсистемы **Журнал заявок.**

Для создания групповой заявки следует:

- нажать кнопку ***Создать группу заявок*** (Рисунок 91);

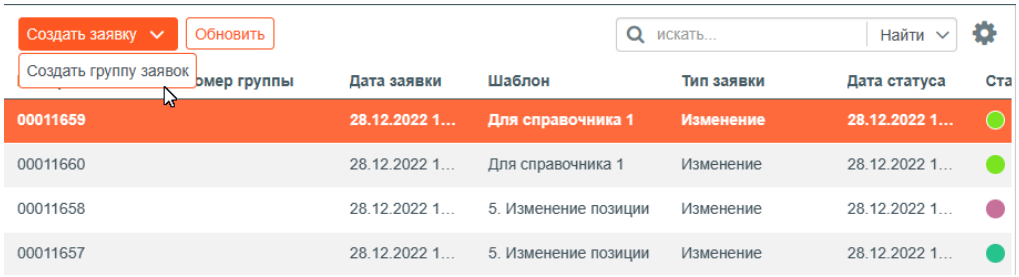


Рисунок 91 - Вызов команды «Создать группу заявок»

- дать группе заявок наименование, нажав на кнопку **Изменить** и введя новое название;
- выбрать один или несколько шаблонов заявки последовательно. Для добавления второго и последующих шаблонов нужно нажать кнопку **Выбрать шаблон заявок** (Рисунок 92) и указать нужный шаблон. Каждый шаблон располагается на отдельной вкладке группы заявок.

Для выбора доступны только те шаблоны заявок. у которых установлен признак «Разрешить использование в групповых заявках».;

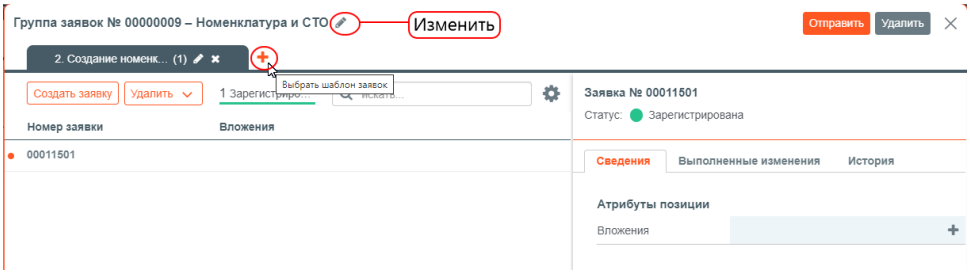


Рисунок 92 - Изменение наименования группы заявок
и добавление нового шаблона

- на вкладке с наименованием нужного шаблона нажать кнопку **Создать заявку** требуемое количество раз. Каждая заявка будет иметь свой номер и обрабатываться экспертом НСИ отдельно. В таблице с заявками количество столбцов соответствует количеству атрибутов, требующих заполнения;
- при ручном создании заявок значения атрибутов могут быть заполнены вручную либо на вкладке **Сведения** в карточке заявки либо непосредственно в таблице. Для входа в редактирование ячейки таблицы необходимо нажать клавишу <Enter>, для подтверждения ввода требуется нажать <Enter> повторно. При редактировании можно воспользоваться выпадающим списком, если указана позиция – аналог (Рисунок 93);

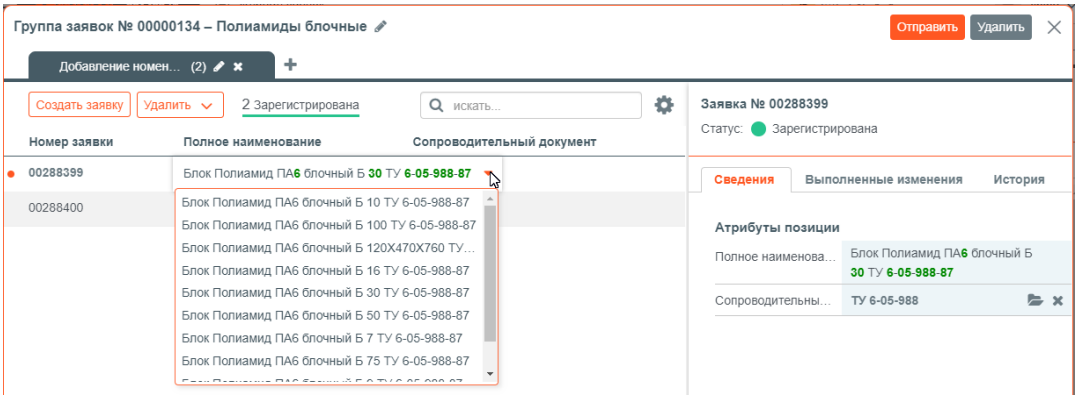


Рисунок 93 - Редактирование значения атрибута в таблице

- также можно не создавать заявки, а скопировать данные из таблицы (Excel) и вставить их через <Ctrl>+<V> в табличное представление с заявками. В этом случае Semantic MDM автоматически создаст необходимое количество заявок и заполнит атрибуты заявок значениями. Если порядок следования колонок в таблице Excel не совпадает с порядком следования атрибутов в табличном представлении заявок, то копирование можно выполнять по колонкам;
- после создания и заполнения всех необходимых заявок нажать кнопку **Отправить**.

После отправки групповой заявки на исполнение, Заявитель может открывать групповую заявку и отслеживать исполнение каждой из заявок, поданных в рамках данной группы заявок. Исполнители обрабатывают заявки групповой заявки по отдельности, в соответствии с установленными правилами и регламентами.

3.12.4 Диспетчеризация заявок

Если в шаблоне заявки (см. п. 3.12.13) настроены диспетчеры заявок или если в шаблоне указаны несколько экспертов НСИ, но при этом не указаны диспетчеры и не определены условия передачи заявки одному из них, то такие заявки после отправки получают статус **Ожидает диспетчеризации** и размещаются на вкладке **Диспетчеризация** (Рисунок 94). Работа с заявками на этой вкладке доступна только пользователям, входящим в состав диспетчеров заявок либо пользователям с привилегией *Менеджер заявок*.

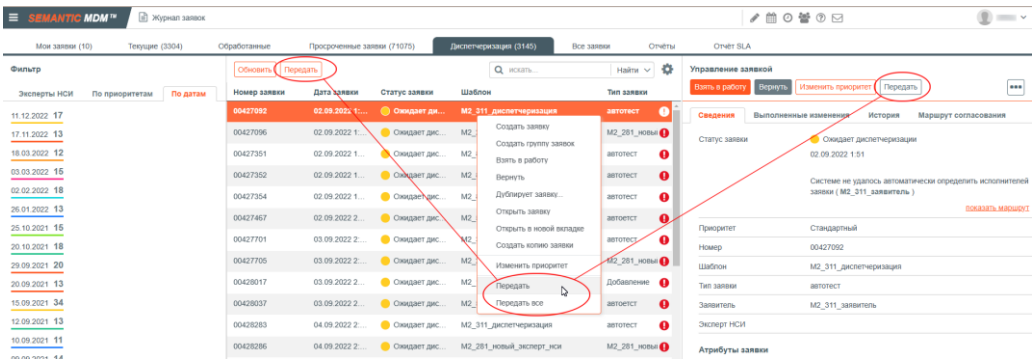


Рисунок 94 - Заявки, ожидающие диспетчеризации

Задачей диспетчера, является передать заявку / заявки исполнителю, используя команду **Передать** / **Передать все** из контекстного меню или кнопкой ***Передать***. Для передачи заявок специалистам требуется:

- на вкладке Диспетчеризация:
  - установить курсор на заявке в таблице;
  - выделить несколько заявок щелчком левой кнопки мыши при зажатой клавише <Ctrl>;
  - выполнить фильтрацию заявок по любым основаниям - приоритетам, датам, заявителям и т.п;
- нажать кнопку **Передать** или выбрать команду **Передать** / **Передать все** в контекстном меню;
- в открывшемся окне **Выберите эксперта НСИ** выбрать нужного специалиста;
- в появившемся окне **Подтверждение** (Рисунок 95) диспетчер может дать необходимые пояснения, ссылки и т.п. и нажать кнопку ***Ок***.

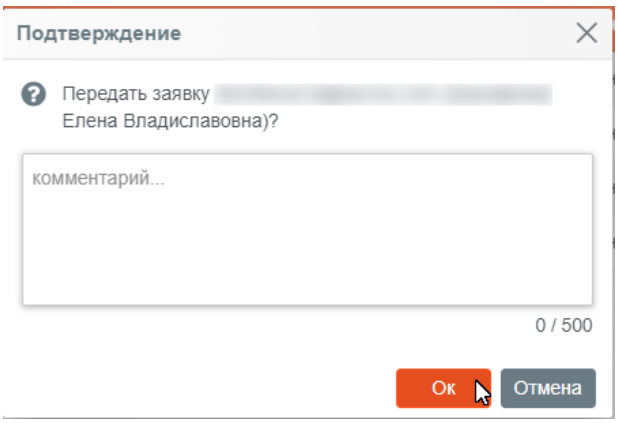


Рисунок 95 - Подтверждение передачи заявки выбранному эксперту НСИ

Диспетчер и Эксперт НСИ могут передать заявку только тем экспертам, которые были назначены как возможные исполнители в шаблоне заявки. Передача заявки любому эксперту НСИ доступна только пользователю с привилегией Менеджер заявок

Если на заявку не были назначены или определены по формуле / условиям (3.12.18) эксперты НСИ, то в этом случае заявке будет установлен статус *Ожидает диспетчеризации* и, в качестве диспетчеров, будут установлены первые 25 пользователей с привилегией *Менеджер заявки*.

Менеджеру заявки в окне **Выберите эксперта НСИ** доступна установка маркера **Показать всех Экспертов НСИ**, которая позволяет передать заявку на исполнение тем экспертам НСИ, которые не были указаны в шаблоне как исполнители (Рисунок 96).

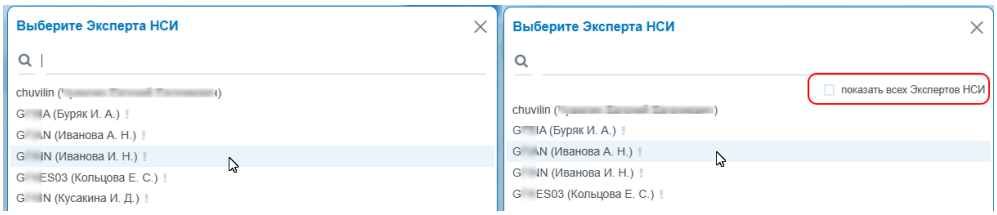


Рисунок 96 - Выбор эксперта НСИ для передачи заявки / заявок: слева окно выбора для Диспетчера / Эксперта НСИ, справа - для *Менеджера заявок*

Переданные заявки будут перемещены на вкладку **Текущие** выбранного эксперта. Кроме того, уведомления будут отправлены на электронную почту исполнителя. Если для одного и того же исполнителя отправляется более одного уведомления по одному и тому же статусу, то они будут сгруппированы в одном письме. Для групповых уведомлений отправляемых на почту, в теле письма будут перечислены статусы заявок и количество заявок с этими статусами, а также ссылки на сами заявки.

Количество заявок, находящихся у каждого эксперта НСИ, также отображается на специальной вкладке слева от таблицы заявок на вкладке **Диспетчеризация** (Рисунок 94). Для каждого эксперта НСИ отображается количество заявок на диспетчеризации, которые он может обработать (первая

цифра) и количество заявок, которые у него в данный момент находятся на вкладке **Текущие** (вторая цифра).

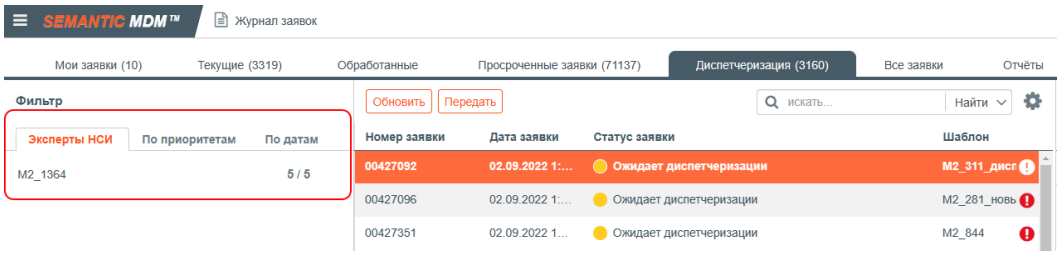


Рисунок 97 - Количество заявок у эксперта НСИ

Кроме передачи заявок исполнителю, диспетчер имеет возможность изменять приоритеты выбранных заявок. Для изменения приоритета нужно выделить необходимые заявки на сетке данных и нажать кнопку Изменить приоритет. В открывшемся окне **Выберите приоритет** (Рисунок 98) выбрать требуемый приоритет и нажать кнопку ***Ок***. Количество заявок, для которых изменяется приоритет, а так же значение устанавливаемого приоритета при диспетчеризации не ограничивается.

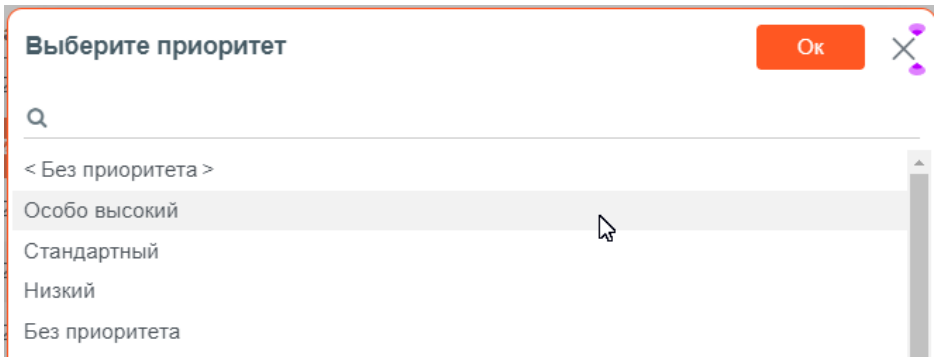


Рисунок 98 - Изменение приоритета заявки при диспетчеризации

3.12.5 Менеджер заявок

Особая привилегия пользователя, которая обеспечивает ему возможность обработки любой заявки в обход настроенных *Экспертов НСИ* и *Согласующих*, настраивать рабочее время для экспертов НСИ и назначать заместителей, диспетчировать заявки, а также дает возможность

просматривать отчёты SLA (подробнее см. п. 3.12.12). Привилегия настраивается системным администратором (см. п. 3.28.1).

Данная привилегия обеспечивает сокращение времени обработки заявок при отсутствии на рабочем месте любого из исполнителей – *Диспетчера, Эксперта НСИ* или *Согласующих*. Менеджер заявок может взять в работу любую заявку с любым статусом.

3.12.6 Работа Эксперта НСИ с заявками

Заявки для обработки экспертом НСИ располагаются на вкладке **Текущие** (Рисунок 99). В скобках рядом с названием вкладки указывается число заявок. На вкладке могут находиться заявки со статусами **Ожидает обработки**, **Ожидает согласования** (в случае, если пользователь выполняет роль согласующего), **Возвращена заявителю**, **В работе**.

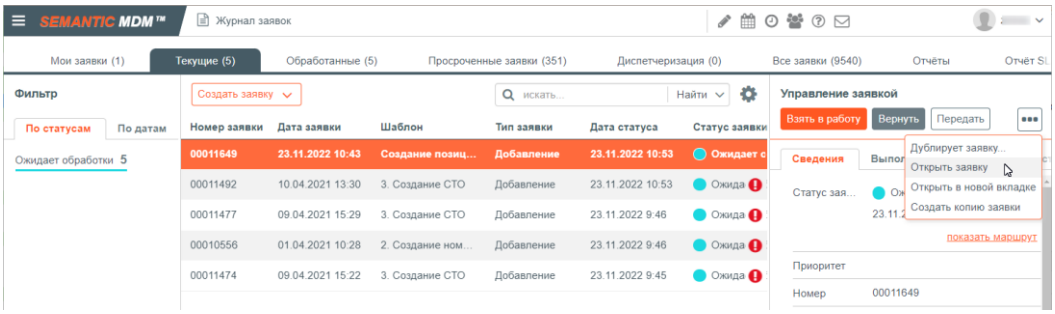


Рисунок 99 - Журнал заявок на вкладке **Текущие**

Кроме вкладки **Текущие** количество заявок, требующих реакции пользователя, отображается в виде специального значка (колокольчик) (Рисунок 100) на панели рабочего окна справочников. Число показывает количество заявок. Щелчок левой кнопкой мыши по пиктограмме открывает вкладку **Текущие** в **Журнале заявок**.

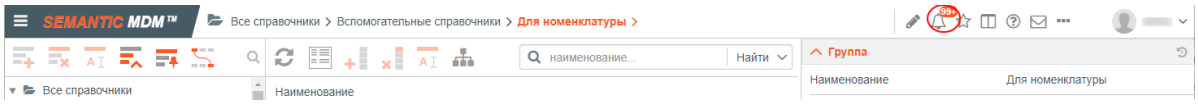


Рисунок 100 - Количество заявок, требующих внимания Эксперта НСИ

Область **Управление заявкой** (Рисунок 101) справа от таблицы имеет свою инструментальную панель для работы с заявками. Часть инструментов скрыта в выпадающем меню кнопки , расположенной в правом верхнем углу области.

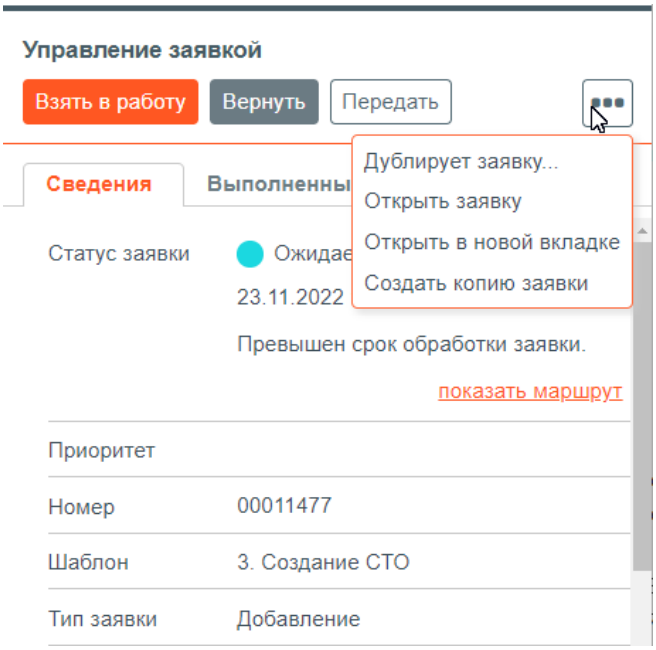


Рисунок 101 - Управление заявками для *Эксперта НСИ*

Кнопка ***Взять в работу*** (Рисунок 102) позволяет взять заявку для её обработки. После нажатия на кнопку необходимо подтвердить, что заявка принимается на обработку нажав на кнопку ***Ок*** или отказаться, нажав кнопку ***Отмена***.

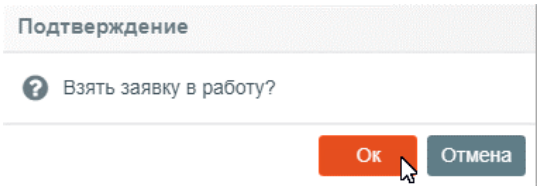


Рисунок 102 - Подтверждение взятия заявки в работу

После взятия заявки в работу, система перейдёт в режим редактирования данных по заявке, то есть создаст пакет изменений (см. раздел 3.13) и откроет справочник, чтобы эксперт НСИ мог выполнить ввод или корректировку данных. При взятии заявки в работу отображаются только заполненные атрибуты позиции в заявке. В открытом справочнике курсор позиционируется

на справочной группе или конкретной позиции, в зависимости от настроек в используемом шаблоне заявки (см. п. 3.12.13).

Окна заявки может быть закреплено между таблицей и карточкой заявки по кнопке ***Закрепить*** **(**Рисунок 103**)**, расположенной в заголовке окна. Выбранный режим работы с заявкой сохраняется индивидуально для каждого пользователя. Повторное нажатие на кнопку ***Закрепить*** позволяет перевести окно заявки в плавающий режим.

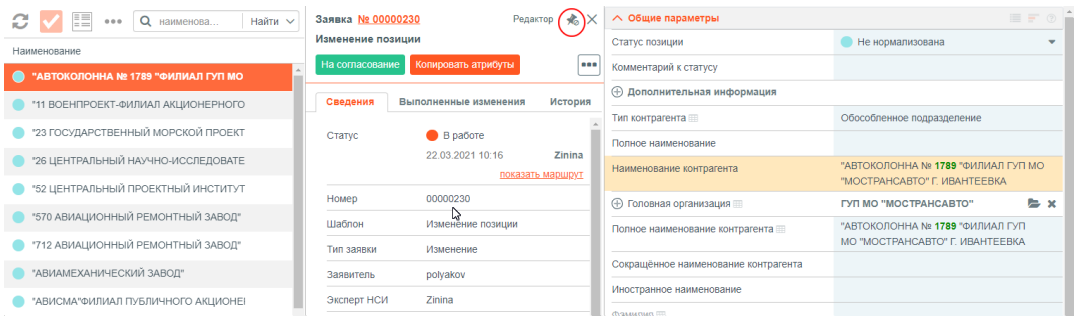


Рисунок 103 - Закрепление окна редактора заявки

В окне редактора *Эксперт НСИ* может выполнить действия, представленные на рисунке ниже.

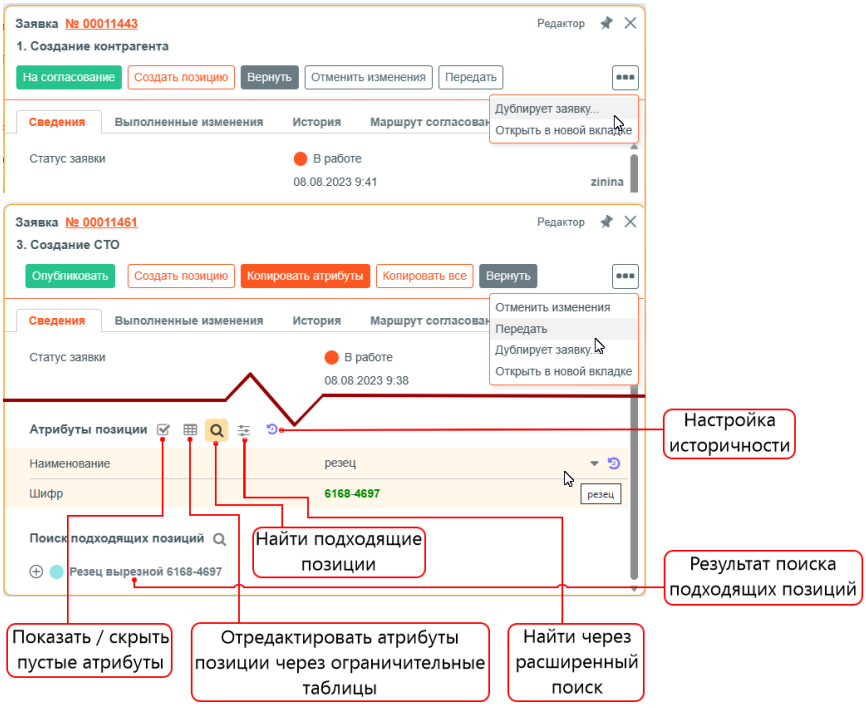


Рисунок 104 - Окно редактора для работы заявкой

- **На согласование** – отправить заявку *Согласующему* для подтверждения возможности обработки. Кнопка доступна только при наличии настроенного маршрута согласования в шаблоне заявки;
- **Опубликовать** – выполняет публикацию поведенных по заявке изменений и перевод заявки в статус **l Обработана**;

При отправке на согласование / публикацию заявки с пакетом изменений, Система проверяет наличие в пакете изменений позиций, не удовлетворяющих обязательным ограничениям, а также наличие позиций со статусом «Не нормализована». При наличии таких позиций, система будет запрашивать подтверждение у пользователя об отправке такого пакета изменений на публикацию / согласование. При положительном ответе пользователя, текст данного подтверждения и положительный ответ на него будет вписан в историю заявки.

- **Создать позицию** - в текущей справочной группе создается новая пустая позиция. Кнопка может быть нажата несколько раз, если в заявке указаны значения для нескольких аналогичных или связанных позиций, например, марка материала, стандарт на материал и номенклатурная позиция. Кнопка ***Создать позицию*** позволяет автоматически создать новую позицию при этом *Система* выполняет следующие действия:
  - Выполняет поиск дубликатов в текущей, выделенной группе;
  - Если через поиск дубликатов ничего не найдено, то создает позицию и копирует в нее значения атрибутов из заявки;
  - Если через поиск дубликатов были найдены позиции, то выдается окно с предупреждением и количеством найденных дублирующих позиций;
  - Если несмотря на предупреждение, пользователь продолжит создание позиции, то позиция будет создана и ее атрибуты будут заполнены из заявки, в противном случае, Система включит в заявке режим поиска позиций через поиск дубликатов;

- **Копировать атрибуты** – кнопка доступна только для заявок на изменение. Вызывает отдельное окно (Рисунок 105), в котором можно выполнить настройки выборочного копирования атрибутов позиции. Кнопка ***Скопировать значение***, расположенная слева от значения атрибута в ячейке, позволяет указать значение атрибута позиции после выполнения заявки. Светло-зеленым цветом выделены значения, которые будут записаны в позицию, после нажатия на кнопку ***Копировать***;

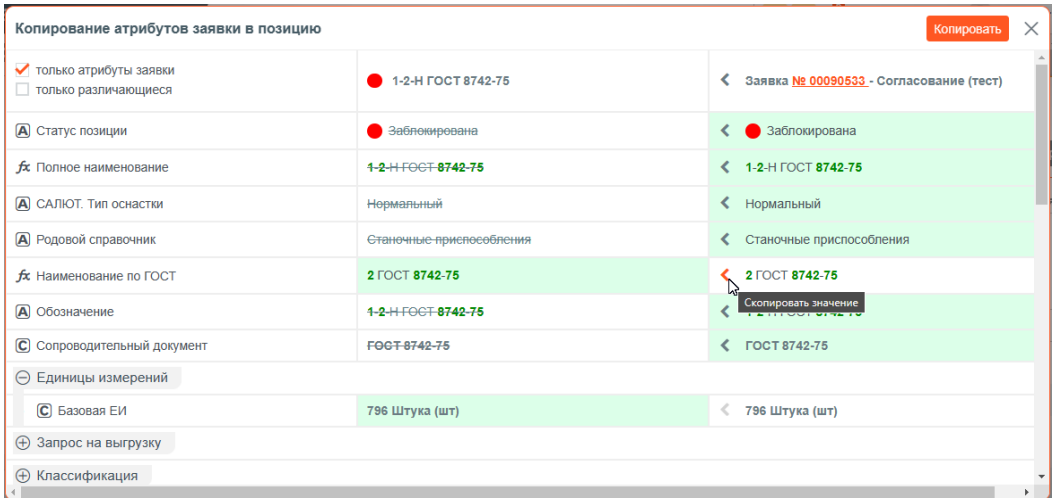


Рисунок 105 - Окно для настройки копирования атрибутов из заявки в позицию справочной группы

- **Копировать все** - значения атрибутов позиции из заявки будут скопированы в соответствующие ячейки позиции (массовое копирование). При копировании данных в карточку позиции можно также воспользоваться копированием значений отдельных атрибутов, для этого в ячейке с наименованием атрибута позиции располагается скрытая кнопка ***Копировать атрибут в позицию***, которая появляется при совмещении указателя мыши с соответствующим атрибутом в заявке (Рисунок 106). В соответствующих атрибутах карточки заявки возможен выбор значения, указанного в заявке, из выпадающего списка;

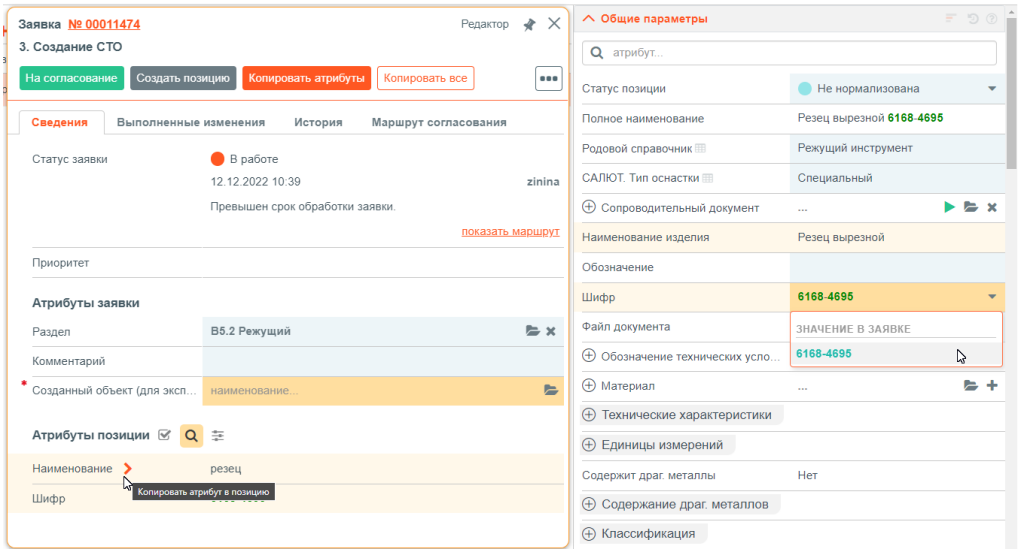


Рисунок 106 – Копирование атрибутов в позицию

- **Вернуть** – вернуть заявку *Заявителю* с указанием причины возврата в специальной форме. Причина возврата может быть указана вручную в текстовом поле окна **Подтверждение** или быть выбрана в виде кода причины возврата (см. п. 3.12.10)(Рисунок 107);

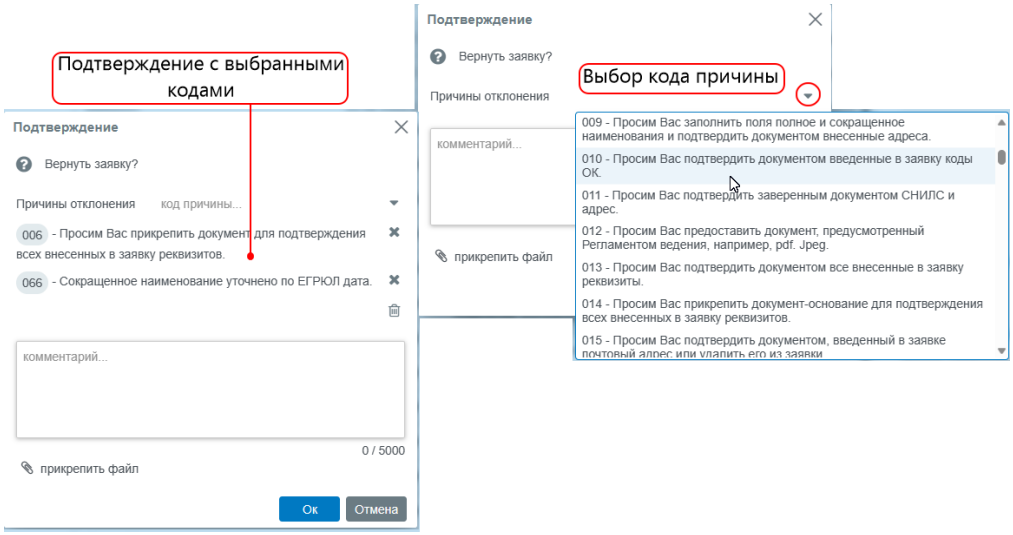


Рисунок 107 - Выбор кода причины возврата заявки

- **Отменить изменения** – внесенные в справочник изменения по заявке будут отменены, окно редактирования закрыто, а заявка получит статус **Ожидает обработки**;
- **Передать** – передать заявку на исполнение другому *Эксперту НСИ*, который выбирается из списка, представленного в окне **Выбор**

Эксперта НСИ. Если экспертов в списке много, то можно произвести поиск, воспользовавшись строкой поиска в верхней части окна выбора. Для передачи заявки нужно щелкнуть левой кнопкой мыши на выбранном эксперте , после чего подтвердить передачу заявки другому эксперту в окне **Подтверждение** нажав на кнопку ***Ок***.

- **Дублирует заявку** – в открывшемся окне заявок позволяет выбрать заявку, которую дублирует текущая. При этом заявка получает статус **Возвращена заявителю** с автоматическим указанием номера заявки-оригинала;
- **Открыть в новой вкладке** – открывает текущую заявку в новой вкладке браузера.
- Для проверки создаваемой позиции, в том числе на наличие дубликатов в *Системе,* в строке **Атрибуты позиции** заявки могут присутствовать кнопки (Таблица 12).

Таблица 12. Перечень кнопок в строке Атрибуты позиции

Кнопка		Функция
	**Показать / скрыть пустые атрибуты**	Скрывает или показывает в заявке строки с незаполненными значениями атрибутов
	**Отредактировать атрибуты позиции через ограничительные таблицы**	Кнопка доступна только при создании / изменении позиции в справочнике, для которого настроены ограничительные таблицы. Вызывает окно с ограничительными таблицами для редактирования значений атрибутов
	**Найти подходящие позиции**	Запускает поиск среди позиций справочника на наличие позиций - дубликатов заявленной. Команда работает только в том случае, если в справочной группе, в которой выполняется поиск, настроен поиск дубликатов. Если вариантов поиска дубликатов в справочной группе несколько, то *Система* последовательно будет проверять наличие дубликатов по всем настроенным вариантам
	**Найти через расширенный поиск**	Запускает панель расширенного поиска, куда переносятся значения атрибутов позиции из заявки
	**Настройка историчности**	Кнопка активирует режим указания исторических значений атрибутов

При необходимости позиция из справочной группы может быть скопирована в атрибут заявки (Рисунок 108). Для этого можно воспользоваться клавиатурными командами <Ctrl> + <C> и <Ctrl> + <V> либо перетащить наименование позиции из таблицы (Drag&Drop) в соответствующую ячейку заявки. При захвате позиции курсор должен быть совмещен с пиктограммой статуса позиции. Перемещение позиции возможно только в атрибуты заявки с типом данных «Ссылка на позицию» или «Ссылка на группу / позицию»

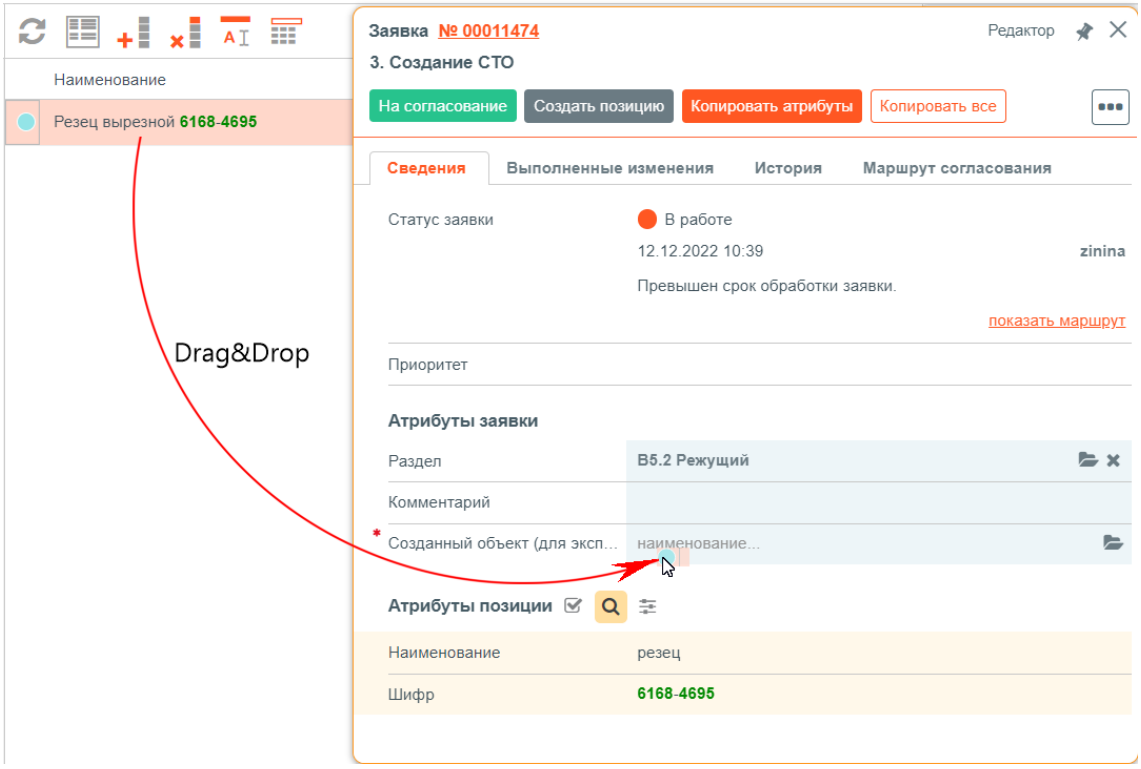


Рисунок 108 - Копирование позиции в атрибуты заявки

Если *Эксперт НСИ* на время прекратил обрабатывать заявку, взятую в работу, то её статус **В работе сохраняется**. При повторном обращении эксперта к этой заявке вид редактора будет как на рисунке ниже.

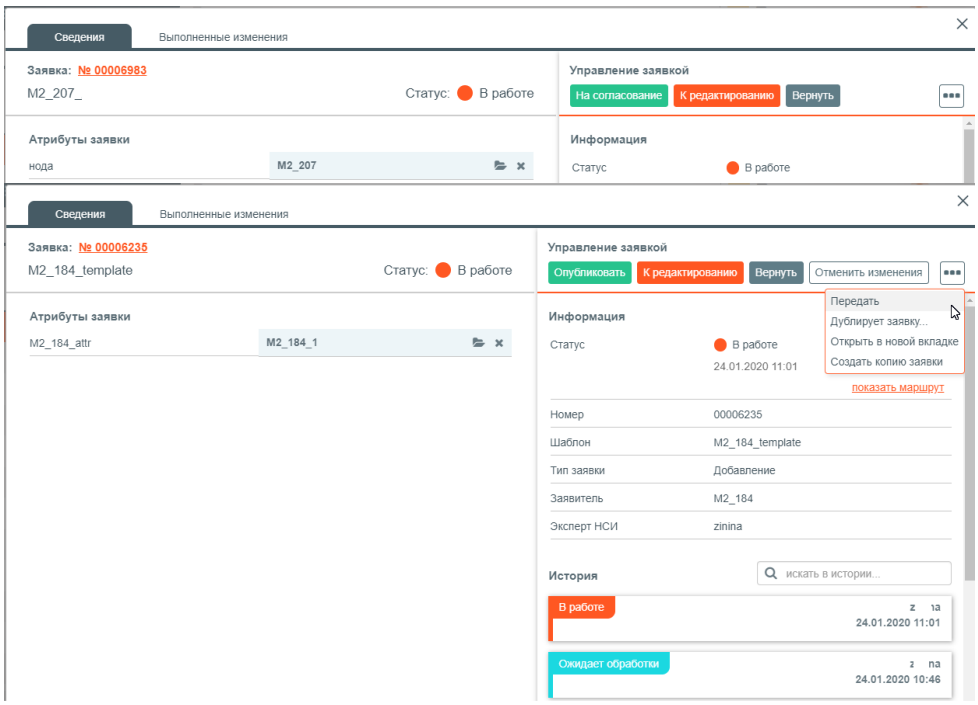


Рисунок 109 - Вид заявки, находящейся в работе

Чтобы вернуться к работе над заявкой следует нажать кнопку ***К редактированию***, подтвердить активацию пакета изменений и продолжить обработку заявки.

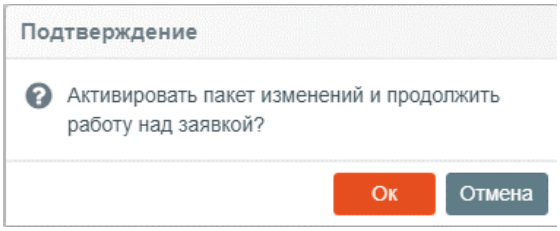


Рисунок 110 - Окно подтверждения активизации пакета изменений

Изменение атрибутов заявки и атрибутов позиции в заявке Экспертом НСИ допускается только, если статус заявки – **Ожидает обработки**, **Ожидает диспетчеризации**, **В работе**, **Возвращена исполнителю**.

Заявки, срок обработки которых был пропущен, перемещаются на вкладку **Просроченные заявки**. Обработка просроченных заявок ведётся также, как и обычных. *Система* предупреждает исполнителя, а также лиц, контролирующих исполнение заявок, о приближении срока исполнения заявки за некоторое время. Предупреждение отправляется на электронную

почту, указанную при регистрации пользователя. Если адрес электронной почты не указан или не доступен, то предупреждения появляются только на вкладке **История** в карточке заявки. Значения сроков исполнения заявок и предупреждения о сроке настраивается в шаблоне заявки или в разделе **Заявки** в карточке группы (см. п. 3.14.18). Сведения о просроченных заявках поступают контролёрам – лицам ответственным за соблюдение сроков исполнения работ, которые назначаются в шаблоне заявки.

По всем событиям, происходящим с заявками, *Эксперт НСИ* получает уведомления от браузера (Рисунок 111). Уведомления могут дублироваться на электронную почту, если это настроено в профиле пользователя.

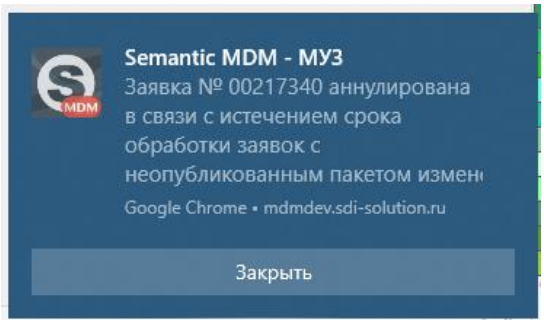


Рисунок 111 - Уведомление браузера об аннулировании заявки

Если заявка была взята в работу, но результат работы не был опубликован до истечения срока обработки заявок с неопубликованным пакетом изменений, то такая заявка получит статус **Аннулирована**. При этом будет отменен пакет изменений, связанный с данной заявкой.

Перед отправкой заявки на согласование или опубликование выполняется автоматическая проверка пакета изменений на отсутствие:

- Нормализованных позиций, не удовлетворяющих обязательным ограничениям;
- Не нормализованных позиций, которые были созданы или изменены в рамках данного пакета изменений.

Система будет выдавать предупреждение пользователю, за исключением случая, когда следующим этапом является этап публикации

пакета изменений, в котором присутствуют нормализованные позиции, не соответствующие обязательным ограничениям. В этом случае, отправка пакета изменений на публикацию будет запрещена и нужно будет устранить проблему в данных.

3.12.7 Работа Согласующего с заявками

Для Согласующего доступны два режима работы, которые определяются настройками в маршруте согласования:

- только согласовывать заявки;
- согласовывать и редактировать записи в справочнике.

Режим работы согласующего зависит от настроек, выполненных в шаблоне заявки.

Заявки, требующие согласования, располагаются на вкладке Текущие.

Кроме вкладки Текущие количество заявок, требующих реагирования пользователя, отображается в виде специального значка (колокольчик) (Рисунок 112) на панели рабочего окна справочников. Число показывает количество заявок. Щелчок левой кнопкой мыши по пиктограмме открывает вкладку Текущие в Журнале заявок.

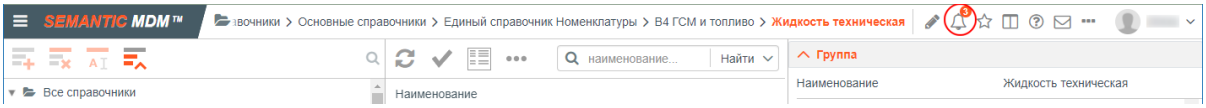


Рисунок 112 - Количество заявок, требующих внимания Согласующего

По всем событиям, происходящим с заявками, Согласующий получает также уведомления от браузера. Уведомления могут дублироваться на электронную почту, если это настроено в профиле пользователя.

При установке курсора на заявке, требующей согласования, на панели Управление заявкой Согласующему становятся доступными следующие функции (Рисунок 113).

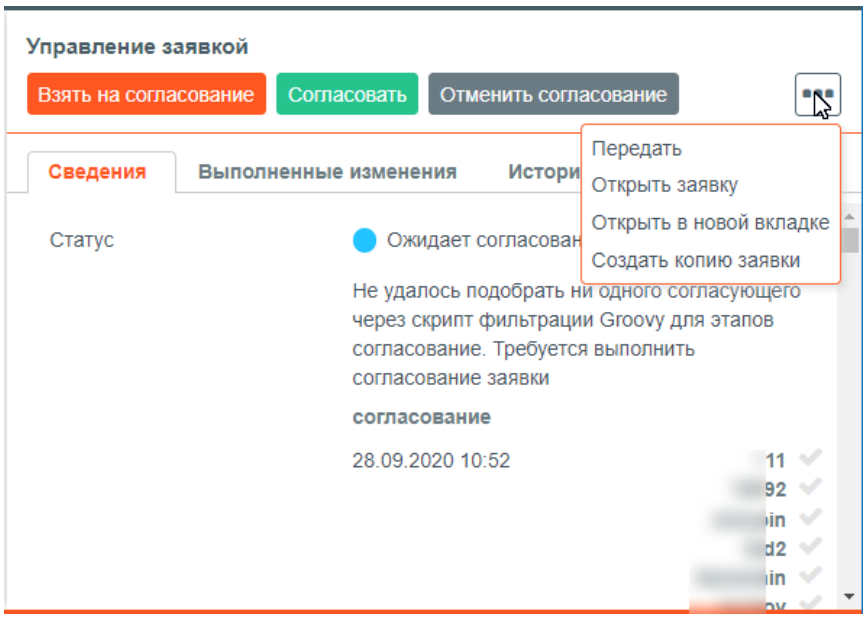


Рисунок 113 - Управление заявкой для Согласующего

- Взять на согласование – позволяет взять заявку на согласование. В специальном всплывающем окне Подтверждение необходимо подтвердить, что текущий пользователь берёт заявку (Рисунок 114);

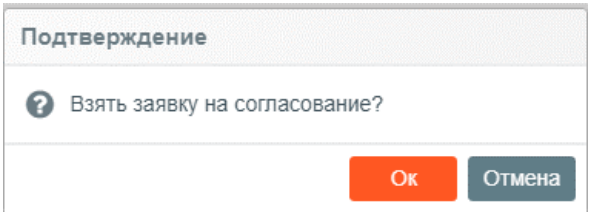


Рисунок 114 - Подтверждение взятия на согласование

- Согласовать – заявка получит статус Обработана и можно будет опубликовать пакет изменений;
- Отменить согласование – позволяет вернуть заявку Эксперту НСИ с указанием причины возврата во всплывающем окне подтверждения. В окне подтверждения отмены согласования может быть прикреплен файл, в случае необходимости. Для этого используется кнопка прикрепить файл (Рисунок 115);

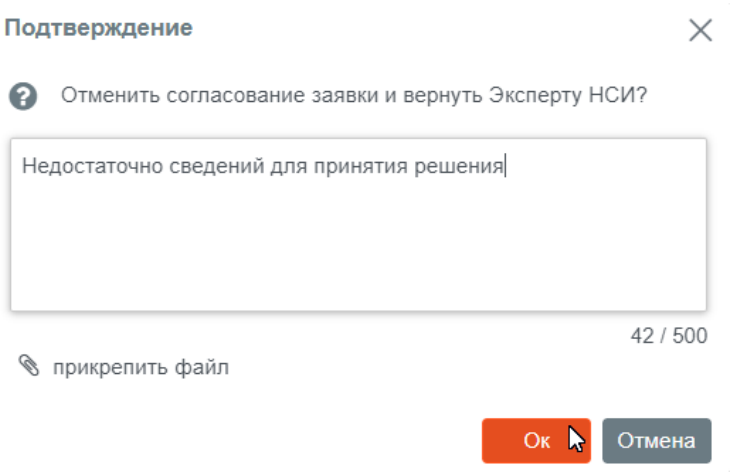


Рисунок 115 - Подтверждение отмены согласования заявки

- Передать - передать заявку на согласование другому специалисту, который выбирается из списка, представленного в окне Выберите согласующего. Если согласующих в списке много, то можно произвести поиск, воспользовавшись строкой поиска в верхней части окна выбора. Для выбора доступны только те пользователи, которые указаны, как Согласующие в шаблоне заявки. Если согласующих в списке много, то можно произвести поиск, воспользовавшись строкой поиска в верхней части окна выбора. Для передачи заявки нужно щелкнуть левой кнопкой мыши на выбранном Согласующем, после чего подтвердить передачу ему заявки в окне Подтверждение нажав на кнопку **Ок**.

Для пользователя с привилегией Менеджер заявок доступна опция «Показать всех согласующих», которая позволяет отобразить всех специалистов, имеющих подобную роль и передать заявку любому из них, в том числе, тому, кто не указан в текущем шаблоне заявки (Рисунок 116);

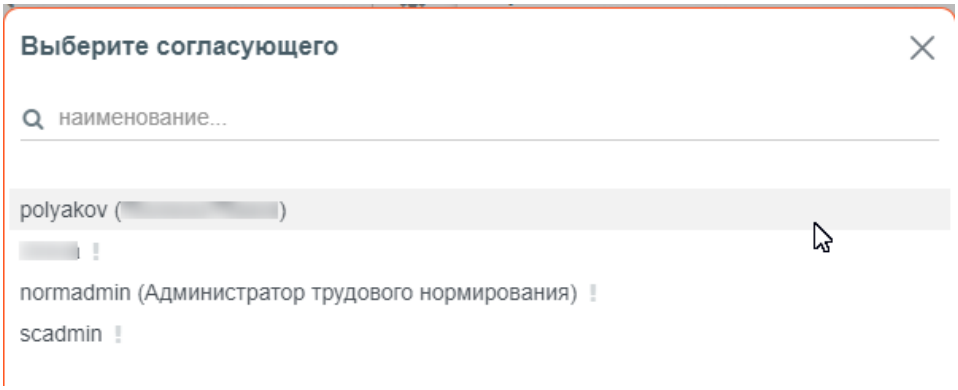


Рисунок 116 - Выбор Согласующего из списка

- ***Открыть заявку*** – открывает заявку во всплывающем окне для подробного рассмотрения содержания;
- ***Открыть в новой вкладке*** – открывает текущую заявку в новой вкладке браузера.

Если Согласующему в настройках шаблона разрешено выполнять редактирование, то после взятия заявки на согласование ему также открывается доступ к справочнику с данными заявки, внесёнными (изменёнными) Экспертом НСИ. Согласующий может внести изменения в нужном ему объёме и лишь затем нажать кнопку Согласовать.

Заявки, срок согласования которых были пропущен, перемещаются на вкладку Просроченные заявки. Обработка просроченных заявок ведётся также, как и обычных. Система предупреждает Согласующего о приближении срока исполнения заявки за некоторое время. Значения сроков согласования заявок и предупреждения о сроке настраивается в шаблоне заявки (подробнее см. п. «Сроки обработки и согласования заявок. Уведомления о сроках»). Сведения о просроченных заявках поступают контролёрам – лицам ответственным за соблюдение сроков исполнения работ, которые назначаются в шаблоне заявки.

3.12.8 Статусы заявок

В зависимости от состояния заявки она принимает различные статусы, сведения о которых отображаются в ячейке Статус заявки в таблице. Слева от

таблицы на панели Фильтр пользователь может видить статусы своих заявок и количество заявок в каждом статусе (Рисунок 117).

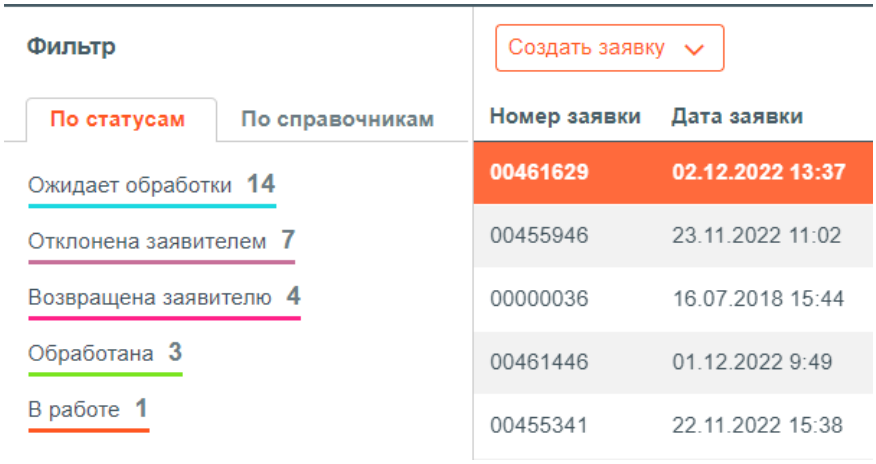


Рисунок 117 - Фильтр заявок по статусами

Последовательность изменения статусов, комментарии Специалистов по ведению НСИ и Согласующих на различных этапах работы с заявкой, название этапов согласования (если их несколько), а также даты и время перехода в новый статус отображаются на вкладке История панели Управление заявкой (Рисунок 118).

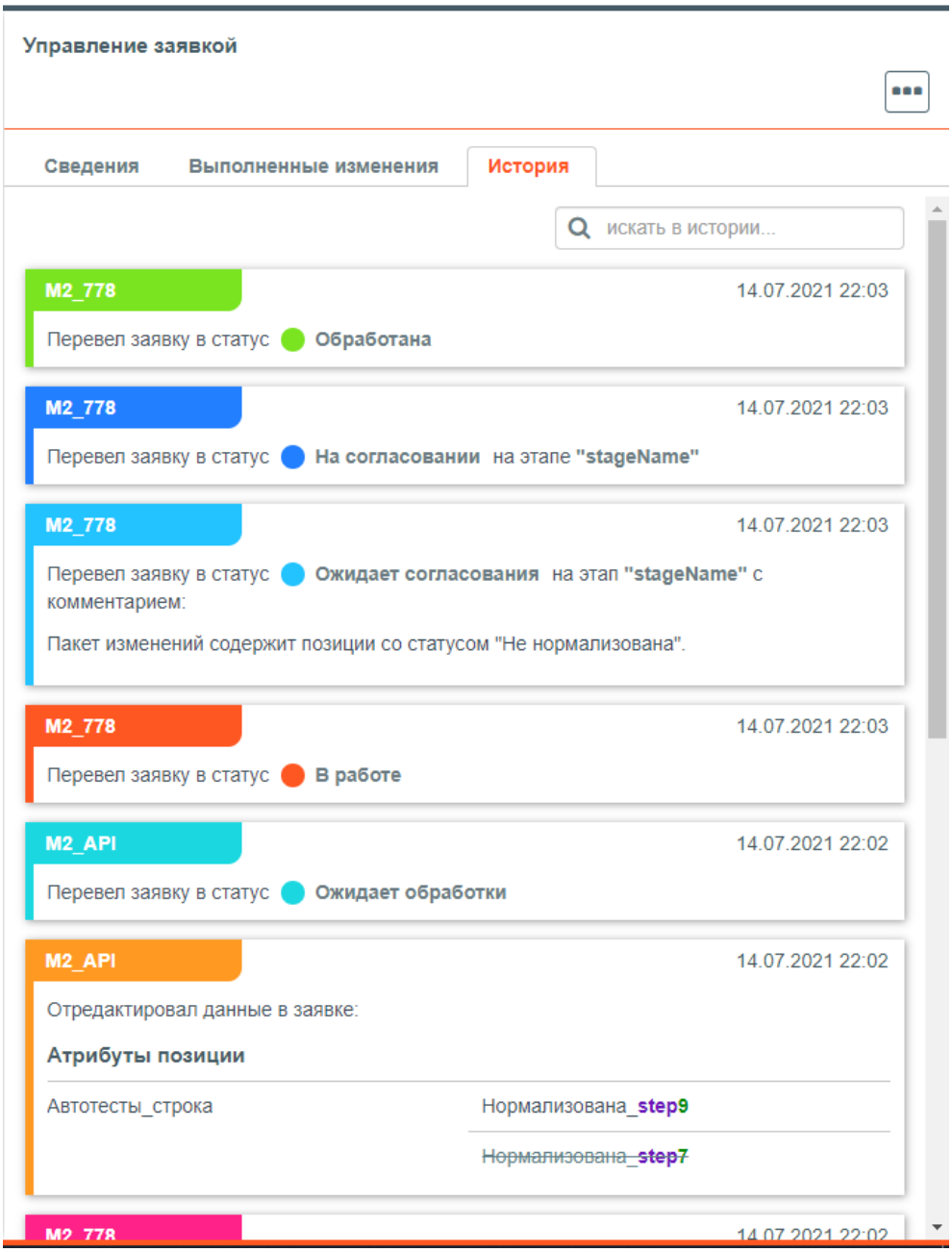


Рисунок 118 - Вид панели «Управление заявкой»

Статусы, которые может принимать заявка, перечислены в таблице 13. Каждый статус сопровождается пиктограммой в виде кружочка определенного цвета.

Таблица 13. Пиктограммы статусов

Пиктограмма	Статус	Комментарий
!	**Зарегистрирована**	Заявка создана *Заявителем*
!	**Ожидает обработки**	Заявка отправлена *Заявителем* на обработку

	Ожидает согласования	Заявка отправлена на согласование. Данный статус всегда дополняется наименованием этапа согласования
	Ожидает диспетчеризации	Для заявки не был автоматически выбран исполнитель и он должен быть назначен *Менеджером заявок*
	На согласовании	Заявка принята *Согласующим* на согласование
	В работе	Заявка принята к исполнению *Специалистом по ведению НСИ*
	Возвращена исполнителю	Заявка возвращена *Специалисту по ведению НСИ* от *Согласующего* или в результате того, что при публикации пакета изменений запись не прошла валидацию
	Возвращена заявителю	Заявка возвращена *Заявителю*
	Аннулирована	Заявка аннулирована в связи с истечением срока обработки заявок с неопубликованным пакетом изменений
	Отклонена заявителем	Заявка отозвана *Заявителем*
	Обработана	Заявка выполнена

На рисунке Рисунок 119 представлена последовательность изменения статусов заявки в процессе её выполнения.

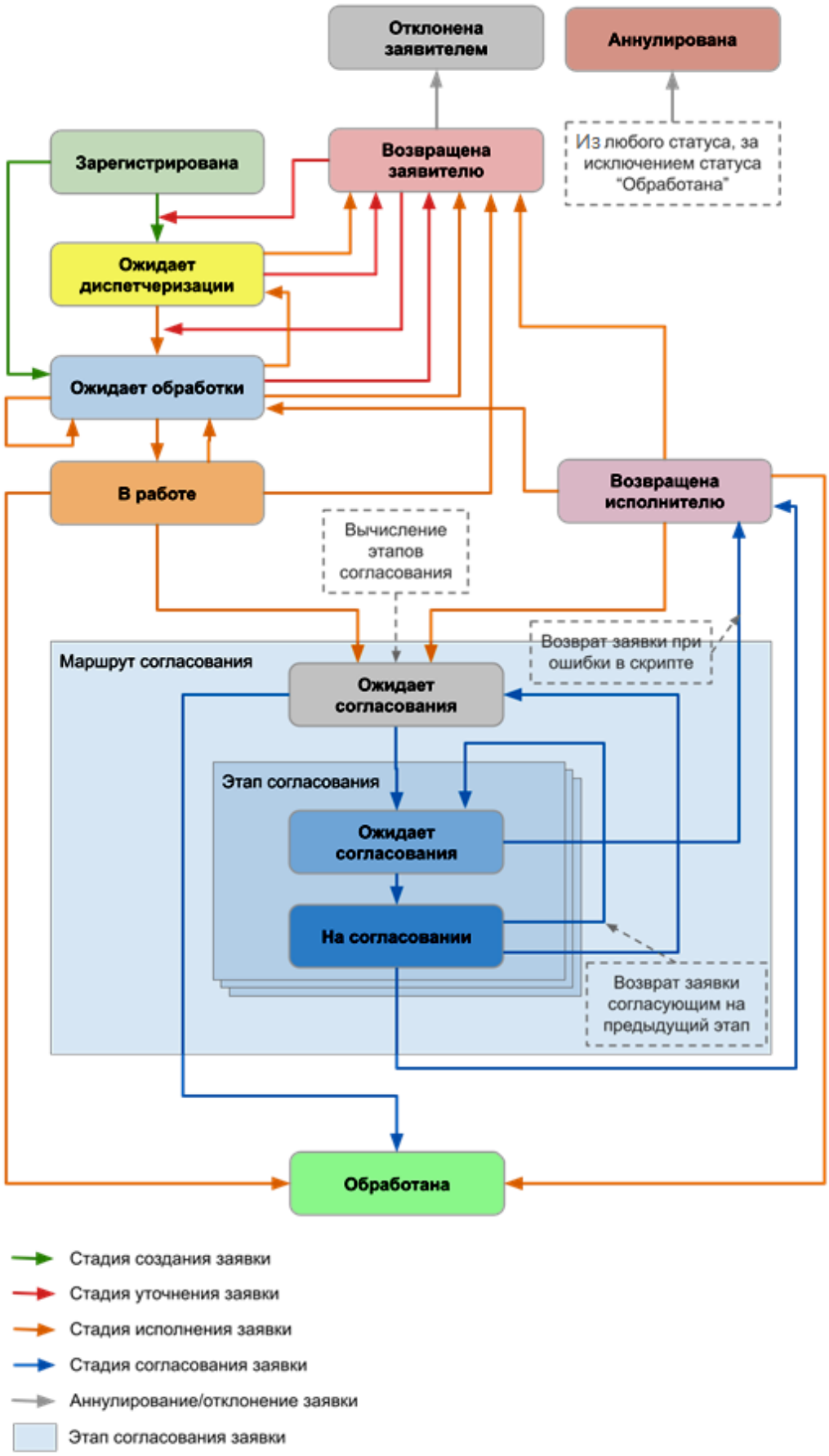


Рисунок 119 - Общая схема статусов заявок

3.12.9 Настройка рабочего расписания

Настройка рабочего времени специалистов, занятых обработкой (диспетчеризацией, исполнением и согласованием) выполняется пользователем с привилегией Менеджер заявок. Настройка рабочего времени необходима для определения сроков исполнения заявок, а также расчета показателей работы специалистов (SLA).

Вызов настройки выполняется кнопкой **Настройка рабочего времени**, расположенной в верхней строке рабочего окна **Журнала заявок** (Рисунок 120).

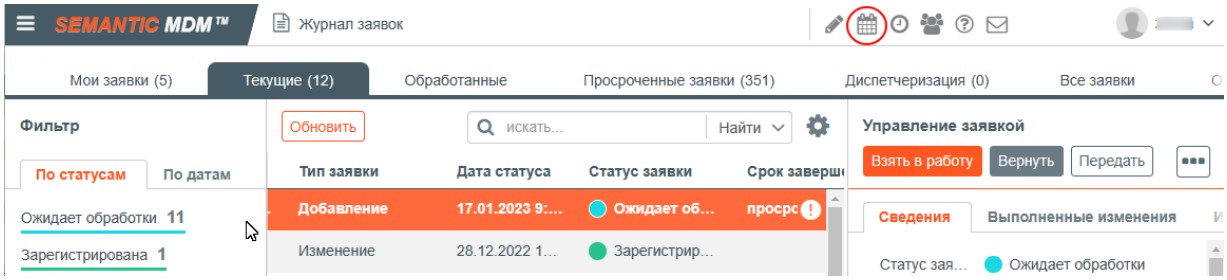


Рисунок 120 - Расположение кнопки Настройка рабочего времени

Кнопка вызывает окно Настройки рабочего времени, где расположен годовой календарь с рабочими (светло-серый цвет) днями. Выходные и праздничные дни отмечены светло-красным цветом. Число текущего рабочего дня отмечено полужирным шрифтом и заключено в кружок. Сокращенные рабочие дни подсвечены светло-оранжевым цветом. Окно имеет две вкладки Общие и Персональные (Рисунок 121).

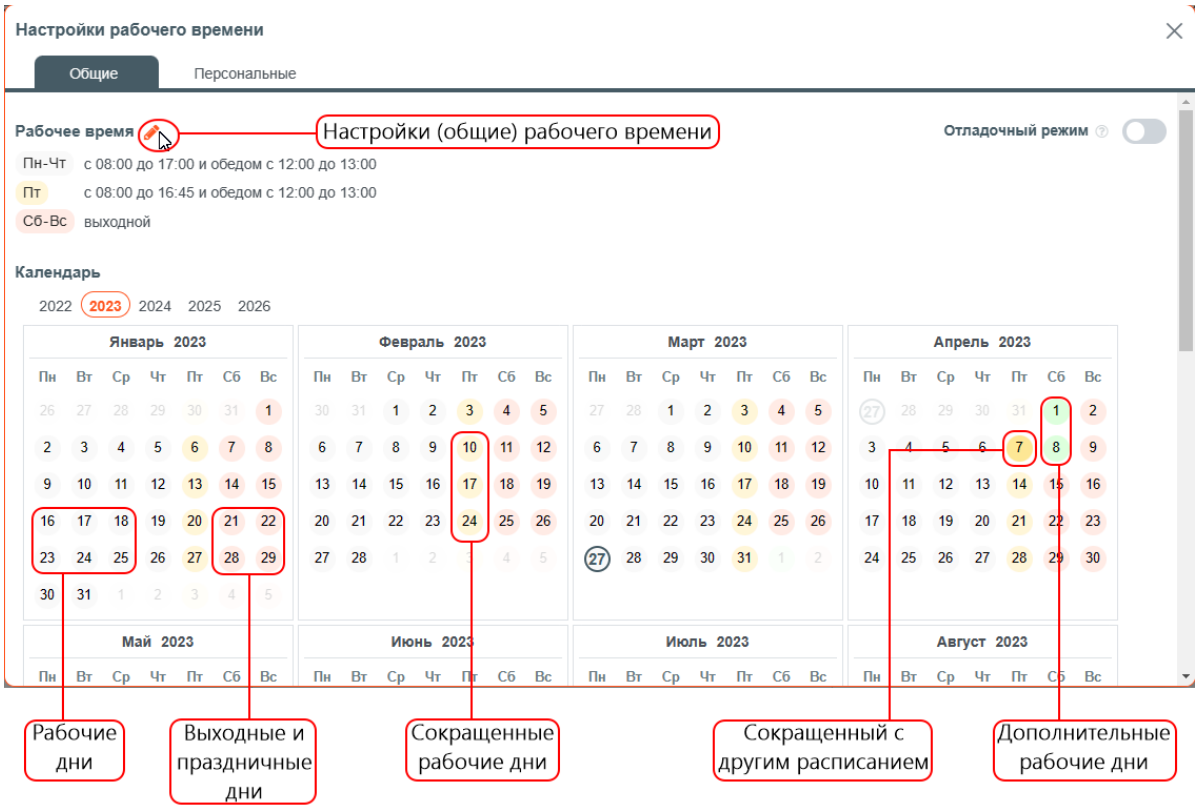


Рисунок 121 - Общий календарь рабочего времени

На вкладке Общие выполняется настройка рабочего времени всех специалистов, работающих с Semantic MDM. На вкладке представлен календарь на текущий, а также прошедший и три последующих года. В календаре на панели Рабочее время представлено текущее расписание, а также расположена кнопка Настройки рабочего времени, которая вызывает диалог настройки начала и окончания рабочего времени и перерывов на обед.

Для того, чтобы настроить общее рабочее время, следует в окне Настройка рабочего времени указать начало и окончание рабочего дня, а также время перерыва на обед. Для настроек по дням недели, необходимо выполнить те же действия в строке требуемого дня недели. После выполнения настройки следует нажать кнопку **Ок** (Рисунок 122).

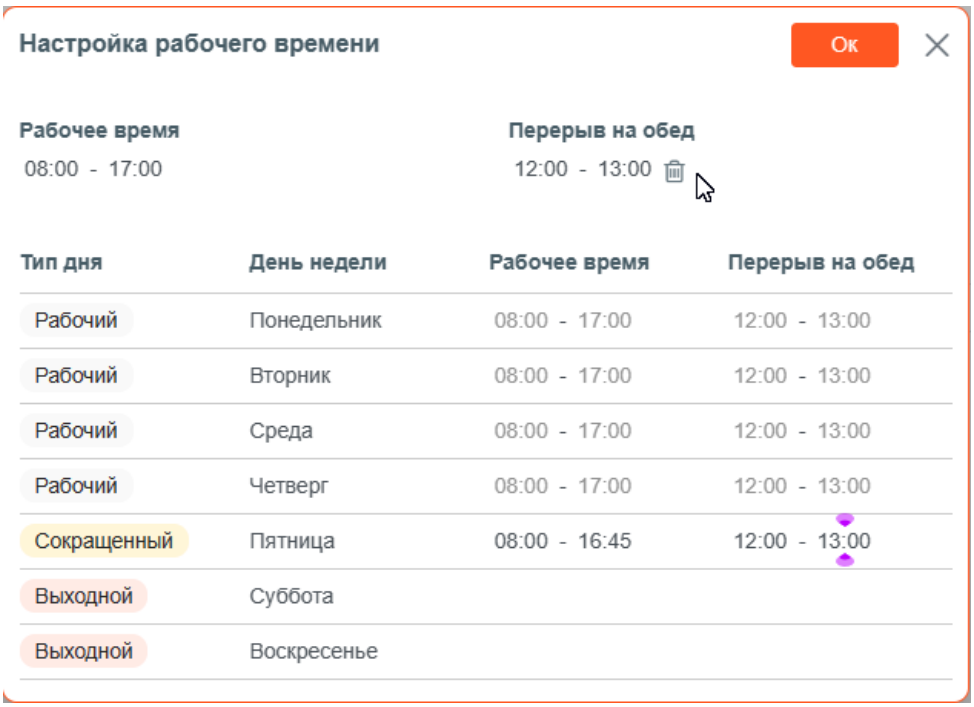


Рисунок 122 - Общая настройка рабочей недели

Для индивидуальных настроек отдельных дней необходимо щелкнуть левой кнопкой мыши на интересующем дне в календаре и указать в диалоге необходимые данные - признак нерабочего дня, рабочее время и перерыв на обед, после чего нажать кнопку **Ок** (Рисунок 123).

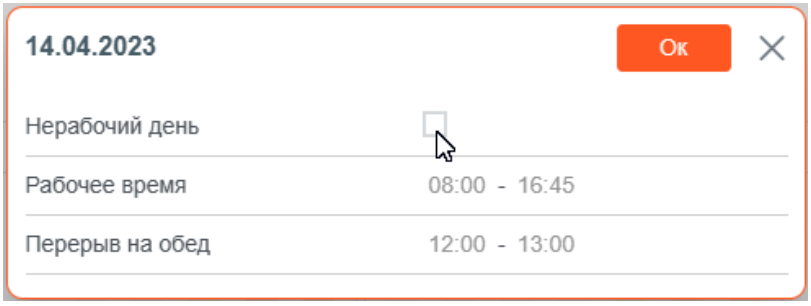


Рисунок 123 - Индивидуальная настройка рабочего или выходного дня

В календаре дни с индивидуальной настройкой также отличаются подсветкой:

- дополнительный выходной день;
- дополнительный рабочий день;
- сокращенный день с отличающимся от остальных сокращенных расписанием.

Вкладка **Персональные** (Рисунок 124) предназначена для настройки производственного календаря отдельных специалистов. Он позволяет учесть отклонения от стандартного графика работы, а также наличие отпуска, больничных листов и другого нерабочего времени. Кроме этого, здесь выполняется настройка заместителя специалиста, который будет выполнять его функции (эксперта, согласующего, контролирующего исполнение) в случае его отсутствия. Задачи отсутствующего специалиста в те дни, которые будут отмечены в его персональном календаре как нерабочие, будут автоматически перенаправлены его заместителю.

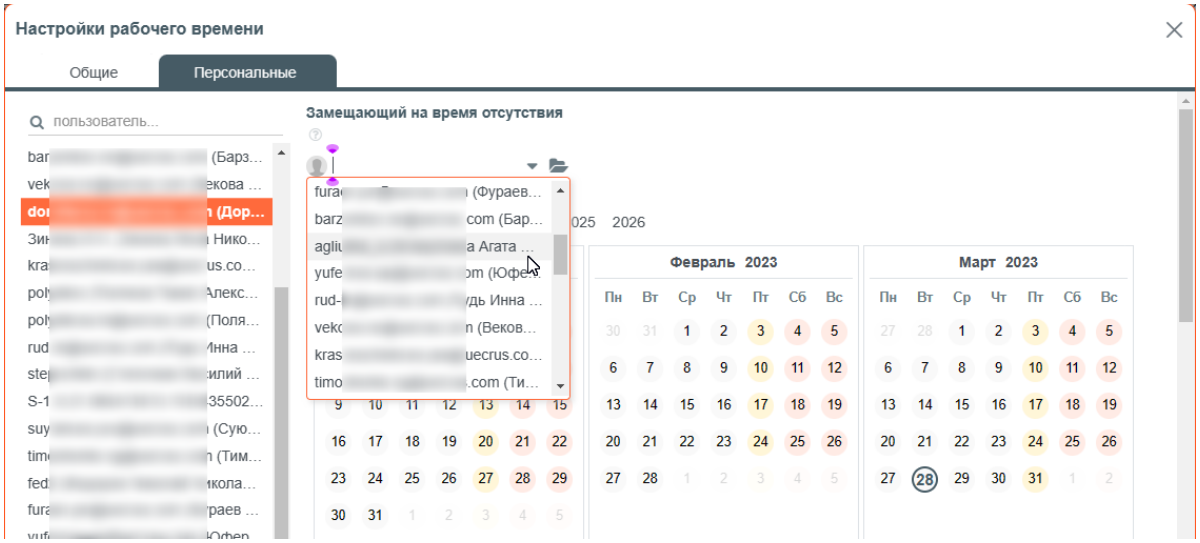


Рисунок 124 - Персональные настройки рабочего времени

Настройки персонального производственного календаря производится путем указания выходных и рабочих дней так, как было описано выше для общих настроек.

Для указания заместителя специалиста необходимо:

- в правой части вкладки **Персональный** найти (можно воспользоваться поиском в верхней части) нужного пользователя и установить на него курсор;
- в ячейке **Замещающий на время отсутствия** выбрать пользователя из выпадающего списка или**,** нажав на кнопку ***Выбрать*** , найти его в окне **Заместители (**Рисунок 125**)** в списке пользователей и нажать кнопку ***Ок***.

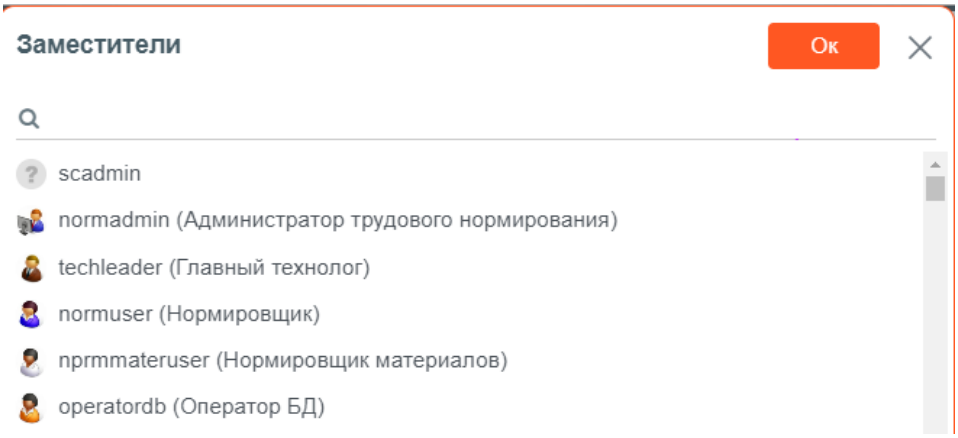


Рисунок 125 - Выбор заместителя из списка

3.12.10 Настройка причины возврата заявок

В Системе возможно ведение справочника типовых причин возврата заявок заявителю. Подобный справочник упрощает указание причин, по которым заявка возвращена Заявителю и значительно снижает трудоемкость и повышает качество вводимого в качестве пояснения текста. Кроме этого, использование подобных кодов позволяет проводить аналитику причин, по которым заявки не принимаются в обработку.

Для вызова справочника на основной панели подсистемы Журнал заявок расположена кнопка Настройка причины возврата заявки (Рисунок 126). По этой кнопке открывается окно Причины возврата заявок для редактирования типовых причин.

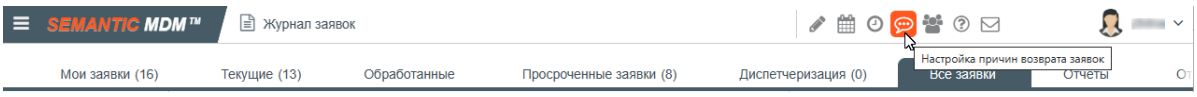


Рисунок 126 - Кнопка для перехода к настройке причин возврата заявок

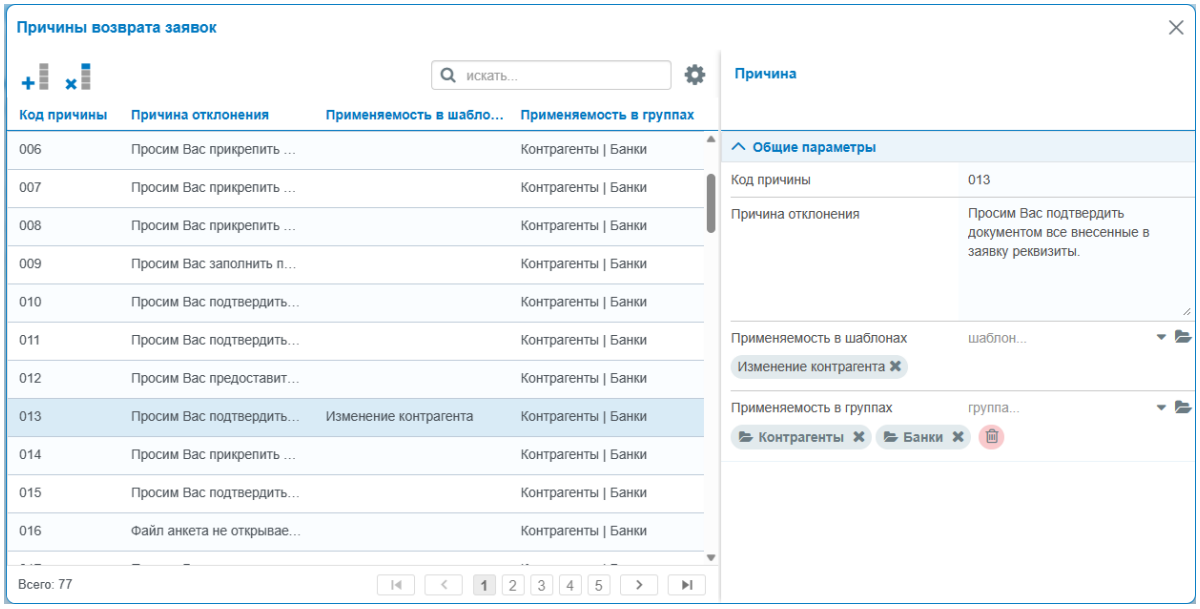


Рисунок 127 - Кнопка доступа к настройке и окно ввода причин возврата заявок

В левой части окна (Рисунок 127) отображена таблица со следующими колонками:

- Код причины - трехзначный цифровой код, который генерируется автоматически при добавлении новой причины. Данный код может быть скорректирован пользователем, но должен оставаться уникальным. Если вводится неуникальный код, то Система сообщит об ошибке: Введенный код уже существует, введите другой код причины (Рисунок 128);

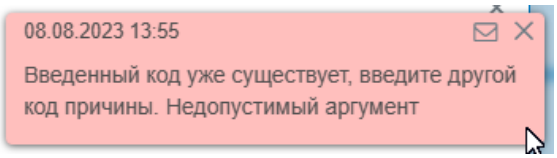


Рисунок 128 - Сообщение о неуникальности введенного кода

- Причина отклонения – текст причины отклонения;
- Применяемость в шаблонах – наименование шаблонов заявок, для которых должна быть доступна данная причина. Если шаблон не

указан, то данная причина отклонения будет доступна для всех шаблонов заявок;

- Применяемость в группах – справочные группы, для которых должна доступна данная причина. Справочная группа в заявке определяется по атрибуту заявки, для которого установлена опция Заполнить при создании заявки из справочника. Если справочная группа не указана, то данная причина отклонения доступна для всех справочных групп.

Над таблицей располагаются кнопки Добавить причину и Удалить. Чтобы добавить новую причину возврата следует нажать на кнопку Добавить причину и заполнить в карточке справа текст причины отклонения, выбрать из списка шаблоны заявок, а также справочные группы, на которые распространяется эта причина.

При возврате заявки заявителю на статус Возвращена заявителю в окне ввода комментария к статусу отображается дополнительное поле в котором можно выбрать одну или несколько причин отклонения из выпадающего списка, при этом поле комментария к статусу также доступно для ввода.

При просмотре возвращенной заявки заявителем в карточке заявки на вкладке История, под статусом заявки, после комментария указаны коды причин отклонения заявки с их описанием (Рисунок 129).

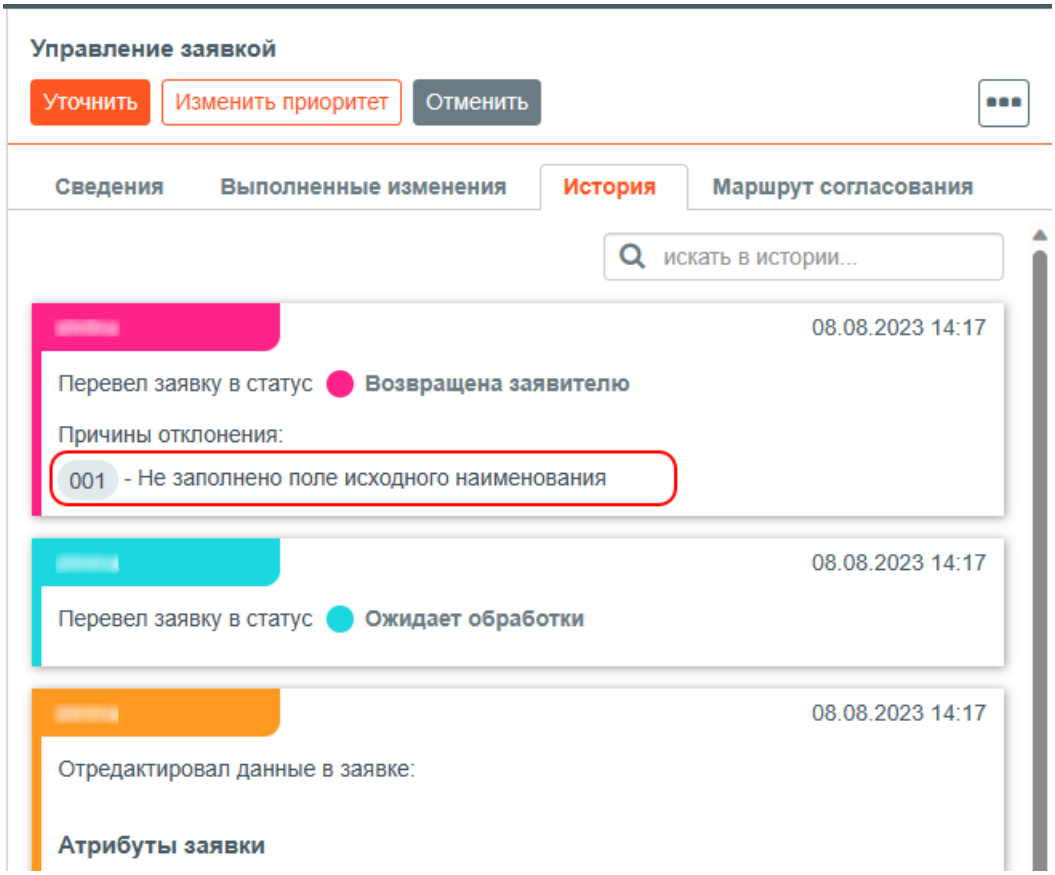


Рисунок 129 – Сохранение кода и причины возврата в истории заявки

3.12.11 Приоритеты

Приоритеты заявок предназначены для определения степени срочности и регламентного времени обработки поступивших заявок.

Приоритеты настраиваются через специальный диалог пользователем с привилегией *Менеджер заявок*. Окно с настройками вызывается кнопкой ***Настройка приоритетов***, расположенной в рабочем окне Журнала заявок или в рабочем окне подсистемы Конфигурация данных.

На вкладке **Приоритеты обработки / согласования заявок** (Рисунок 130) отображается таблица с приоритетами, где первая строка таблицы соответствует наивысшему приоритету, а последняя строка – наименьшему. В таблице предусмотрены следующие параметры приоритета:

- **Наименование** – строковое обозначение приоритета в таблице заявок;
- **Описание** – текстовое описание приоритета;

- **Контроль сроков** – указание «Да» или «Нет» определяет деятельность контролирующих исполнение / согласование пользователей, для заявок с данным приоритетом. Если указано «Нет», то контроль сроков для заявок с этим приоритетом не выполняется даже при наличии контролирующих лиц в шаблоне заявки.

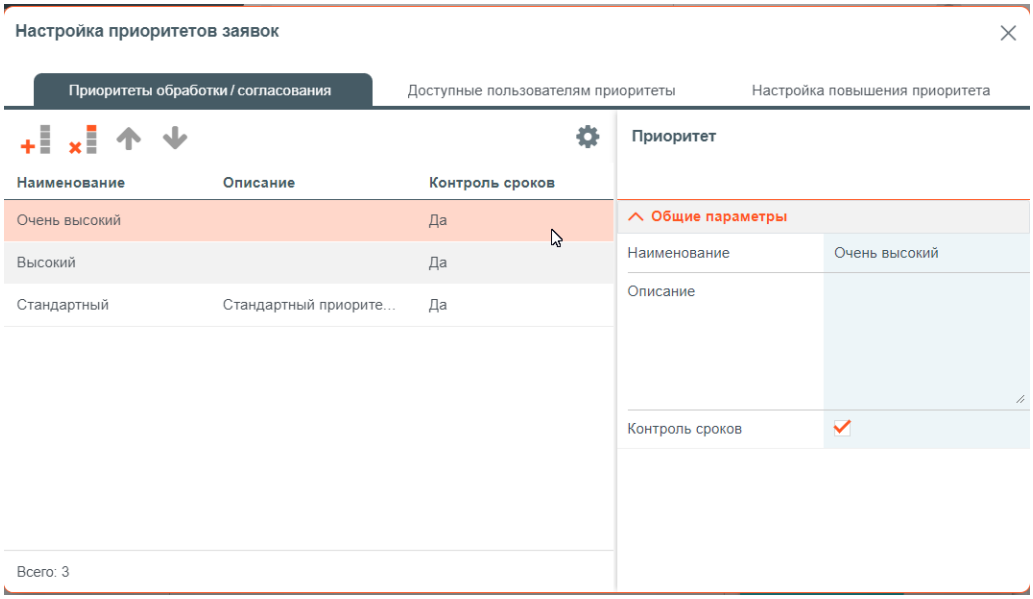
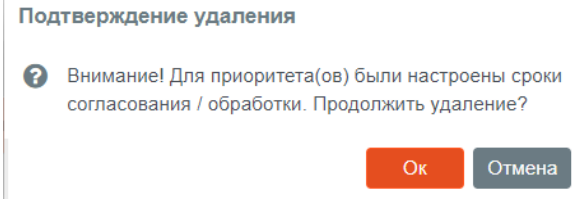


Рисунок 130 – Настройка приоритетов заявок в Semantic MDM

Над таблицей приоритетов доступны кнопки для выполнения настройки приоритетов, представленные в таблице ниже.

Таблица 14. Кнопки для настройки приоритета заявки

Кнопка		Функция
	**Добавить приоритет**	Добавляет приоритет в конец таблицы
	**Удалить**	Удаляет выбранный приоритет из таблицы. Если в Системе уже есть заявки с данным приоритетом или есть настроенные сроки для этого приоритета, то перед удалением выдается предупреждение об удалении приоритета:  Если пользователь подтвердит удаление, то настройки сроков для данного приоритета будут удалены, а в существующих заявках

		этот приоритет остается с присвоенным обозначением, порядковым номером и сроками.
	**Переместить выше**	Перемещает выделенный приоритет выше в таблице
	**Переместить ниже**	Перемещает выделенный приоритет ниже в таблице

Наименование, описание и контроль сроков указываются при настройке в карточке приоритета справа от таблицы.

По умолчанию в таблице присутствует единственный приоритет, со следующими настройками:

- **Наименование** – *Стандартный*;
- **Описание** – *Стандартный приоритет обработки заявки*;
- **Контроль сроков** – *Да*.

Возможно удаление всех приоритетов, в этом случае все заявки создаются без приоритета и контроля сроков обработки / согласования.

Вкладка **Доступные пользователям приоритеты** (Рисунок 131) предназначена для настройки доступа пользователей и групп пользователей к приоритетам в зависимости от справочника.

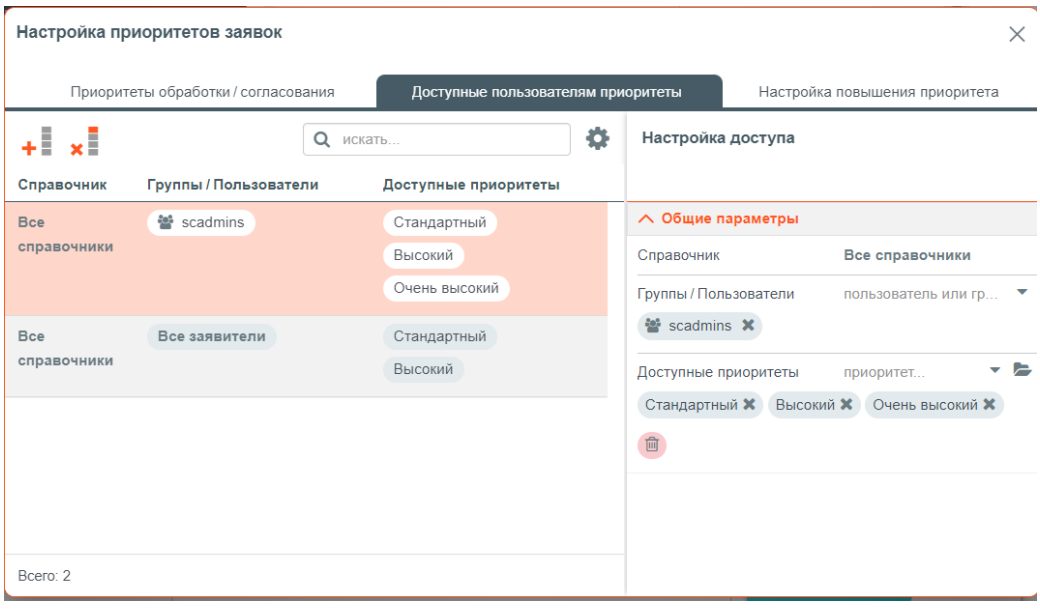


Рисунок 131 – Доступные пользователям приоритеты

Таблица на вкладке содержит следующие столбцы:

- **Справочник** – в столбце указывается справочник, для которого настраиваются доступные приоритеты. Справочник выбирается из списка зарегистрированных (см. п. 3.14.3) в *Системе*. В списке справочников предусмотрен системный справочник «Все справочники»;
- **Группы / Пользователи** – в столбце указывается один или несколько пользователей / групп пользователей, которым назначаются доступные приоритеты. В списке групп пользователей всегда присутствует системная группа «*Все заявители»*, при выборе которой все остальные значения в этой ячейке удаляются;
- **Доступные приоритеты** – в столбце указывается один или несколько зарегистрированных приоритетов, которые доступны для заявок пользователей, указанных в столбце **Группы / Пользователи** для справочника, указанного в столбце **Справочник**.

Для добавления новой настройки доступа используется кнопка ***Добавить***, а для удаления существующего – ***Удалить***, расположенные над таблицей слева. Выбор значений для настройки выполняется в карточке справа от таблицы.

Все записи в таблице сортируются сначала по колонке **Справочник**, при этом строки со значением «*Все справочники*» отображаются первыми, а затем выполняется сортировка справочников по наименованию. В таблице разрешен быстрый поиск по названию справочника, наименованию пользователя или группы, названию приоритета. Отображение колонок в таблице настраивается по кнопке  ***Настройки таблицы***.

Если таблица доступа пользователей к приоритетам пустая, то заявки создаются без приоритета, а приоритет у заявки может поменять только пользователь с привилегией *Менеджер заявок* или диспетчер заявок на этапе диспетчеризации.

Повышение приоритета заявки требуется для ускорения процедуру ее обработки. Каждому заявителю можно определить разрешенное количество повышений приоритета заявки в сутки в пределах приоритетов, настроенных на вкладке **Доступные пользователям приоритеты**. Повышение приоритета заявителем допускается для заявок в статусах «Зарегистрирована» и «Ожидает диспетчеризации». Настройка разрешенного числа повышений приоритета выполняется на вкладке **Настройка повышения приоритета**.

На вкладке **Настройка повышения приоритета** (Рисунок 132) отображается таблица для настройки разрешённого числа раз повышения приоритета в сутки для каждого пользователя со следующими столбцами:

- **Пользователь** – в столбце выбирается пользователь, для которого будет настраиваться количество разрешенных повышений за сутки;
- **Количество повышений** – в столбце настраивается разрешенное количество раз повышения приоритета в сутки. Можно ввести число или выбрать значение «*Не ограничено*», устанавливаемое кнопкой « ∞ », расположенной в той же строке справа.

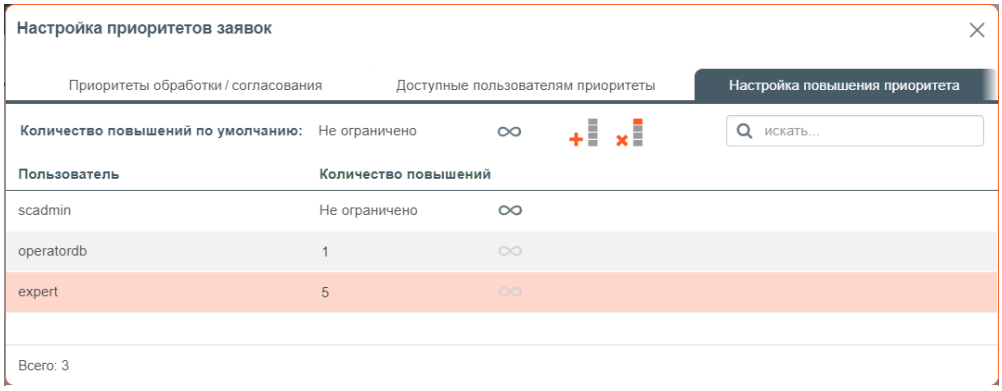


Рисунок 132 – Настройка повышения приоритетов

Над таблицей отображается поле **Количество повышений по умолчанию** для ввода количества повышений по умолчанию для всех пользователей.

3.12.12 Отчёт SLA

SLA (Service Level Agreement) – это соглашение об уровне предоставляемого сервиса. Содержание SLA со стороны сервисного отдела — это набор целевых метрик, к которым стремятся исполнители. Поэтому соблюдение SLA является одним из основных ключевых показателей эффективности (KPI). Для оценки эффективности работы экспертов НСИ по выполнению заявок в *Semantic MDM* присутствуют отчеты SLA.

Доступ к отчетам SLA возможен только в подсистеме **Журнал заявок** и разрешен только для пользователей с привилегией *Менеджер заявок.* Отчёты располагаются на отдельной вкладке (Рисунок 133).

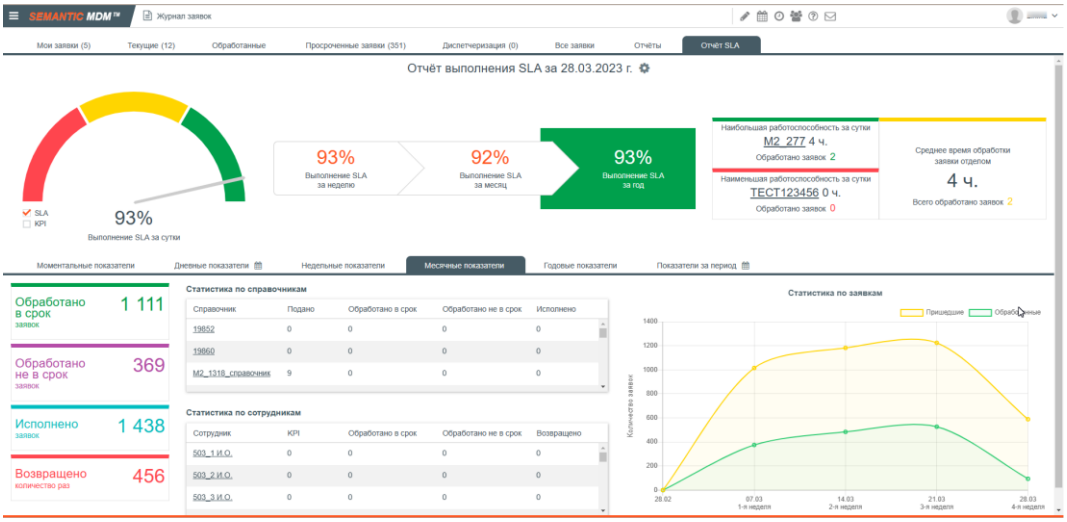


Рисунок 133 – Внешний вид вкладки Отчёты SLA

Данная вкладка имеет персонализированные настройки под каждого пользователя (Рисунок 134). Настройки доступны по кнопке , расположенной рядом с заголовком отчета.

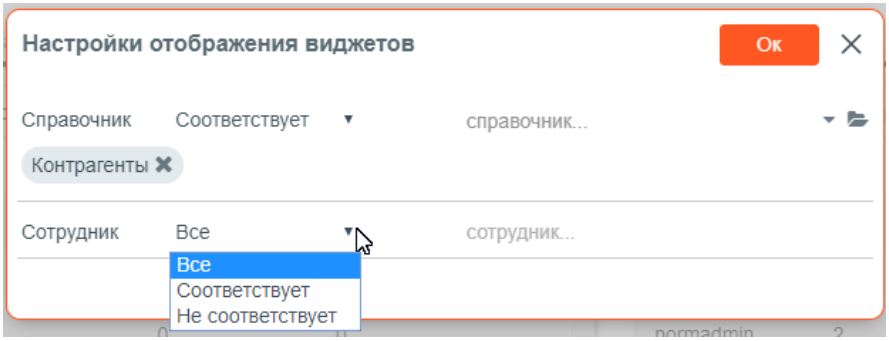


Рисунок 134 – Настройка отображения отчётов SLA

- **Справочник** – поле для выбора справочника (справочников), по которым требуется представлять отчет. Справочник выбирается из выпадающего списка;
- **Сотрудник** – имя пользователя (пользователей), по которым требуется отображать аналитику. Имя выбирается из выпадающего списка.

Условие включения в отчет выбирается из выпадающего списка:

- **Все** – отчет включает в себя данные по всем справочникам и / или пользователям системы;
- **Соответствует** – в отчет включаются только данные по указанным справочникам и / или пользователям;
- **Не соответствует** – в отчет включаются все данные, кроме данных по указанным справочникам и / или пользователям.

В верхней части окна располагаются общие статистические данные, которые могут быть представлены как в виде показателей SLA, так и KPI. Переключатель представления расположен под изображением шкалы эффективности справа. Общие данные включают в себя:

- Выполнение SLA (KPI) за неделю – процентный показатель;
- Выполнение SLA (KPI) за месяц – процентный показатель;
- Выполнение SLA (KPI) за год - – процентный показатель.

Процент выполнения KPI, % - для каждого сотрудника рассчитывается сумма KPI по каждому из выбранных справочников, где: KPI для справочника = (количество обработанных заявок сотрудником по справочнику за заданный период / (норма обработки справочника * (количество рабочих дней в

заданном периоде – нерабочие дни сотрудника)))*100.

SLA отдела – это итоговый параметр по работе отдела. Расчет показателя выполняется по формуле: 100 – (количество заявок обработанных сотрудником с просрочкой / количество обработанных сотрудником заявок)*100.

- **Наибольшая работоспособность за сутки** – отношение количества выполненных заявок к величине отработанного времени для наиболее эффективного сотрудника, с указанием его логина в системе;
- **Наименьшая работоспособность за сутки** – отношение количества выполненных заявок к величине отработанного времени для наименее эффективного сотрудника, с указанием его логина в системе;
- **Среднее время обработки заявки отделом** – отношение количества выполненных заявок к величине затраченного времени.

Ниже общих статистических данных располагаются четыре вкладки с показателями, отражающими эффективность работы в зависимости от периода времени:

- **Моментальные показатели** – для анализа используются заявки с изменениями за последние прошедшие 24 часа, начиная с предыдущего часа. Например, если сейчас 9:10, то берется период с 9:00 предыдущего дня до 9:00 текущего дня, при условии что предыдущий день является рабочим. Если предыдущий день не является рабочим, то к анализируемому периоду добавляется по 24 часа, за каждый предшествующий нерабочий день. Например, если сейчас понедельник 9:10, а последний рабочий день это пятница, то берется период 72 часа, а именно, с пятницы 9:00 до понедельника 9:00;
- **Дневные показатели** (Рисунок 135)– для анализа используются статистические данные по обработке заявок за текущие сутки с разбивкой по часам. Дата выбирается из выпадающего календаря;

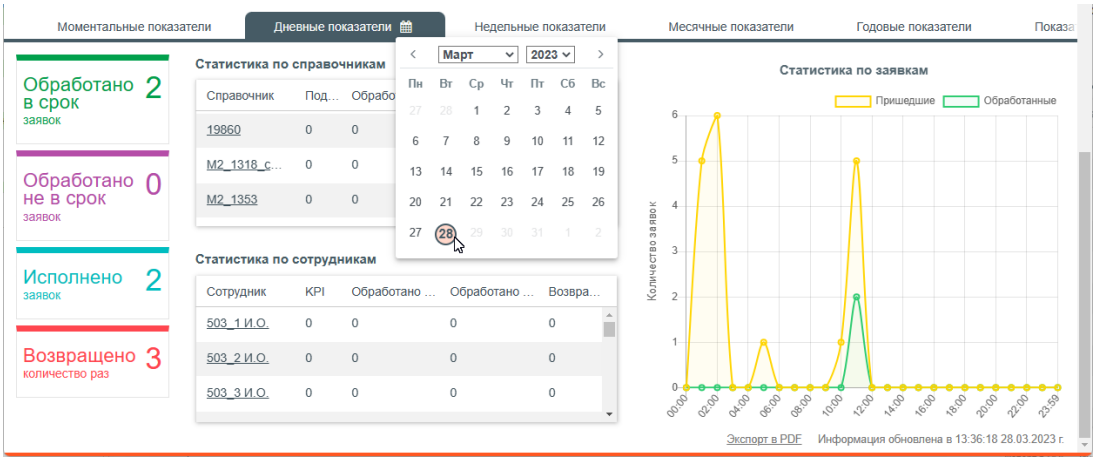


Рисунок 135 – Дневные показатели обработки заявок

- **Недельные показатели** – для анализа используются заявки с изменениями за последние 7 календарных дней, начиная с предыдущего дня;
- **Месячные показатели** – для анализа используются заявки с нужными изменениями за последние 30 дней, начиная с предыдущего дня;
- **Годовые показатели** – для анализа используются заявки с нужными изменениями за последние 365 дней, начиная с предыдущего дня.
- **Показатели за период (**Рисунок 136**)** – для анализа используются статистические данные по обработке заявок за выбранные период времени. В зависимости от выбранного периода, по оси X откладываются: *Часы, Дни, Недели, Месяцы, Кварталы*. Выбираемый период не может превышать 4 года и устанавливается при помощи всплывающего окна для настройки интервала начала и конца периода. Интервал может быть установлен при помощи подвижных пределов или с помощью выпадающего календаря.

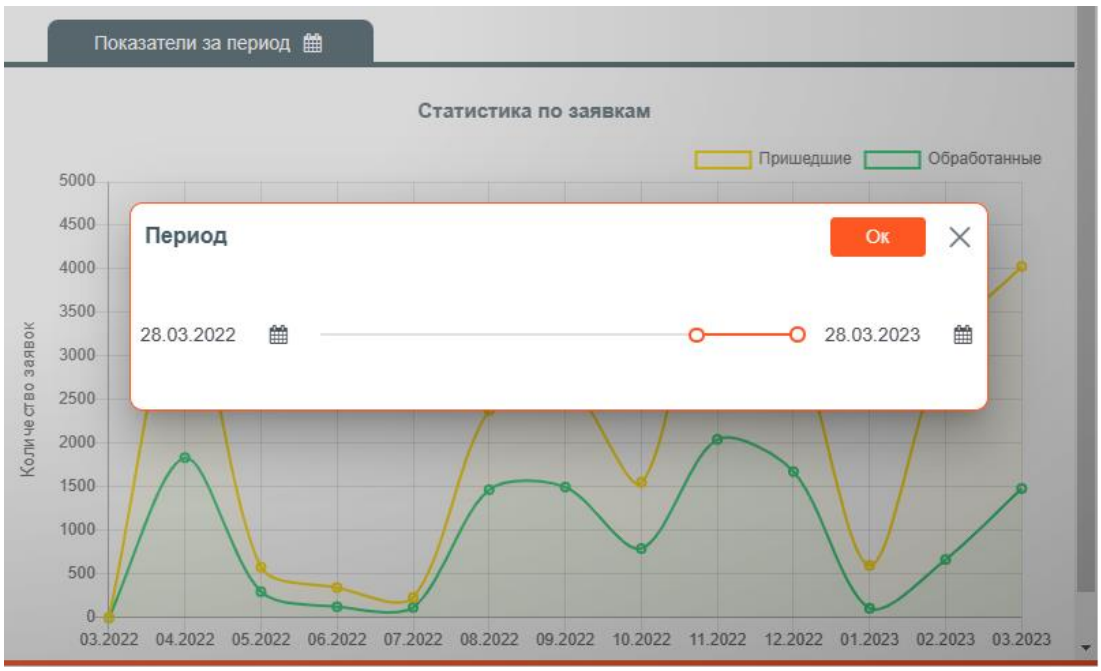


Рисунок 136 – Настройка показателей поступления и обработки заявок за выбранный за период

На вкладке **Моментальные показатели** можно ознакомиться с данными по всем заявкам на сегодняшний день: количество просроченных, необработанных, находящихся в работе и на уточнении.

Здесь же представлены две таблицы – данные обработки заявок по справочникам и данные по эффективности работы сотрудников отдела. Щелчок левой кнопкой мыши по наименованию справочника позволяет перейти к отчету SLA по выбранному справочнику, а щелчок по имени сотрудника к отчету по конкретному сотруднику.

Все вкладки, кроме вкладки **Моментальные показатели**, помимо статистики по справочникам и сотрудникам содержат графики поступления и обработки заявок по неделям, месяцам, годам и за выбранный период соответственно. Ось Y – это количество заявок, ось X – это время. При наведении курсора на точку графика во всплывающем окне отображается количество заявок в выбранной точке.

Все отчеты SLA могут быть выгружены в файл PDF. Для этого в нижней правой части окна располагается ссылка Экспорт в PDF.

3.12.13 Шаблоны заявок

Создание и сохранение шаблонов заявок производится в подсистеме **Конфигурация данных**, обратиться к которому можно через кнопку ***Навигация***. Работа с **Конфигурацией данных** разрешена только для пользователей с привилегией редактирования метаданных (см. п. 3.28.1). Для работы в этом разделе режим редактирования (управления пакетами изменений) включать не требуется.

3.12.13.1 Методика разработки шаблонов заявок

В зависимости от принятой в компании организации работы со справочными данными, можно условно выделить три подхода.

В первом случае, для каждой справочной группы назначается один или несколько *Экспертов НСИ*, которые отвечают за правильность вносимых в справочник данных и обрабатывают заявки. При таком подходе настройка *Экспертов НСИ*, а также сроков, маршрутов и условий выполнения заявки производится в карточке справочной группы в разделе **Заявки**. Для такого варианта работы при настройке шаблона заявок необходимо выбирать условия получения экспертов, сроков и маршрутов согласования «из группы» (см. п. 3.12.18)

Для второго похода характерно отсутствие определенных экспертов НСИ на уровне справочных групп. В этом случае эксперты назначаются в рамках шаблона заявки, где также прописываются условия выбора, маршрут согласования и согласующие лица.

Третий вариант заключается в том, что любой пользователь, входящий в группу экспертов НСИ может обрабатывать заявки по любым справочным группам. В этом случае при настройке шаблона заявки нужно указать сразу всех возможных экспертов без настройки условий фильтрации (см. п. 3.12.18) либо не указывать никого. При таком подходе передача заявки в работу будет определяться на этапе диспетчеризации либо диспетчером, указанным в шаблоне заявки, либо *Менеджером заявок*, который выступает в данном

случае диспетчером и передаст заявку конкретному специалисту на свое усмотрение.

Отдельный вариант представляют собой заявки на нормализацию и изменение позиции, создающиеся автоматически, при импорте данных из внешних информационных систем. Такие заявки настраиваются на обменных классах (см. п. 3.30.4), однако принцип настройки в целом остается неизменным. С точки зрения эксперта НСИ и согласующих работа с такими заявками не отличается от работы с обычными заявками. При этом, в случае, если речь идет о заявке на нормализацию, объект НСИ уже создан в *Системе* и требует проверки на соответствие требованиям нормализации и установки статуса «Нормализована».

3.12.14 Создание новой группы шаблонов

Если в **Конфигурации данных** отсутствует нужная группа в разделе **Шаблоны заявок** (Рисунок 137), пользователь может создать её самостоятельно. Рекомендуется группировать шаблоны по области использования, например, шаблоны на добавление номенклатурного объекта, шаблоны для удаления и т.п.

Для того, чтобы создать новую группу необходимо:

- перейти в подсистему **Конфигурация данных**;

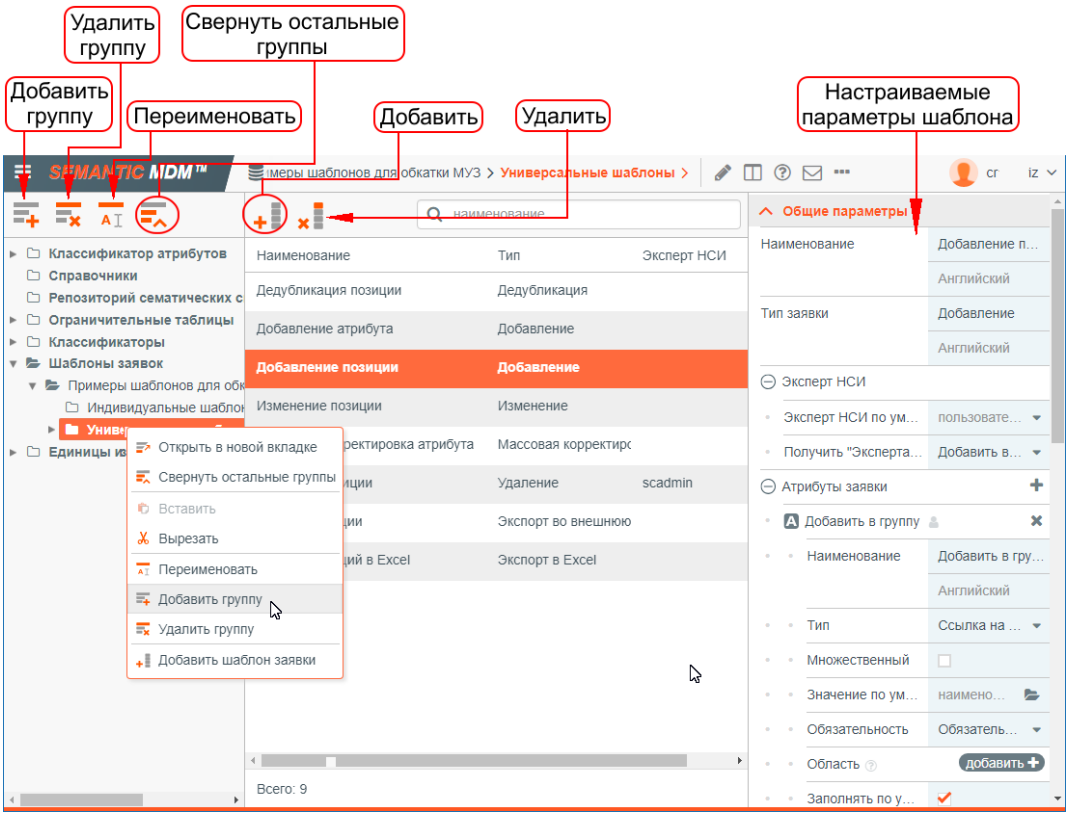


Рисунок 137 - Контекстное меню для работы с заявками

- установить курсор в разделе **Шаблоны заявок** и нажать кнопку ***Добавить группу*** или выбрать ту же команду из контекстного меню;
- в правой части рабочего окна в разделе **Общие параметры** ввести наименование группы на русском и английском (по необходимости) языках и её описание (необязательно).

Созданная группа будет размещена в дереве.

3.12.15 Новый шаблон заявки

Для того чтобы создать новый шаблон заявки требуется:

- перейти в подсистему **Конфигурация данных**;
- установить курсор в разделе **Шаблоны заявок** на нужной группе и вызвать из контекстного меню команду **Добавить шаблон заявки** либо, находясь на сетке данных, нажать кнопку ***Добавить***;
- к карточке шаблона заявки ввести наименование шаблона на русском и английском (по необходимости) языках;
- в ячейке **Тип заявки** из выпадающего списка выбрать один из

существующих типов или ввести собственный;

- ячейка **Справочник** предназначена для выбора из выпадающего списка справочника, для которого будет доступен данный шаблон (Рисунок 138). Для выбора доступны только те справочники, которые зарегистрированы в *Системе*. Ячейка может быть оставлена пустой, в этом случае шаблон будет доступен из любого справочника или справочной группы;

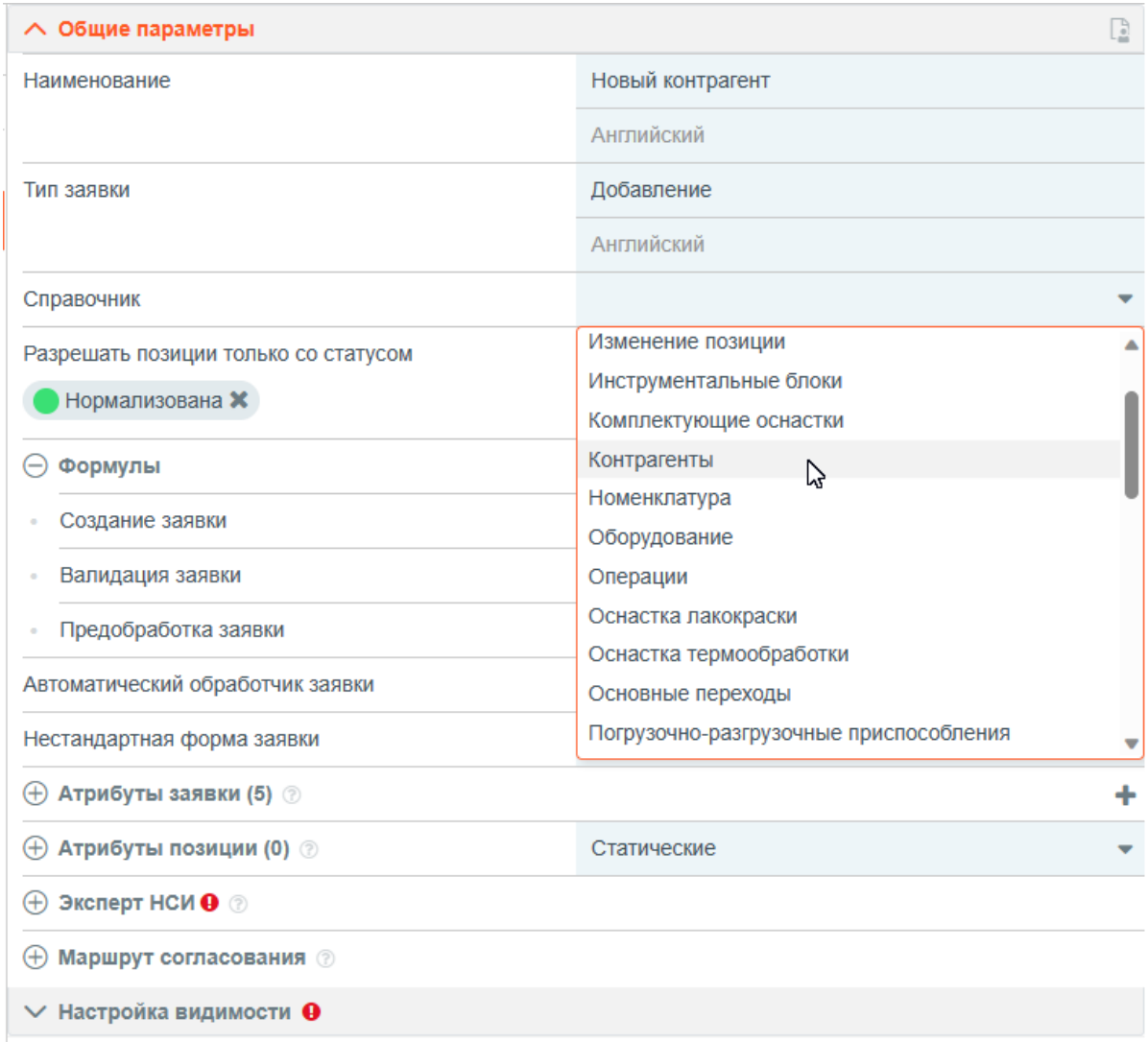


Рисунок 138 - Выбор справочника в шаблоне заявки

- **Разрешать позиции только со статусом** - выпадающий список, в котором пользователь может выбрать статусы, с которым допускается публиковать новые и изменённые позиции. Это позволяет запретить публикацию пакетов изменений по заявкам, в которых новая или

изменённая позиция, например, имеет статус *Не нормализована* или *Заблокирована*;

- категория **Формулы** (см. п. 3.12.20) содержит поля для ввода следующих видов формул для работы с заявкой:
  - ячейка **Создание заявки** может содержать скрипт GROOVY, разработанный в редакторе формул (вызывается кнопкой в правой части ячейки) для автоматического заполнения полей заявки после ее создания. Этот скрипт позволяет заполнить поля значениями «по умолчанию» и скорректировать набор доступных для заполнения атрибутов, в зависимости от заявителя;
  - ячейка **Валидация заявки** может содержать скрипт GROOVY, разработанный в редакторе формул (вызывается кнопкой в правой части ячейки) для выполнения проверки заполнения и корректности введенных в заявку данных, как для основной, так и для вложенных позиций и их атрибутов. Проверка выполняется по отдельной кнопке ***Проверить атрибуты*** (Рисунок 139), которая автоматически появляется в карточке заявки при её создании и заполнении, если в шаблоне задана формула валидации.

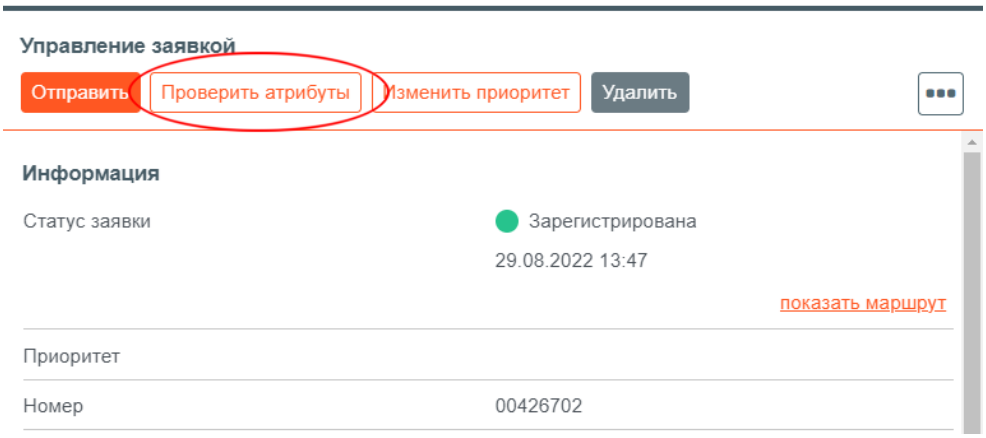


Рисунок 139 - Кнопка ***Проверить атрибуты*** в заявке по шаблону с формулой валидации

Если заявка не прошла не обязательные проверки, то *Система* выдаст предупреждение перед отправкой и запретит отправлять такую заявку на обработку;

- в ячейке **Предобработка заявки** можно вызвать **Редактор формул**, где при помощи скрипта Groovy можно создать формулу, согласно которой будет проводится предварительная обработка заявки, в зависимости от содержания её атрибутов и / или назначенной ветки справочника. Доступ к **Редактору формул** осуществляется кнопкой  в правой части ячейки;

Для всех формул возможно, автоматическое добавление справочных групп, из области атрибута заявки с установленным маркером **Заполнять при создании из заявки**.

- **Автоматический обработчик заявки** - выбор стандартного или пользовательского обработчика из выпадающего списка (Рисунок 140). Стандартный обработчик срабатывает на событие, связанное с изменением статуса заявки с «Зарегистрирована» на «Ожидает обработки» / «Ожидает диспетчеризации» (подробнее про работу обработчика см. п. 3.31.4) либо позволяет после выполнения заявки и публикации пакета изменений, выполнить автоматический экспорт позиции, указанной в заявке, во внешние ИС того же предприятия, в котором находится *Заявитель*. Чтобы удалить выбранный обработчик нужно нажать кнопку  ***Удалить*** в соответствующей ячейке. Кнопка  ***Настройка модуля*** отвечает за особенности настройки выбранного автоматического обработчика. Если никаких настроек разработчиком расширения не предусмотрено, то открывающаяся форма настройки будет содержать только одну строку - **Настройки подключения к Semantic MDM**.

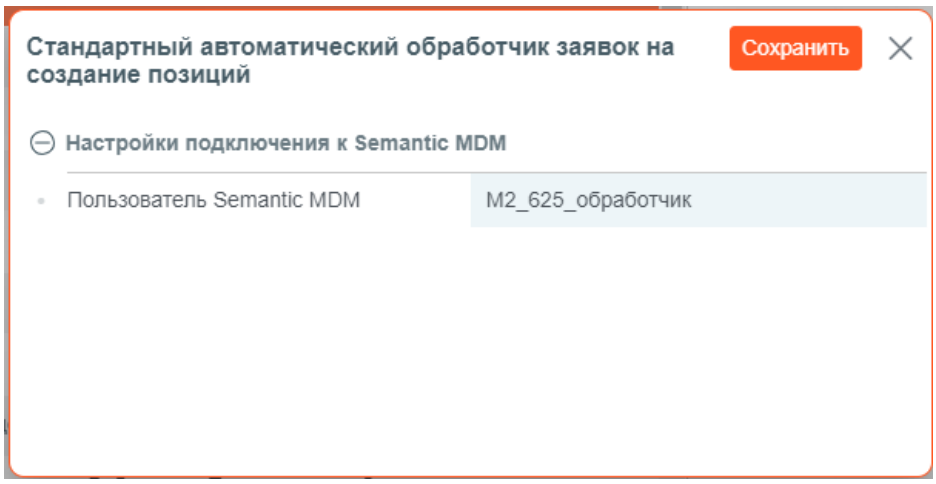


Рисунок 140 - Настройки стандартного обработчика заявок

- **Нестандартная форма заявки** - выбор из выпадающего списка специально разработанного расширения, зарегистрированного в *Системе*, представляющего собой кастомизированную форму заявки, разработанную на Angular. ниже на рисунке показан демонстрационный пример заявки, интерфейс которой отображает текст JSON-сообщения, используемого при импорте / экспорте данных (см. раздел 3.30).

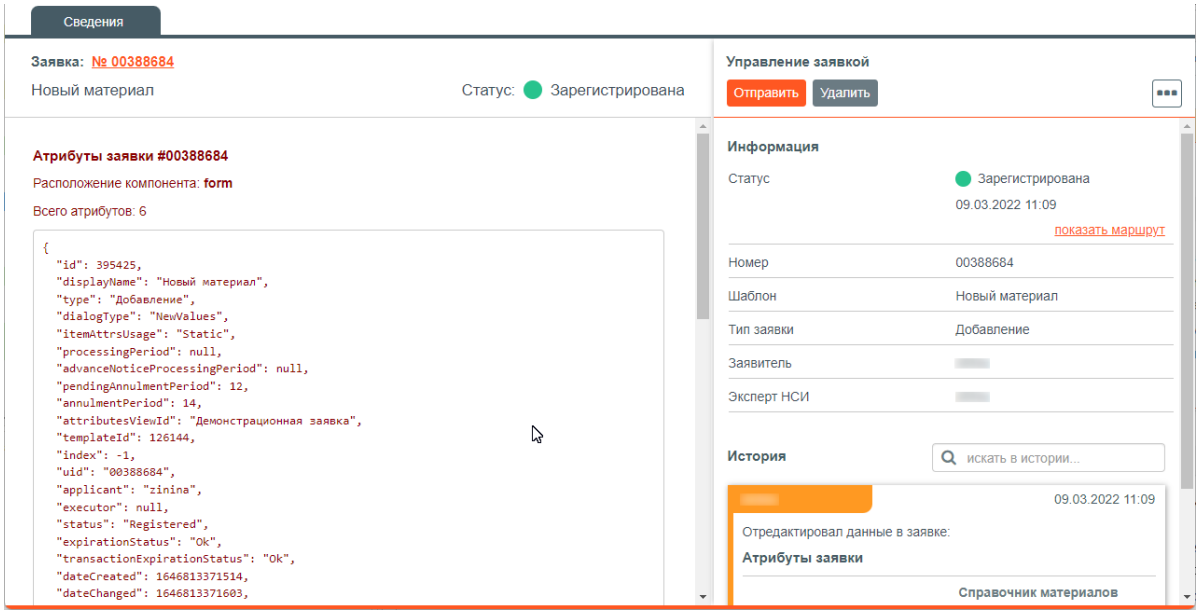


Рисунок 141 - Пример кастомизированной формы заявки

- **Разрешить использование в групповых заявках** (Рисунок 142) - опция, установка которой разрешает использовать данный шаблон в групповых заявках. Опция доступна только для шаблонов заявок с вариантом отображения атрибутов позиции – *Новые значения*.

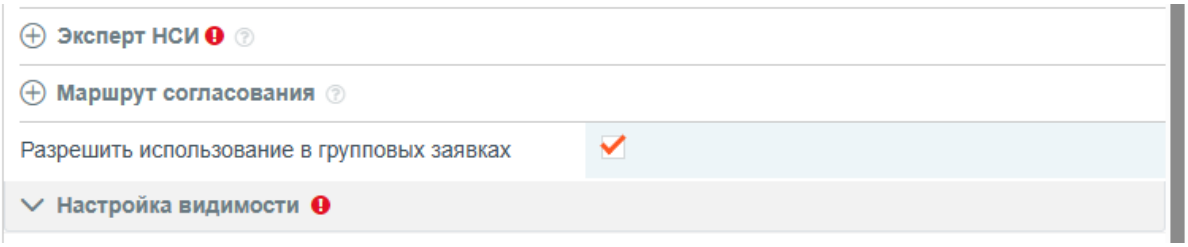


Рисунок 142 - Установка опции **Разрешить использование в групповых заявках**

3.12.16 Настройка атрибутов заявки

Атрибуты заявки предназначены для описания существа запрашиваемых изменений в системе. Значения этих атрибутов не передаются в позицию.

В разделе **Атрибуты заявки** карточки шаблона заявки нажать кнопку ***Добавить атрибут заявки***. В открывшемся окне ***Атрибуты заявок*** (Рисунок 143) выбрать из списка нужный атрибут и нажать кнопу ***Ок***.

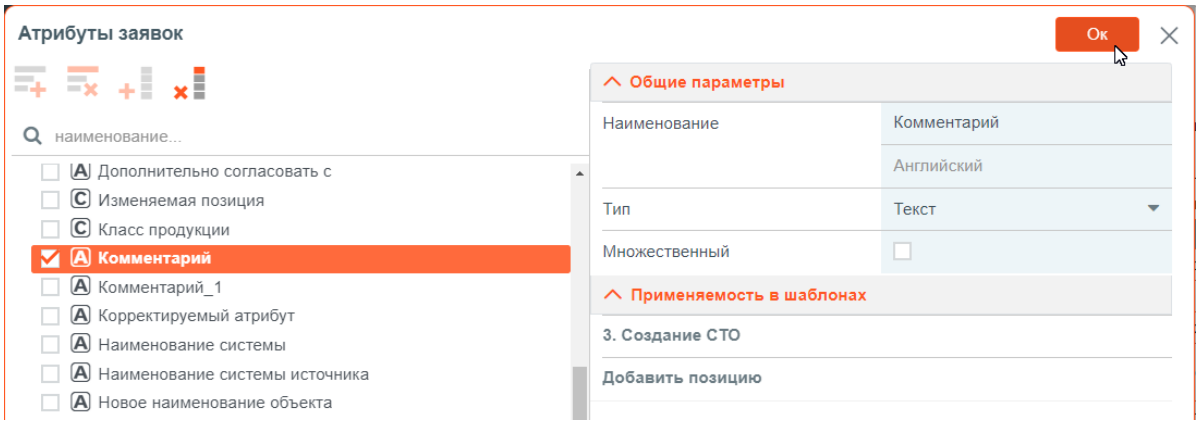


Рисунок 143 - Добавление атрибута заявки в шаблон

Если среди предлагаемых атрибутов заявки нет нужного, то можно добавить его в этом же окне используя кнопки **Добавить группу** / **Добавить**.

Атрибут добавляется и хранится в специальном разделе **Классификатор атрибутов заявки**, работа с которым полностью идентична работе в **Классификатором атрибутов** (см. раздел 3.16). При удалении атрибута заявки из классификатора система проверяет его использование в шаблонах заявок. Если атрибут используется в каком-либо шаблоне заявки, то удалить его можно, только после удаления из шаблона.

В карточке атрибута заявки необходимо указать:

- **Наименование** – название атрибута на русском и английском (опционально) языке;
- **Тип** – тип создаваемого атрибута. В шаблонах поддерживаются типы, представленные в таблице ниже.

Таблица 15. Типы атрибутов заявки

Тип атрибута	Описание назначения
Строка	Строка длиной от 0 до 250 символов. Для наименований и обозначений
Текст	Используется для хранения текста произвольной длины, обычно применяется для текстовых описаний и пояснений
Файл	Позволяет присоединять к заявке файлы любых документов, как текстовых, так и графических, любых форматов
Логический тип	Тип данных (boolean), которые могут принимать два возможных значения «Да» или «Нет». В заявке предоставляет ячейку для установки маркера. Не может являться множественным!
Ссылка на группу/позицию	Позволяет вставить в заявку справочную группу или позицию, принадлежащую ей
Ссылка на группу	Позволяет добавить в заявку ссылку на существующую справочную группу. Применяется для шаблонов на удаление и редактирование справочных групп
Ссылка на позицию	Позволяет добавить в заявку ссылку на существующую справочную позицию. Применяется для шаблонов на удаление и редактирование позиций
Ссылка на атрибут	Позволяет добавить в заявку ссылку на существующий в классификаторе атрибут. Применяется для шаблонов на удаление и редактирование атрибутов
Ссылка на декларацию	Позволяет добавить в заявку ссылку на существующую в справочной группе декларацию атрибута. Применяется для шаблонов на удаление и редактирование деклараций атрибутов справочных групп и позиций

- **Множественный** – установленный маркер означает, что для данного атрибута в заявке можно указать несколько значений, например,

банковские счета контрагента или частоты вращения шпинделя для станка со ступенчатым регулированием.

- **Применяемость в шаблонах** – раздел со списком шаблонов заявок, в которых используется данный атрибут. Формируется автоматически, не редактируется.

После добавления атрибута заявки в шаблон нужно выполнить дополнительные настройки декларации, состав которых зависит от типа данных атрибута:

- **Значение по умолчанию** – содержит запись, выводимую в атрибут при заполнении заявки по умолчанию:
  - Для атрибутов-ссылок становится доступной кнопка ***Выбрать объект*** и указываются группа, позиция, атрибут или декларация, выводимые по умолчанию, если заявитель не указал значение;
  - Для атрибутов типа «файл» – можно добавить файл, например, пустой шаблон документа;
- **Обязательность** – признак выбирается из выпадающего списка и имеет три возможных значения:
  - Необязательный – атрибут может оставаться незаполненным;
  - Обязательный для Заявителя – атрибут должен быть обязательно заполнен Заявителем, в противном случае заявка не будет отправлена;
  - Обязательный для Эксперта НСИ – атрибут должен быть обязательно заполнен Экспертом НСИ, в противном случае заявка не будет отправлена на согласование.

Обязательный атрибут отмечается в заявке символом * (звёздочка) красного цвета слева от наименования (Рисунок 144).

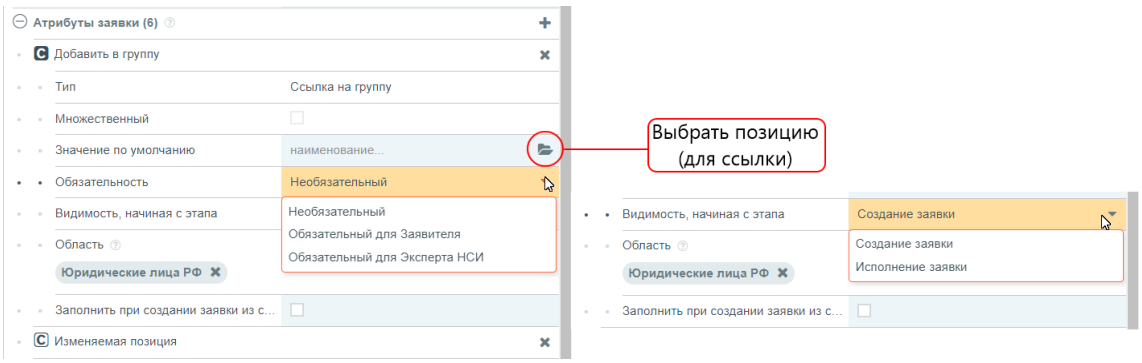


Рисунок 144 - Указание значения по умолчанию, обязательности и видимости атрибута в заявке

- **Видимость начиная с этапа** – признак выбирается из выпадающего списка и имеет два возможных значения:
  - Создание заявки – атрибут виден Заявителю на этапе заполнения заявки;
  - Исполнение заявки – атрибут виден только после отправки заявки на исполнение, начиная с получения статуса «Ожидает диспетчеризации»;

Настройки **обязательности** и **видимости** выводятся в виде двух пиктограмм в поле с наименованием атрибута заявки (Рисунок 145), позволяющих, не раскрывая настроек , определить, является ли атрибут обязательным для заполнения Заявителем (зеленая звездочка) или Исполнителем заявки (красная звездочка), а также является ли атрибут видимым начиная со стадии создания (зеленый глаз) или исполнения заявки (красный глаз).

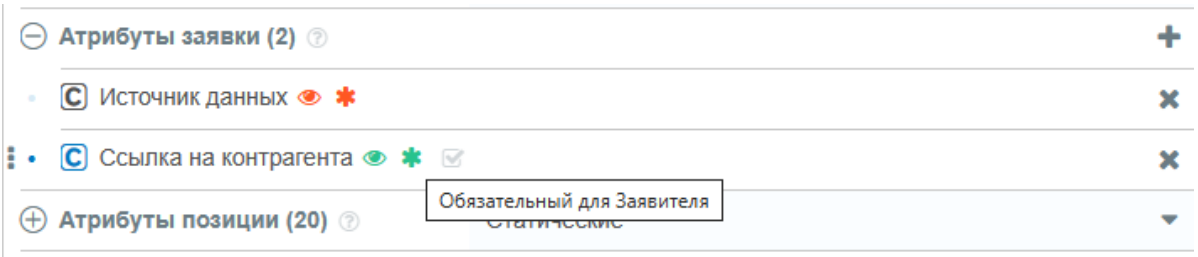


Рисунок 145 - Пиктограммы в поле наименования атрибута заявки

- **Маска ввода** – позволяет ввести маску (см. п. 3.16.9) для заполнения атрибута типа «строка». Для остальных типов атрибутов настройка не доступна;
- **Список значений** - в ячейке можно указать список значений, которые будет выбираться *Заявителем* при создании заявки из выпадающего списка. Свойство доступно только для атрибута типа «строка»;

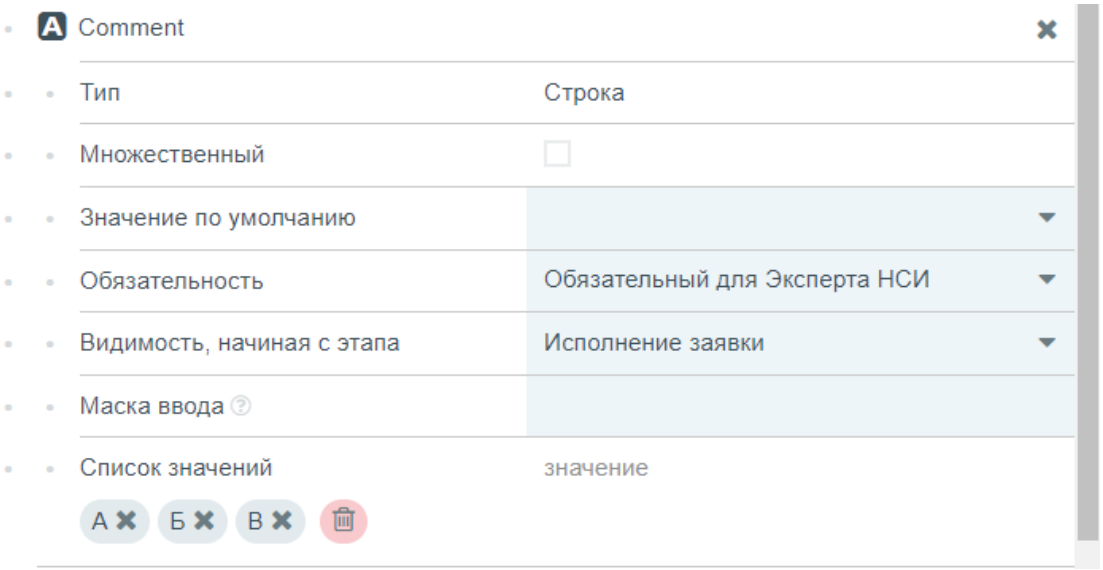


Рисунок 146 - Вид списка значений для атрибута заявки типа «строка»

- **Область** – позволяет указать справочную группу или классификатор (один или несколько), объекты которой (которых) могут использоваться для заполнения значения атрибута. Настройка доступна только для атрибутов-ссылок. Добавление области осуществляется кнопками ***Классификатор*** (для иерархических классификаторов, например, ОКПД2) и ***Группы*** (для справочных групп) (Рисунок 147);

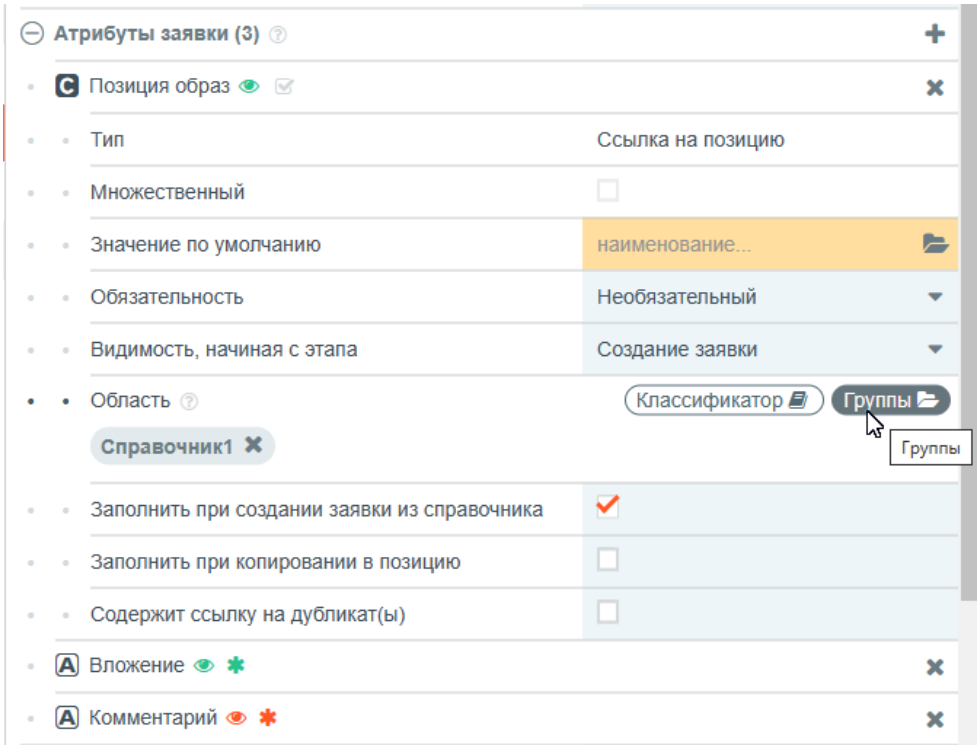


Рисунок 147 - Добавление области для выбора значения атрибута

- **Инициализировать заявку указанной группой / позицией** – маркер доступен только для атрибутов-ссылок. При установке данного маркера для заявки с динамическими атрибутами (см. п. 3.12.17), определяет из какой справочной группы / классификатора будет получен состав атрибутов позиции. Для заявки в целом, установленный маркер определяет для какой справочной группы какие атрибуты позиции в заявке требуется исключить, какие сделать обязательными, а какие сделать информационными;
- **Заполнить при копировании в позицию** – маркер доступен только для атрибутов-ссылок на позицию. Обеспечивает автоматическую запись ссылки на позицию, которая была создана или изменена из заявки с помощью кнопок: ***Создать позицию***, ***Копировать атрибуты***, ***Копировать все***, ***Копировать выбранное значение***.
- **Содержит ссылку на дубликат(ы)** – маркер доступен только для атрибутов-ссылок на позицию. Применяется для настройки заявки на дедубликацию. Для настройки шаблона заявки на дедубликацию следует настроить два атрибута заявки типа «ссылка на позицию»,

один из которых будет содержать ссылку на оригинал, а второй на один или несколько дубликатов. В заявке для атрибута с маркером **Содержит ссылку на дубликат(ы)** доступна кнопка выполнения поиска дубликатов. Настроенная для атрибута множественность позволяет определить единственность или множественность дубликатов, задаваемых пользователем, а также определяет наличие кнопки ***Удалить все***, по которой можно удалить все добавленные ранее дубликаты. Если в шаблоне заявки определен атрибут заявки, в котором будут указываться ссылка на дубликат(ы), то ссылка на оригинал всегда указывается в атрибуте заявки с маркером **Заполнить при создании заявки из справочника**;

- **Информационный атрибут** – маркер доступен только для простых атрибутов (см. п. 3.12.17). Установка данного маркера означает, что значение этого атрибута заполняется скриптом GROOVY на создание заявки или нестандартным обработчиком создания заявки, и не может быть отредактирован пользователем через интерфейс *Системы*;
- **Расширения файлов** – настройка доступна только для атрибутов заявки с типом данных «файл». Позволяет указать расширения файлов, которые разрешено присоединять к заявке, например, PDF, DOC и т.п. Файлы, расширения которые не были указаны в списке, не могут быть добавлены в заявку.

3.12.17 Настройка атрибутов позиции

Атрибуты позиции необходимы для обеспечения возможности автоматического создания позиции в справочной группе и копирования значений атрибутов из заявки в позицию при её изменении .

Существует три вида представления атрибутов позиции в заявке, которые выбираются из выпадающего списка (Рисунок 148).

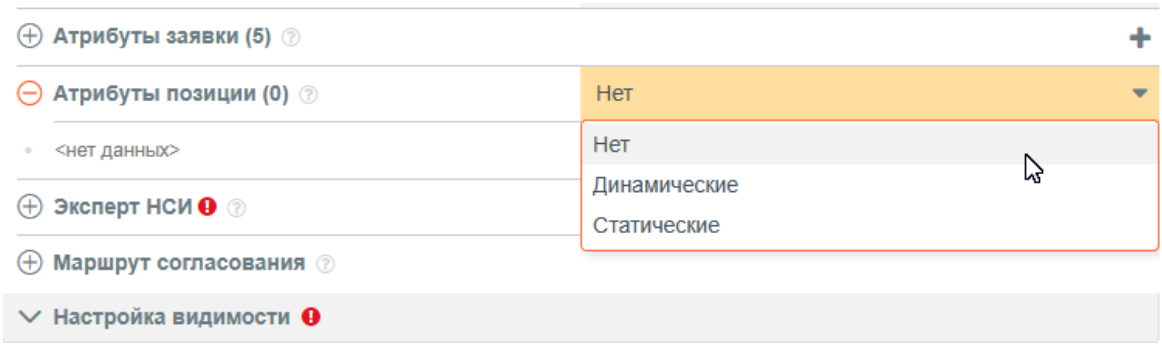


Рисунок 148 - Выбор типа добавления атрибутов позиции в заявку

- **Нет** – применяется для таких шаблонов, как удаление позиций и групп, дедубликация, выполнение экспорта. В этих случаях достаточно использовать атрибуты заявки для ссылки на объект, так как никаких изменений в самой позиции не предусмотрено;
- **Динамические** – позволяет сослаться в ячейке **Получить атрибуты позиции из группы** на атрибуты справочной, которая была настроена в атрибутах заявки, принадлежащих к типам «ссылка на группу/позицию», «ссылка на группу», «ссылка на позицию». У атрибута заявки должен быть обязательно установлен маркер **Инициализировать заявку указанной группой / позицией**. При этом если выбраны атрибуты заявки типа «ссылка на группу/позицию», «ссылка на группу» то для них в качестве отображаемых значений атрибутов позиции доступен только один вариант – *Новые значения*. Для атрибутов заявки типа «ссылка на позицию» возможны три варианта – *Исходные значения, Новые значения, Исходные и новые значения*, при этом в настройках атрибута типа «ссылка на позицию» должен быть обязательно установлен маркер **Заполнить при создании заявки из справочника**, в противном случае, будет доступен только вариант – *Новые значения*;
- **Статические** – позволяет выбрать любые атрибуты из любой группы справочника, при этом список в заявке является постоянным (Рисунок 150). Если в атрибутах заявки присутствует атрибут типа «ссылка на позицию» и для него установлен маркер **Заполнить при создании**

заявки из справочника, то возможны три варианта отображения значений статического списка – *Исходные значения, Новые значения, Исходные и новые значения*, в противном случае, будет доступен только вариант – *Новые значения.*

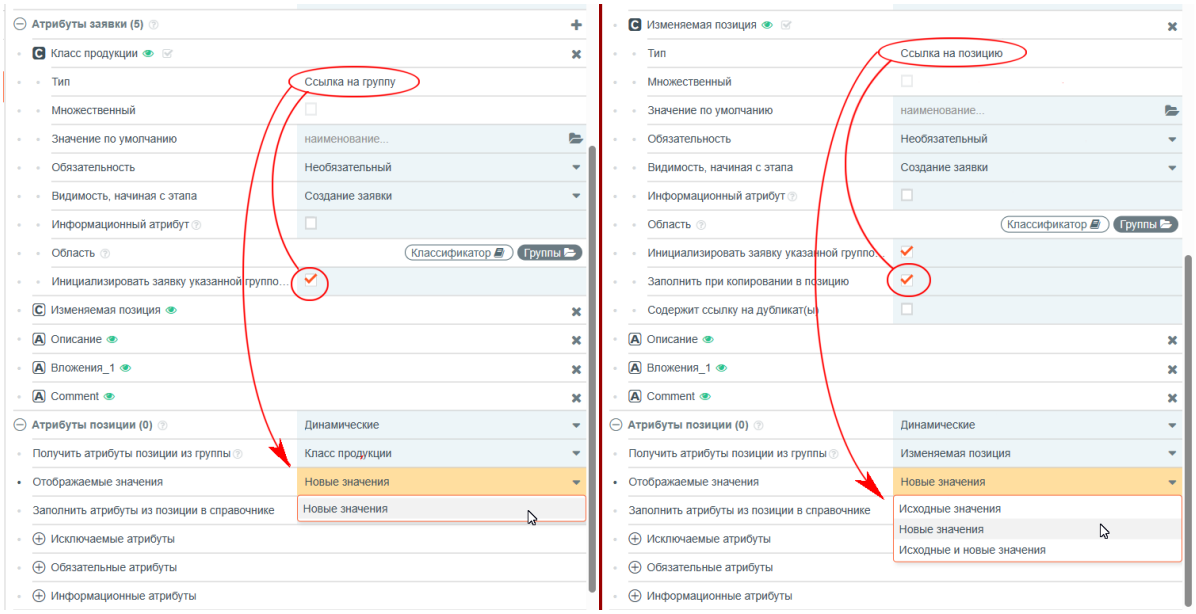


Рисунок 149 - Настройка отображаемых значений атрибутов в заявке: а – для ссылок на группу, б – для ссылок на позицию с маркером Заполнить при создании заявки из справочника

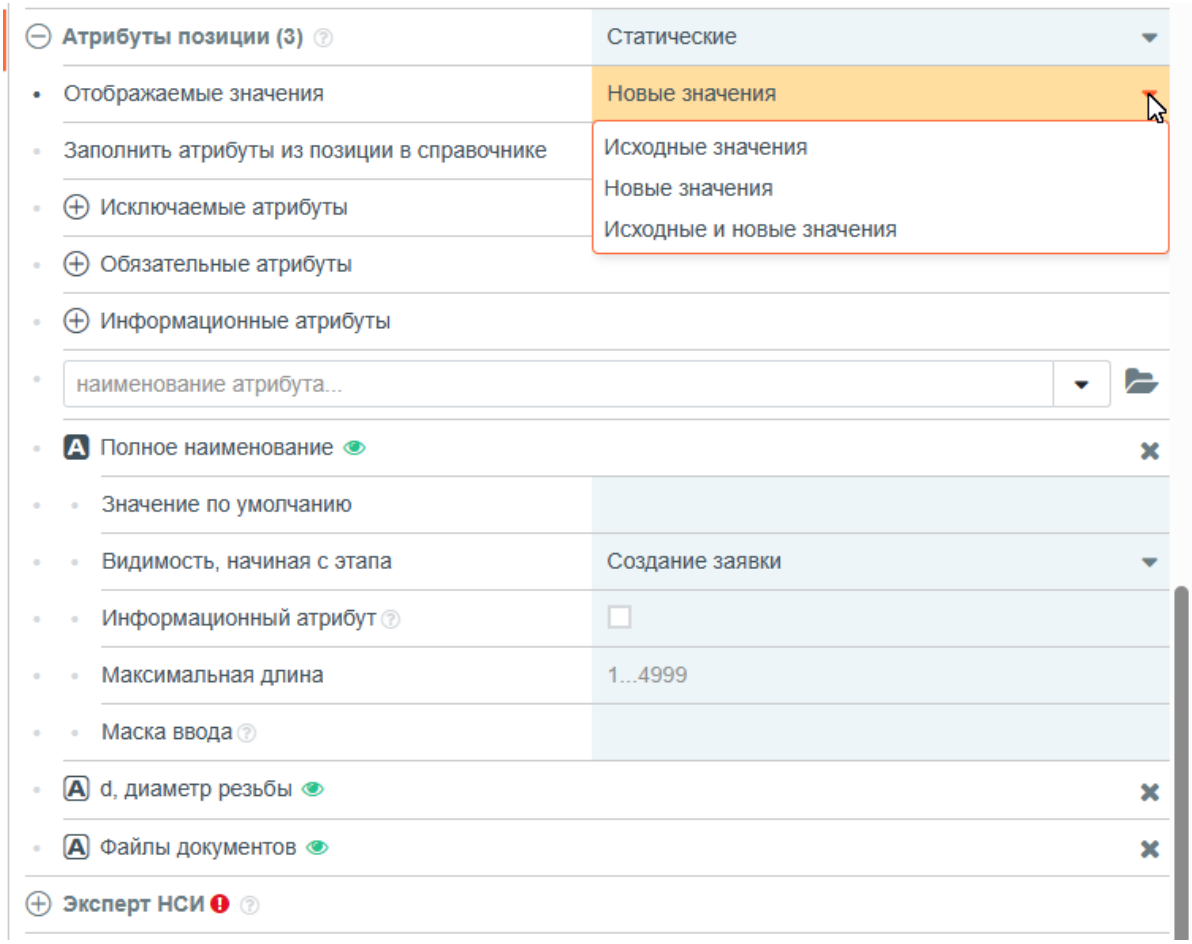


Рисунок 150 - Настройка статических атрибутов

Настройка выбранных атрибутов позиции статического списка включает определение следующих параметров:

- **Значение по умолчанию** – содержит значение, выводимое в атрибут при заполнении заявки по умолчанию. Настройка не применяется для атрибутов связи (агрегации и композиции);
- **Видимость начиная с этапа** – признак выбирается из выпадающего списка и имеет два возможных значения:
  - Создание заявки – атрибут виден Заявителю на этапе заполнения заявки;
  - Исполнение заявки – атрибут виден только после отправки заявки на исполнение, начиная с получения статуса «Ожидает диспетчеризации»;
- **Информационный атрибут** – маркер доступен только для простых атрибутов. Установка данного маркера означает, что значение этого

атрибута заполняется скриптом GROOVY на создание заявки или нестандартным обработчиком создания заявки, и не может быть отредактирован пользователем через интерфейс *Системы*;

- **Область** – параметр, определяемый только для атрибутов связи (агрегация или композиция). Здесь следует выбрать справочник, в который будут сохраняться сведения из заявки;
- **Максимальная длина** – свойство присутствует только у атрибутов типа «строка». Позволяет определить количество символов, которые разрешено ввести в значение атрибута;
- **Маска ввода** – применяется только для строковых атрибутов;
- **Единицы измерения** – применяется только для атрибутов типа «число с плавающей точкой»;
- **Точность** – количество знаков после запятой. Применяется только для атрибутов типа «число с плавающей точкой».

Для статических атрибутов доступна настройка категорий (Рисунок 151), аналогично тому, как это выполняется в карточке позиции (см. п. 3.14.4) в подсистеме **Справочники**. Категория в заявке создаётся путём перетаскивания и совмещения (Drag&drop) атрибутов.

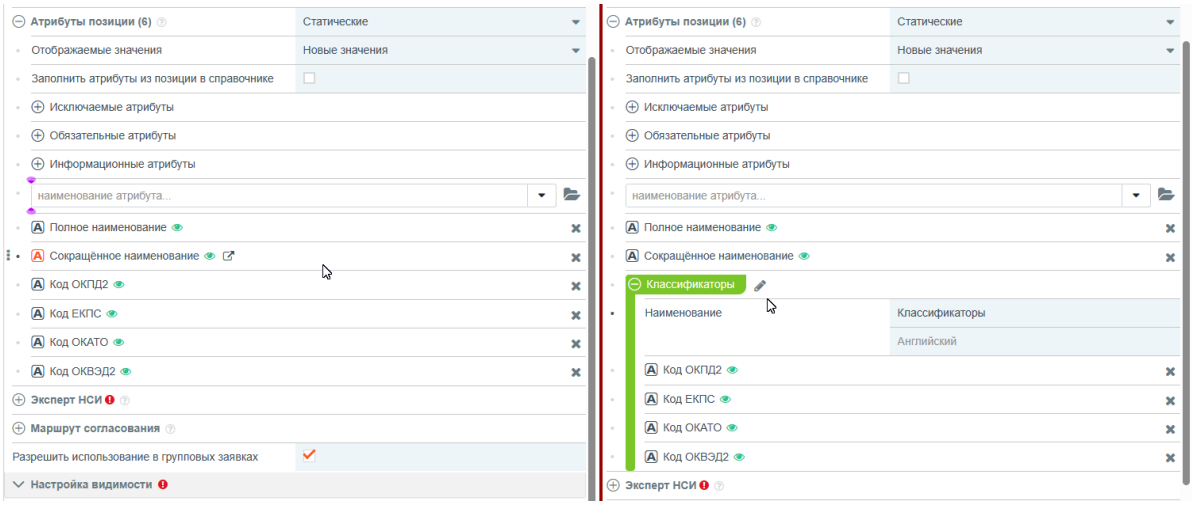


Рисунок 151 - Настройка категории для статических атрибутов заявки: слева до объединения в категорию, справа - после объединения

Для динамических и статических атрибутов позиции доступны следующие общие настройки:

- **Заполнить атрибуты из позиции в справочнике** – данная опция доступна только для заявки с режимом отображения значений *Новые значения* и позволяет заполнить атрибуты позиции в заявке значениями атрибутов текущей позиции в справочнике, при создании заявки из справочника. Если опция не установлена, то значения атрибутов в заявке будут заполняться пользователем вручную, вне зависимости от положения курсора при создании заявки;
- **Исключаемые атрибуты** – атрибуты, которые не будут передаваться в заявку из указанных справочных групп. Настройка применяется для исключения системных атрибутов, а также любых других атрибутов, не подлежащих заполнению при подаче заявки:
  - **Системные атрибуты** - выпадающий список системных атрибутов, которые могут быть исключены из заявки (Рисунок 152). К ним относятся: *Статус позиции*, *Родительская позиция*, *Дата начала действия*, *Дата окончания действия*;

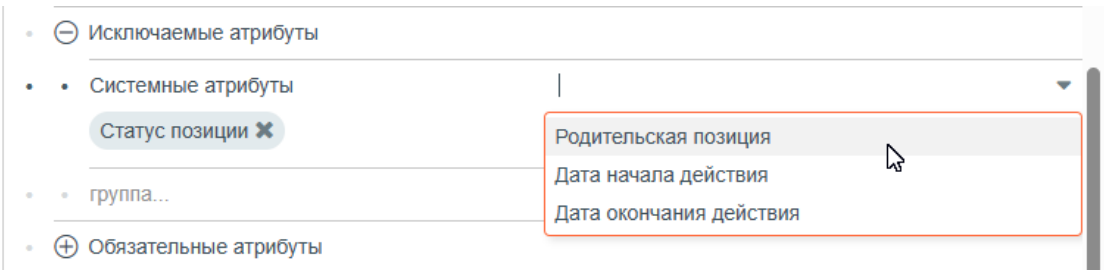


Рисунок 152 - Настройка исключаемых системных атрибутов

- Чтобы настроить список исключений необходимо в разделе **Исключаемые атрибуты** (Рисунок 153) выбрать из выпадающего списка или по кнопке ***Выбрать группу*** или ***Выбрать классификатор*** нужную справочную группу (группы) / классификатор, а затем для каждой справочной группы, используя выпадающий список с возможностью поиска

или по кнопке ***Выбрать атрибуты***, указать исключаемые атрибуты. Кнопка ***Удалить*** обеспечивает удаление справочной группы, а кнопка ***Удалить все***, расположенная в конце списка атрибутов - удаление всех выбранных атрибутов. При указании атрибутов в окне **Выберите атрибуты** пользователю доступна опция «*показать только продекларированные*», которая позволяет выбрать только те атрибуты, которые продекларированы в указанной на предыдущем шаге настройки группе / классификатор;

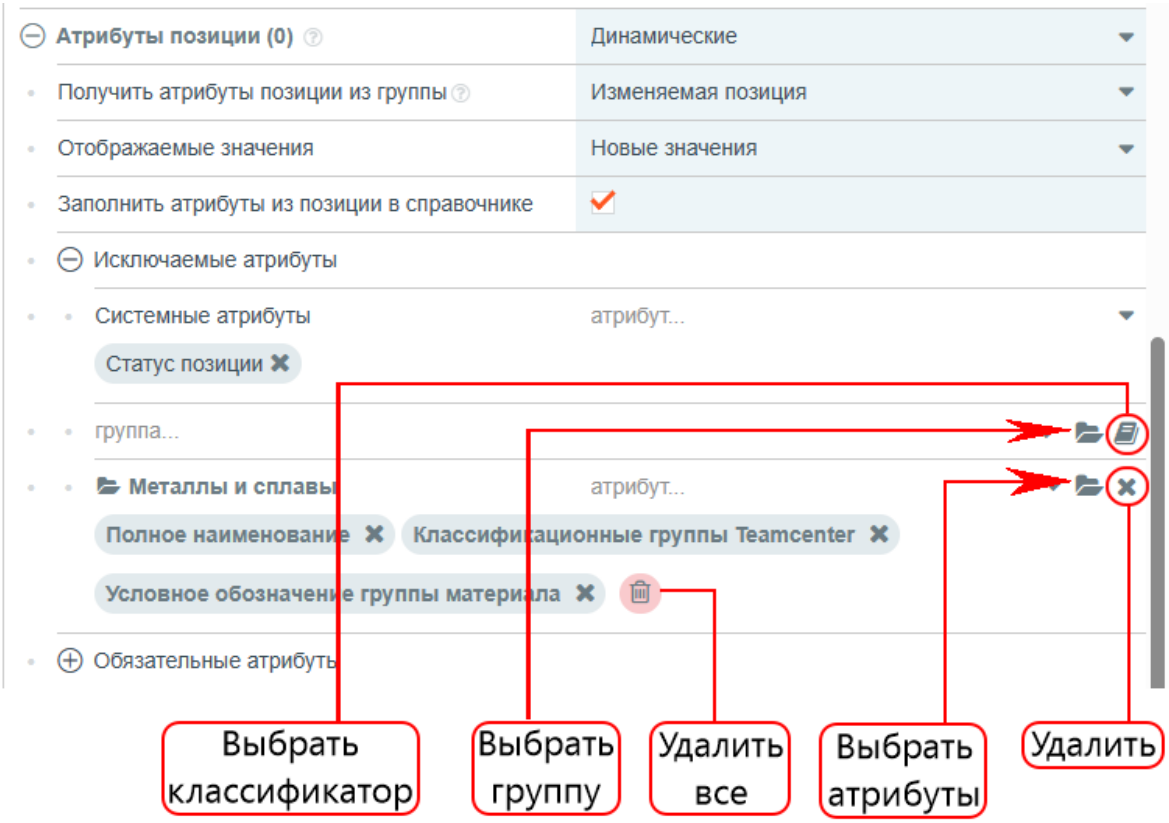


Рисунок 153 - Настройка исключаемых атрибутов

- **Обязательные атрибуты** – атрибуты, значения которых должны быть обязательно указаны в заявке, в противном случае заявка не может быть отправлена .Чтобы настроить список обязательных атрибутов необходимо в разделе **Обязательные атрибуты** выбрать из выпадающего списка или по кнопке ***Выбрать группу*** или ***Выбрать***

классификатор нужную справочную группу (группы) / классификатор, а затем для каждого выбранного объекта, используя выпадающий список с возможностью поиска или по кнопке ***Выбрать атрибуты***, указать исключаемые атрибуты. Кнопка ***Удалить*** обеспечивает удаление справочной группы, а кнопка ***Удалить все***, расположенная в конце списка атрибутов - удаление всех выбранных атрибутов. При указании атрибутов в окне **Выберите атрибуты** пользователю доступна опция *«показать только продекларированные»*, которая позволяет выбрать только те атрибуты, которые продекларированы в указанной на предыдущем шаге настройки группе / классификаторе;

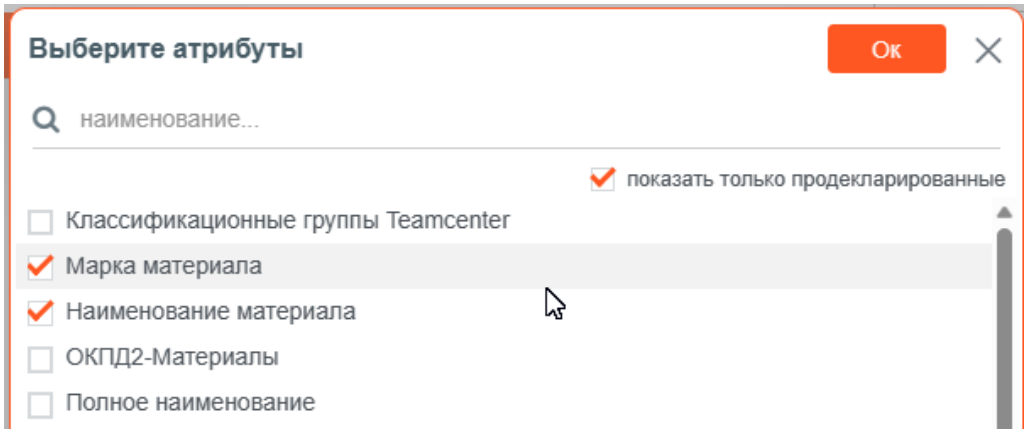


Рисунок 154 - Выбор исключаемых / обязательных / информационных атрибутов в заявке

- **Информационные атрибуты** – атрибуты, значения которых заполняются автоматически и, которые пользователь должен видеть, но не должен иметь возможность изменить. Настройка выполняется полностью аналогично с исключаемыми и обязательными атрибутами.

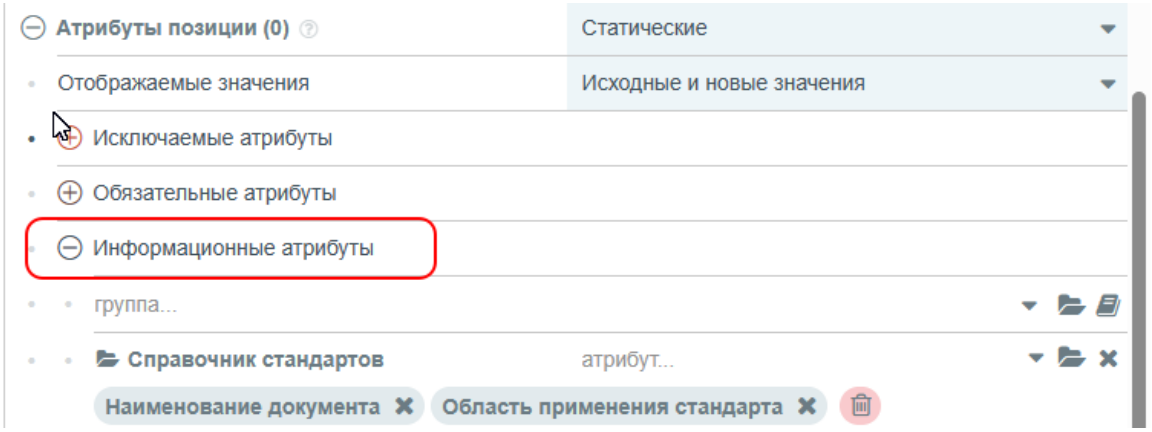


Рисунок 155 - Настройка информационных атрибутов

3.12.18 Эксперт НСИ и диспетчер заявок

В разделе **Эксперт НСИ** следует выбрать из выпадающего списка ячейки **Возможные эксперты НСИ по умолчанию** одного или нескольких пользователей либо группу пользователей, которые будут являться экспертами по заявкам, создаваемым по данному шаблону. В качестве варианта можно получить эксперта из справочной группы. Для получения эксперта из справочной группы, в выпадающем списке нужно выбрать наименование атрибута заявки, в котором присутствует ссылка на эту группу.

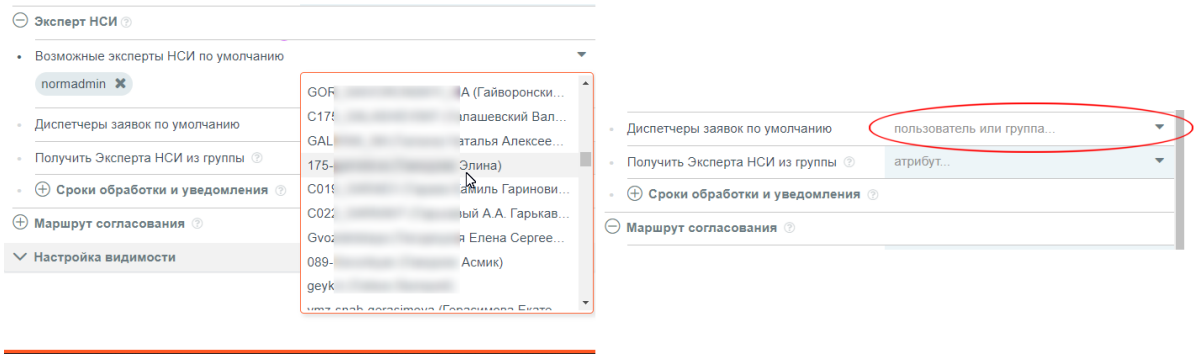


Рисунок 156 - Настройка *Эксперта НСИ* и диспетчера для заявок

Пиктограмма в панели Эксперт НСИ и на сетке данных означает, что для данного шаблона не указано ни одного исполнителя или не выбрана группа, откуда его получать.

Диспетчеры для заявок, созданных по данному шаблону, определяются в ячейке **Диспетчеры заявок по умолчанию**. Диспетчеры выбираются из выпадающего списка. можно выбрать одного или нескольких пользователей, либо группу пользователей, все участники которой будут выполнять роль диспетчера по данной шаблону.

Если выбрано несколько экспертов НСИ и группа пользователей в заявках, то можно настроить формулу фильтрации, которая обеспечит выбор конкретного эксперта в зависимости атрибутов заявки либо заявителя. Настройка фильтрации выполняется в скрипте Groovy, который размещается в ячейке **Формула предобработки заявки**.

3.12.19 Группы пользователей в заявках

Группы пользователей в заявках (Рисунок 157) – это уникальный глобальный список групп пользователей, предназначенный для формирования списков пользователей, участвующих в исполнении, согласовании и контроле заявок. Использование группы пользователей в заявка сокращает время на выбор необходимых специалистов и позволяет объединять их в группы согласно квалификации, выполняемым работам и т.д.

Настройка группы доступна пользователю с привилегией *Менеджер заявок*. Настройка может быть выполнена в подсистемах **Конфигурация данных** или **Журнал заявок**. Форма настройки вызывается кнопкой ***Группы пользователей в заявках***, расположенной на основной панели инструментов рабочего окна.

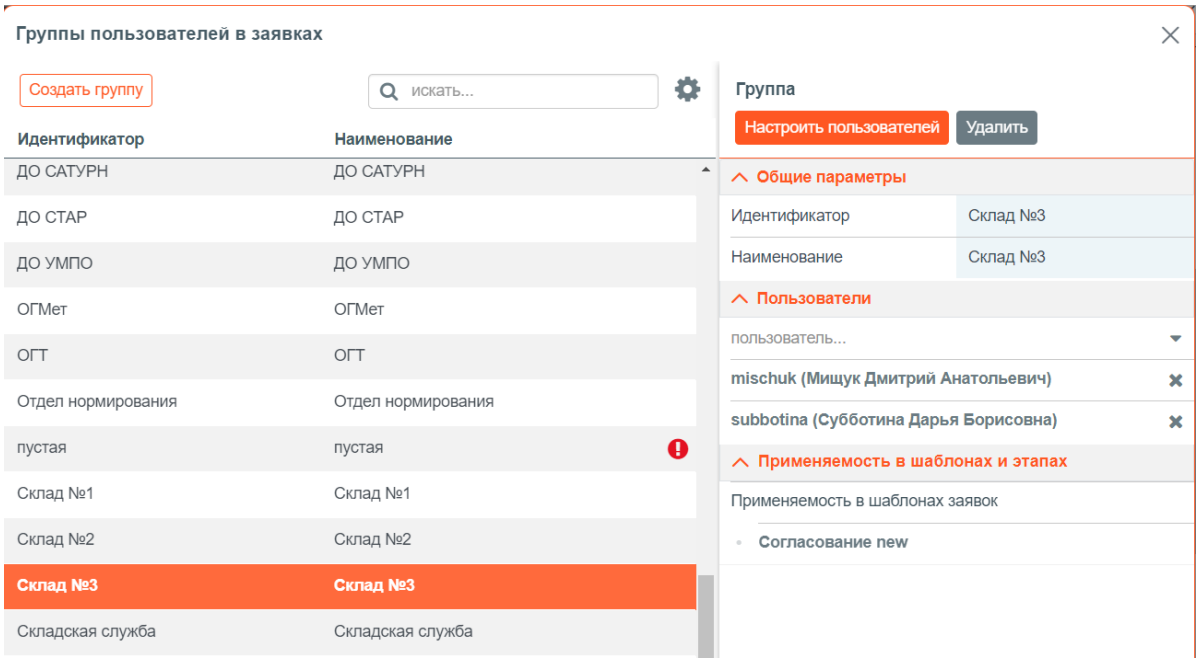


Рисунок 157 - Группы пользователей в заявках

Каждая группа пользователей в заявках имеет в карточке следующие настраиваемые параметры:

- **Идентификатор** – по умолчанию идентичен наименованию и должен быть уникальным в рамках *Системы*. Идентификатор используется в

формуле предобработки заявки;

- **Наименование** – должно быть уникальным в рамках *Системы*. Данное наименование отображается для пользователей при настройке шаблонов заявки;

  Пользователи – список пользователей, входящих в группу. Пользователи выбираются при настройке из выпадающего списка или в диалоге, поднимаемом по кнопке Настроить пользователей, где необходимо маркерами отметить специалистов, входящих в настраиваемую группу (Рисунок 158). Один пользователь может входить в несколько таких групп.

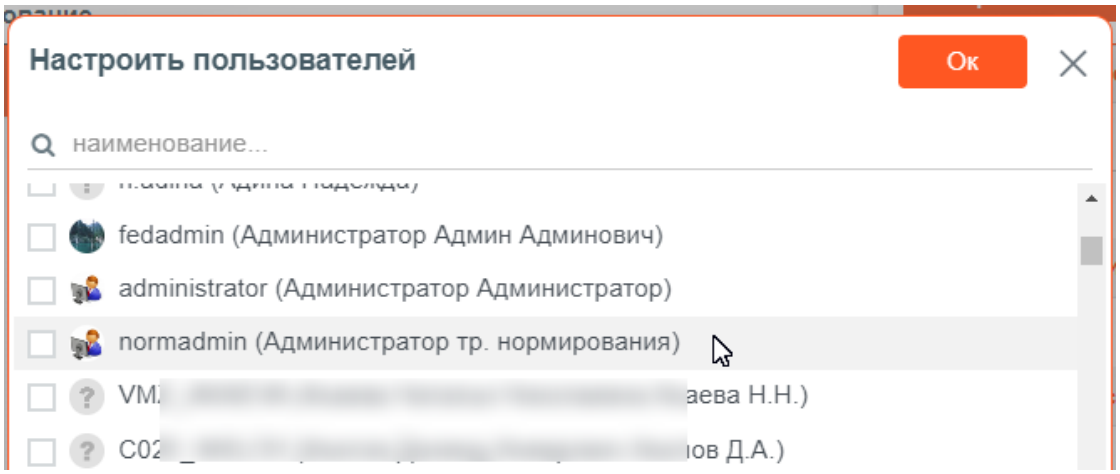


Рисунок 158 - Выбор пользователей в группу

Кроме настраиваемых параметров в карточке содержится раздел **Применяемость в шаблонах и этапах**, где автоматически указывается информация о том, в каких шаблонах заявок и на каких этапах обработки / согласования используется данная группа.

Настроенные группы пользователей в заявках могут быть использованы:

- в поле Возможные эксперты НСИ по умолчанию для шаблона заявки;
- в поле Возможные эксперты НСИ в настройках экспертов НСИ для справочной группы;
- в поле Диспетчеры заявок по умолчанию для шаблона заявки;
- в поле Диспетчеры заявок в настройках экспертов НСИ для справочной группы;

- в поле Участники согласования для шаблона заявки и справочной группы;
- в поле Пользователи, контролирующие срок обработки для шаблона заявки и в справочной группе;
- в поле Пользователи, контролирующие срок согласования для этапа согласования в шаблоне заявки и в справочной группе.

3.12.20 Формулы обработки заявки

Формулы **Создание заявки**, **Валидация заявки**, **Предобработка заявки**, а также **Формула предобработки этапа** это скрипты Groovy, который могут быть использованы для обогащения и проверки заявки, предварительной обработки с учетом заполненных значений, в том числе, для фильтрации пользователей - заявителей, списков экспертов НСИ и /или списков пользователей, контролирующих исполнение.

Для разработки формулы следует:

- в ячейке, предназначенной для ввода формул, в карточке шаблона заявки нажать кнопку ***Редактор формул***;

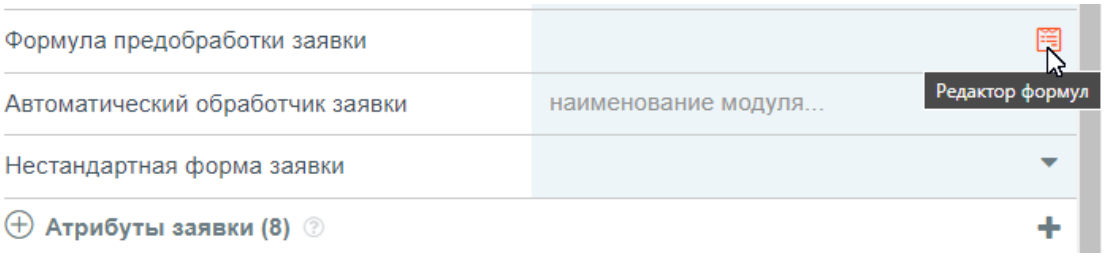


Рисунок 159 – Открытие редактора формул

- в открывшемся окне **Редактор формул** (Рисунок 160) написать скрипт Groovy, включая в него необходимые атрибуты.

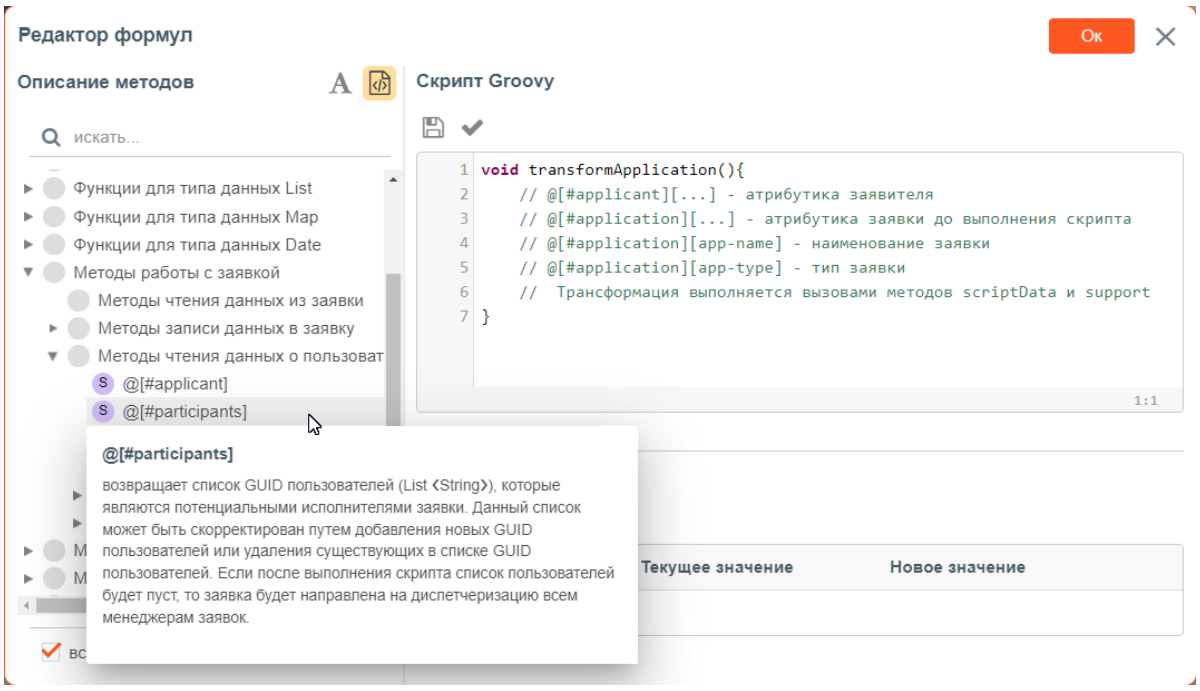


Рисунок 160 - **Редактор формул** для формулы предобработки заявки

В скрипте могут использоваться атрибуты, располагающиеся на панели слева на вкладке A, а также функции и методы Groovy, находящиеся на вкладке </>. В список атрибутов автоматически добавляются атрибуты справочных групп, из области атрибута заявки с установленным маркером **Заполнять при создании из заявки**. Во всех видах формул для заявок также можно использовать системные переменные:

- *Список возможных исполнителей* – для формулы предобработки заявки, позволяет в скрипт добавить системную переменную @[#participants] в которой определен список возможных исполнителей;
- *Список контролирующих пользователей* – для формулы предобработки заявки и предобработки этапа согласования позволяет добавить системную переменную @[#supervisors] в которой определен список контролирующих пользователей;
- *Список согласующих* - для формулы предобработки этапа согласования, позволяет в скрипт добавить системную переменную @[#participants] в которой определен список возможных согласующих.

Для применения атрибутов и функций в формуле достаточно «перетянуть» их с панели мышью в область скрипта.

3.12.21 Маршрут согласования. Редактор этапов

Маршрут согласования предназначен для указания последовательности согласования заявки перед её выполнением. Существует два способа получения маршрута.

В первом случае в разделе **Маршрут согласования** шаблона заявки можно установить маркер **Получать маршрут из изменённых групп**. В этом случае маршрут будет браться из той справочной, в которой производятся изменения позиций по текущей заявке.

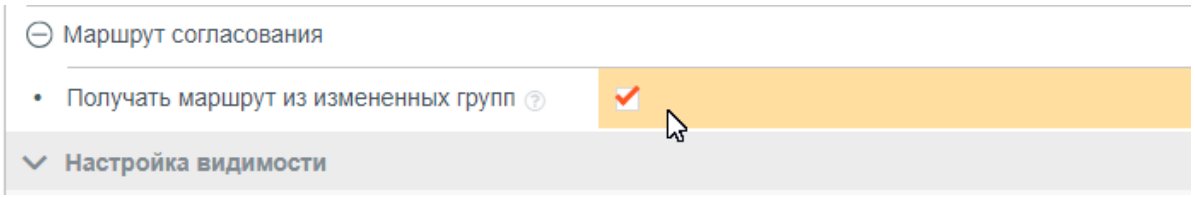


Рисунок 161 - Маршрут согласования по редактируемому справочнику

Второй способ предусматривает настройку маршрута для заявок, создаваемых по текущему шаблону.

Для настройки маршрута согласования непосредственно в справочнике и для настройки в шаблоне заявки есть два способа - описание этапов в карточке шаблона и графический (workflow) через **Редактор этапов**. Способы полностью взаимозаменяемы по функциональности. Маршруты, настроенные с помощью workflow будут отображаться в карточке шаблона и наоборот, маршруты из карточки будут иметь графическое представление.

3.12.22 Разработка маршрута согласования в карточке шаблона

Для создания маршрута согласования (Рисунок 162) необходимо:

- в разделе **Этапы согласования** нажать кнопку *Добавить этап* или кнопку *Редактор этапов*;

- созданному этапу присвоить наименование в ячейке **Наименование**. Оно должно быть уникальным в рамках данного маршрута согласования;
- **Статус** - параметр настройки этапа со следующими значениями: *Неактивный*, *Активный*, *Архив*. Этап согласования участвует в расчете маршрута, только если он имеет статус *Активный*. Значение статуса выбирается из выпадающего списка. Статусы *Неактивный* и *Архив* применяются для отключения этапа согласования для редактирования или в связи с изменением маршрута. При этом сам этап сохраняется в истории и может быть повторно использован в дальнейшем;
- опция **Возможность редактирования** разрешает участникам этого этапа редактировать объекты заявки в справочных группах, если маркер не установлен, то участники этапа могут только согласовывать заявку. Доступность опции зависит от выбранного вида согласования;

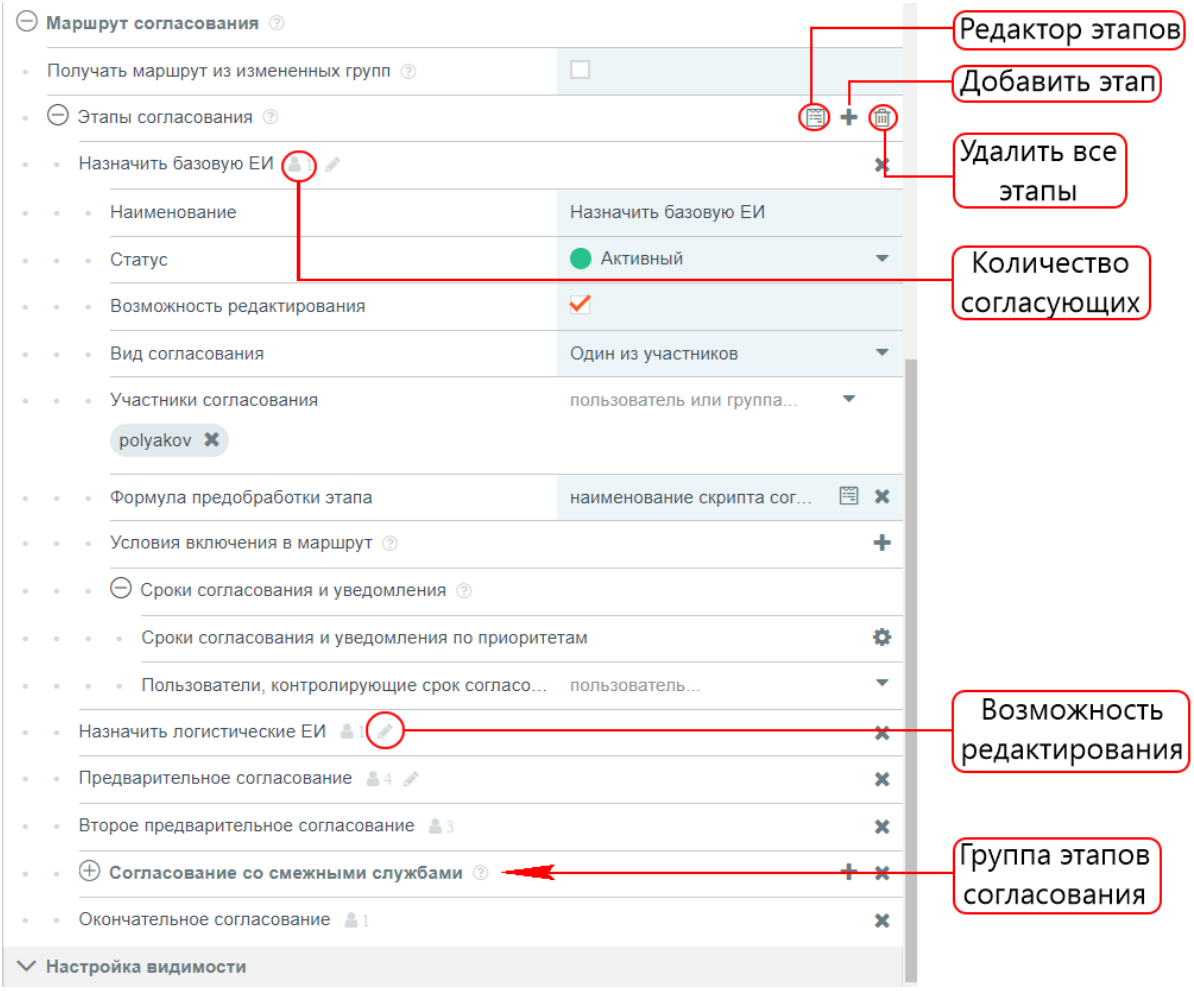


Рисунок 162 - Настройка маршрута согласования в карточке шаблона заявки

- в ячейке **Вид согласования** можно выбрать вариант согласования:
  - *Один из участников* - допускается добавление в список участников согласования как отдельных пользователей, так и групп пользователей в заявках, на основе которых формируется единый список участников согласования. Опция **Возможность редактирования** доступна;
  - *Все участники* - допускается добавление в список участников согласования как отдельных пользователей, так и групп пользователей в заявках, на основе которых формируется единый список участников согласования. Заявка не будет считаться согласованной, пока все участники её не согласуют. Для этапа согласования с данным видом согласования опция **Возможность редактирования** не доступна;

- *Один из участников каждой группы* - допускается добавление в список участников согласования только групп пользователей в заявках. Опция **Возможность редактирования** для данного вида согласования недоступна;

- в ячейке **Участники согласования** необходимо выбрать из выпадающего списка участников этапа (одного или нескольких) либо группу пользователей в заявках. Чтобы удалить участника или группу достаточно нажать на ✖ рядом с именем. Кнопка 🗑 ***Удалить всех*** позволяет удалить всех участников;
- **Формула предобработки этапа** – позволяет разработать в **Редакторе формул** скрипт Groovy, который обеспечивает:
  - подбор участников согласования с возможностью добавления новых участников, которые ранее не были определены в списке;
  - подбор участников контролирующих согласование с возможностью добавления новых участников, которые ранее не были определены в списке;
  - автоматическое согласование этапа с фиксацией этого факта в истории заявки;
  - автоматическое отклонение этапа согласования с возможностью указать на какой этап возвращать заявку, на этап исполнение или на заданный этап согласования;
- **Условия включения в маршрут** - настройка условий включения этапа в маршрут согласования заявки, выполняемая в отдельном редакторе.

Если маршрут согласования должен обеспечить возможность параллельного согласования заявки на нескольких этапах сразу, то для этого эти этапы необходимо объединить в группу. Количество групповых этапов в шаблоне не ограничено, при этом группы указываются последовательно и не могут включаться друг друга. Один этап может быть включен только в одну

группу. Завершением согласования группы этапов, будет являться согласование всех этапов, включённых в группу.

Первоначальное создание группы этапов (Рисунок 163) согласования в шаблоне заявки выполняется путём перетаскивания (функция Drag&Drop) мышью и совмещения наименований ранее созданных этапов. После создания группового этапа, дополнительные этапы в группу добавляются кнопкой ***Добавить этап***.

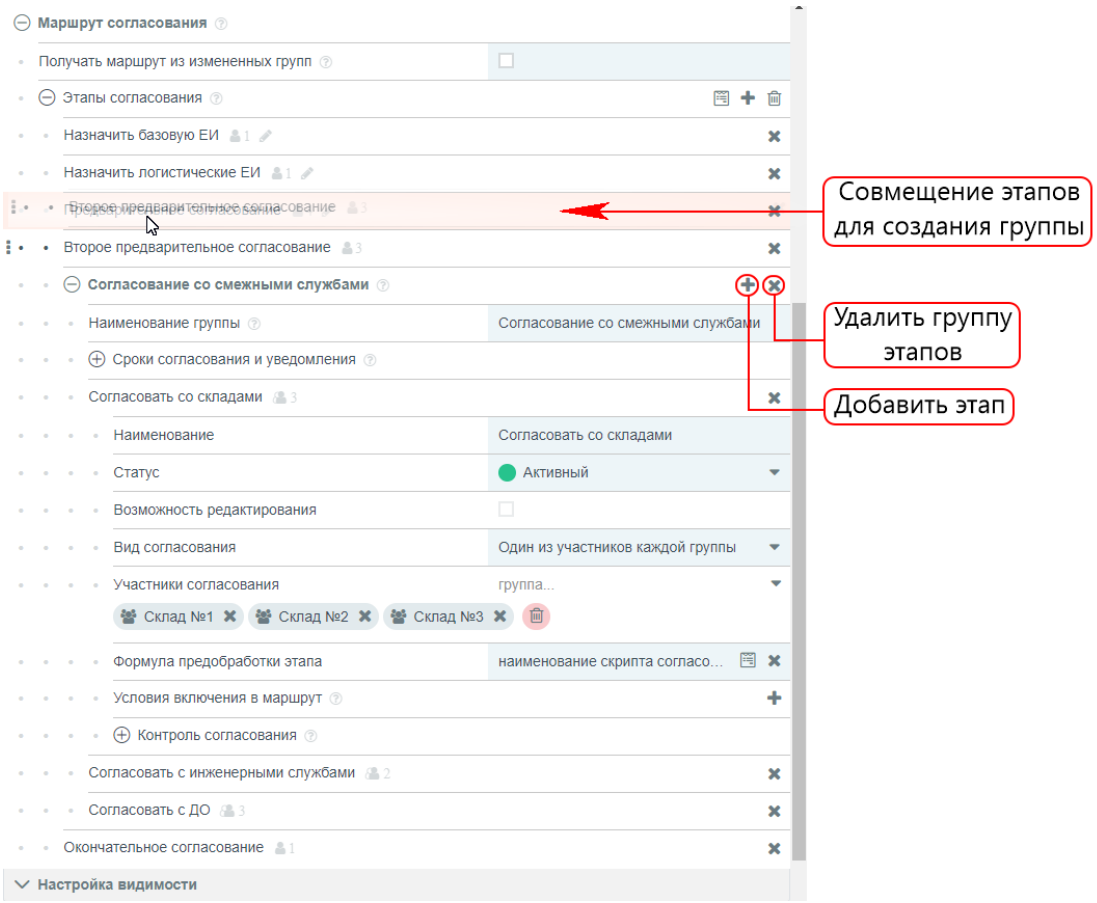


Рисунок 163 - Создание группового этапа

Для группы этапов должны настраиваться следующие параметры:

- *Наименование группы* – наименование группы, объединяющей этапы согласования;
- Сроки создания и уведомления по приоритетам.

Настройки каждого отдельного этапа, входящего в групповой, выполняются стандартными способами, описанными выше в этом пункте.

3.12.23 Разработка маршрута согласования в Редакторе этапов

Разработка маршрута согласования в графическом виде (Рисунок 164) подчиняется тем же правилам, которые применяются для табличного описания. При необходимости можно совмещать табличную и графическую разработку маршрута.

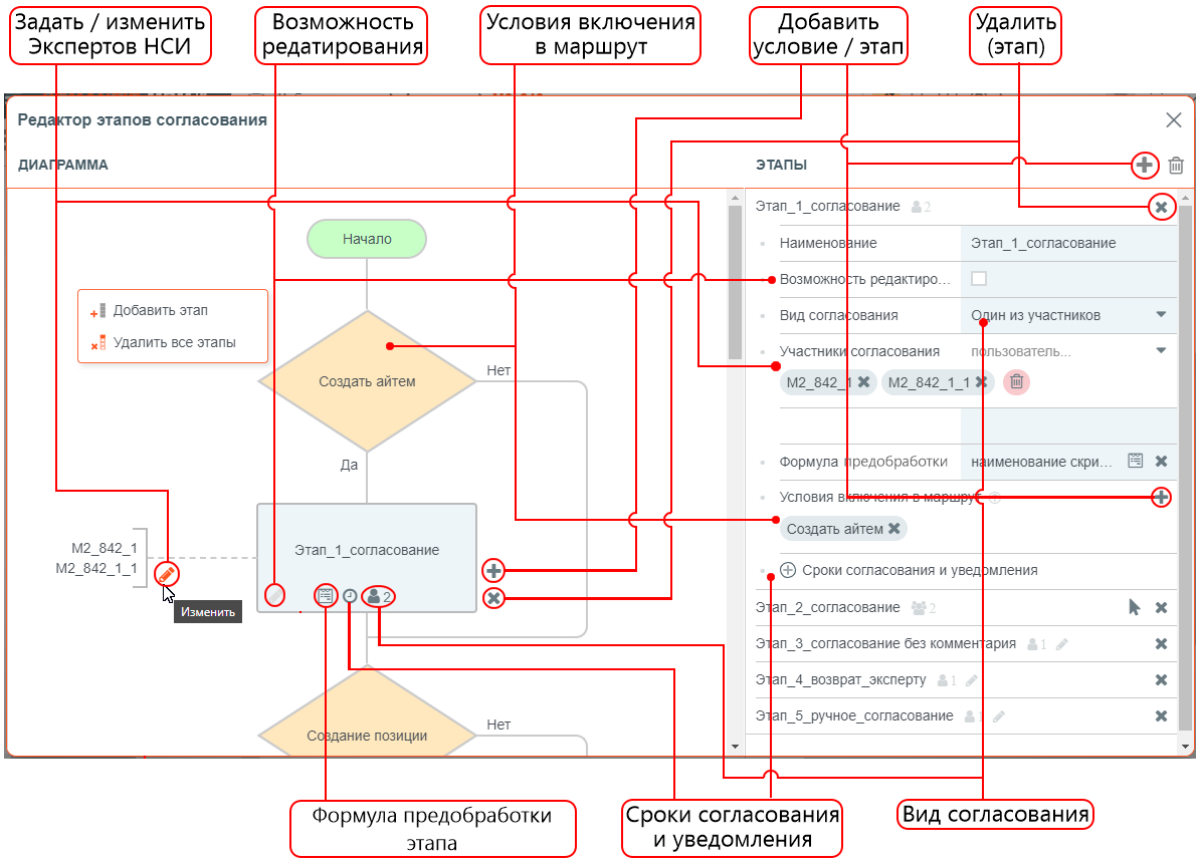
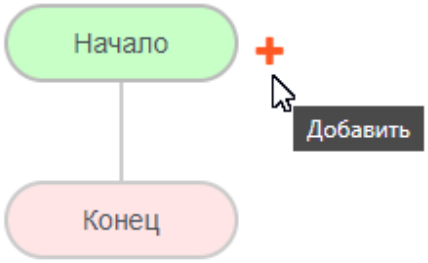


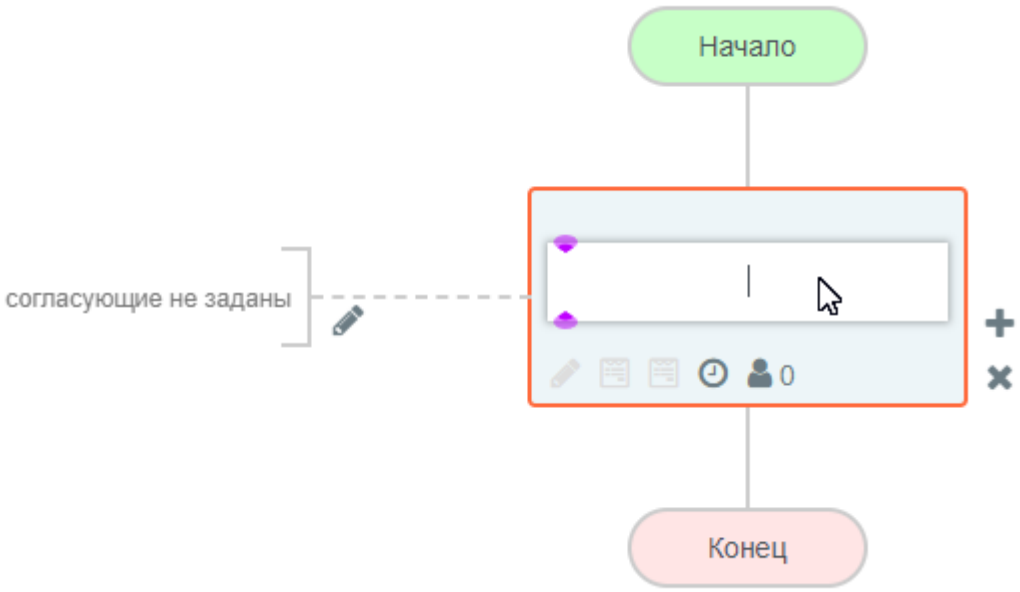
Рисунок 164 - **Редактор этапов согласования** с маршрутом

Чтобы маршрут согласования в **Редакторе этапов** необходимо:

- в разделе Этапы согласования нажать кнопку *Редактор этапов*;
- в открывшемся Редакторе этапов согласования нажать кнопку *Добавить* рядом с графическим компонентом, кнопку *Добавить этап* на панели Этапы или вызвать соответствующую команду из контекстного меню в поле диаграммы;



- созданному этапу присвоить наименование;



- включить, при необходимости, опцию **Возможность редактирования**, которая разрешает участникам этого этапа редактировать объекты заявки в справочных группах, если маркер не установлен, то участники этапа могут только согласовывать заявку;
- Используя пиктограмму слева от поля этапа выбрать согласующих заявку (одного или нескольких);
- Используя пиктограммы в поле этапа вызвать редактор для **Формулы предобработки** этапа;
- Также с использованием пиктограмм в поле этапа установить Сроки согласования и уведомления - настройка производится во всплывающем окне и Вид согласования - переключатель **Один из участников** 3 / **Все участники** 3;

- С помощью кнопки **Добавить условие** ➕ (появляется рядом с элементом этапа справа) добавить условие включения этапа (см. п 3.12.24) в маршрут согласования. Созданный элемент располагается над этапом и может быть отредактирован или удален.

Следующий созданный этап располагается под предыдущим, если этапы нужно поменять местами, то это можно сделать на панели Этапы справа от диаграммы путем перетаскивания (Drag&Drop).

Создать групповой этап можно только в табличном представлении маршрута согласования. На графической схеме группа этапов согласования (Рисунок 165) отображается в виде вытянутого по горизонтали прямоугольника с наименованием группового этапа левом верхнем углу. Все этапы согласования добавляются по горизонтали и вписываются в прямоугольник группы, включая условия, настроенные для каждого этапа согласования. Если для этапа согласования настроен вид согласования *Один из участников каждой группы*, то для этапа согласования будет отображена специальная пиктограмма, а слева, перечислены наименования групп, объединённые квадратной скобкой справа. При наведении курсора мыши на наименование группы пользователей появляется всплывающее окно со списком пользователей, согласующих этап от имени группы. Если для группы пользователей в заявках не определен список пользователей, то наименование группы будет подсвечено красным цветом и выведена пиктограмма с белым восклицательным знаком в красном кружке.

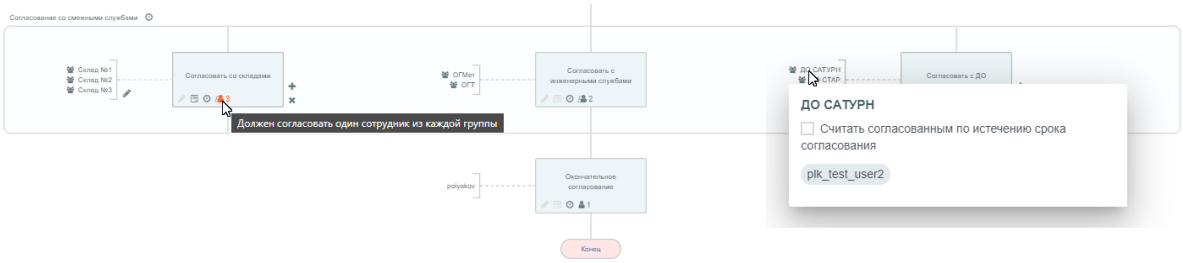


Рисунок 165 - Группа этапов согласования на графической схеме

Удаление этапа / условия выполняется:

- кнопкой ***Удалить*** рядом с элементом диаграммы;
- кнопкой ***Удалить*** в ячейке этапа справа от наименования в карточке;
- командой из контекстного меню, вызванного в поле диаграммы (Рисунок 166).

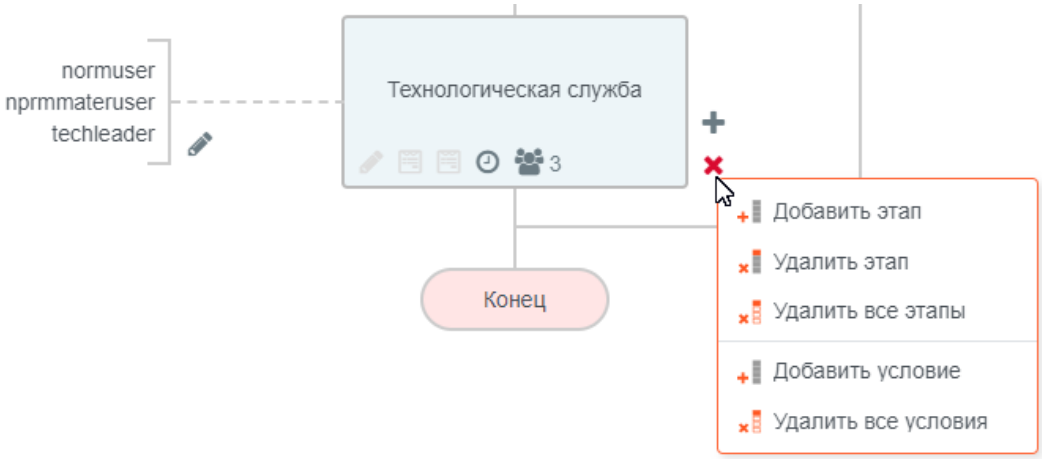


Рисунок 166 – Контекстное меню на графической схеме согласования

Удаление всех этапов согласования может быть выполнено через контекстное меню в поле диаграммы или кнопкой ***Удалить все этапы***, расположенной в правом верхнем углу рабочего окна **Редактора этапов**.

3.12.24 Условия включения в маршрут

Настройка условий включения этапа согласования в маршрут выполняется в специальном окне **Условия включения этапа**, которое вызывается кнопкой ***Добавить условие***. Если на этап назначено несколько условий, то для включения этапа в маршрут должно выполняться хотя бы одно из них.

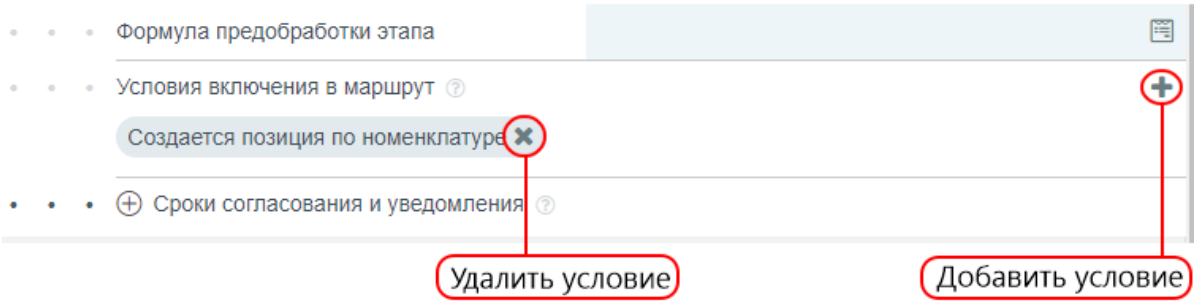


Рисунок 167 - Добавление и удаления условия включения этапа в маршрут в шаблоне заявки

Настройка условий выполняется по различным основаниям, данные по которым располагаются на отдельных вкладках:

- **Изменения данных** - настраивается выполнение проверки изменений, выполненных для позиций;
- **Изменения модели данных** - настраивается выполнение проверки изменений, выполненных в метаданных:
  - состава деклараций;
  - формулы вычисления атрибута;
  - постобработчиков;
  - ограничений;
- **Условия на заявку** - настраивается выполнение проверки на наличие связи с другой заявкой и условий на значение атрибутов в этой заявке;
- **Groovy-скрипт** - используется для разработки скрипта, который реализует необходимые условия проверки.

Процедура настройка условия включения в маршрут выполняется следующим образом:

- нажать кнопку *Добавить условие* в строке Условия включения в маршрут;
- в открывшемся окне Условия включения этапа последовательно (при необходимости) настроить:
- <наименование не указано> – название создаваемого условия. Редактируется после нажатия на кнопку *Изменить*;
- на вкладке Изменения данных (Рисунок 168):

- **Действия над позицией** – условие включения в маршрут по выполняемым действиям – создание позиции, удаление позиции или изменение атрибутов. Условие изменения атрибутов действует, если изменён хотя бы один из атрибутов, указанных в списке. Если атрибуты не заданы, то учитываются изменения любых атрибутов;

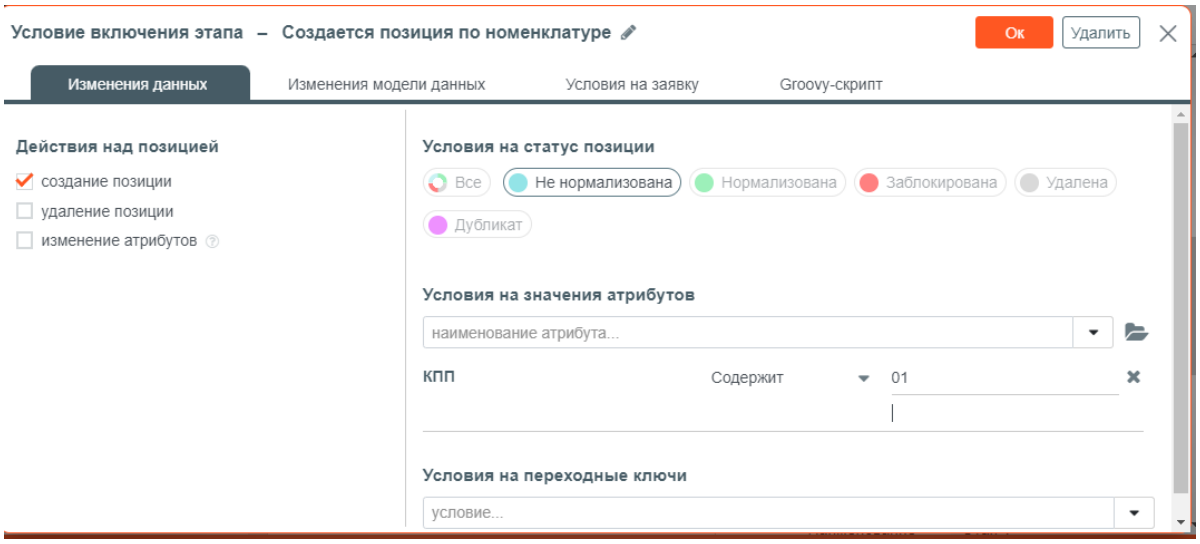


Рисунок 168 - Настройка условий включения этапа по изменению данных

- **Условия на статус позиции** – этап согласования будет действовать, если позиция (группа) в заявке имеет указанный статус (статусы). Если статусы не указаны, то это значит, что статус может быть любым;
- **Условия на значения атрибутов** – список атрибутов, наличие и содержание которых учитывается при выборе этапа согласования. Атрибуты могут быть выбраны из выпадающего списка или с помощью **Классификатора атрибутов** (кнопка ***Выбрать атрибут***). На выбранное значение могут быть назначены дополнительные условия из выпадающего списка;
- **Условия на переходные ключи** (Рисунок 169) – позволяет включить этап согласования, если на объект НСИ есть переходной ключ с какой-либо внешней ИС.

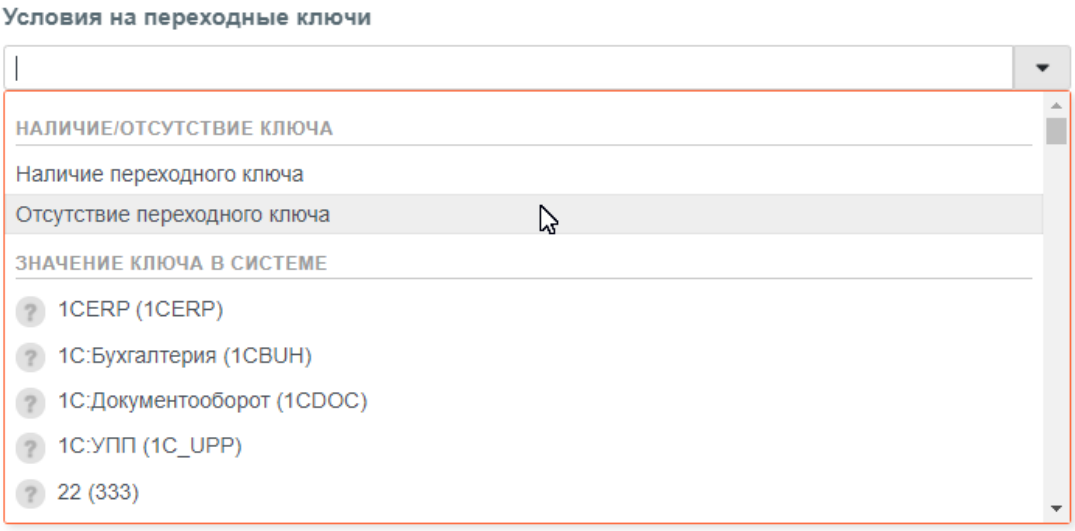


Рисунок 169 - Настройка условия на переходные ключи

В условия на значения атрибута не могут быть включены следующие типы данных: массивы, изображение, ссылка на композицию, многострочный текст, файл, формула, цвет.

- на вкладке **Изменения модели данных** (Рисунок 170) выполняются следующие настройки:
  - **Изменение деклараций** в составе:
    - *Изменение состава деклараций атрибутов* – опция, с помощью которой определяется наличие изменений в составе деклараций группы, за исключением изменения параметров существующей декларации;
    - *Изменение формулы вычисления атрибута* – опция, с помощью которой определяется наличие изменений формулы в декларации атрибута группы. Для данной опции можно задать атрибуты, по которым требуется отслеживать изменение формул. Атрибуты выбираются из выпадающего списка в ячейке **Атрибуты, для которых отслеживать изменение формулы**. В списке поддерживается нечеткий поиск по вводимым символам.

Если атрибуты не определены, то отслеживаются изменения формул в любой декларации атрибута. Изменением также считается формула, созданная во вновь добавленной или удалённой декларации атрибута;

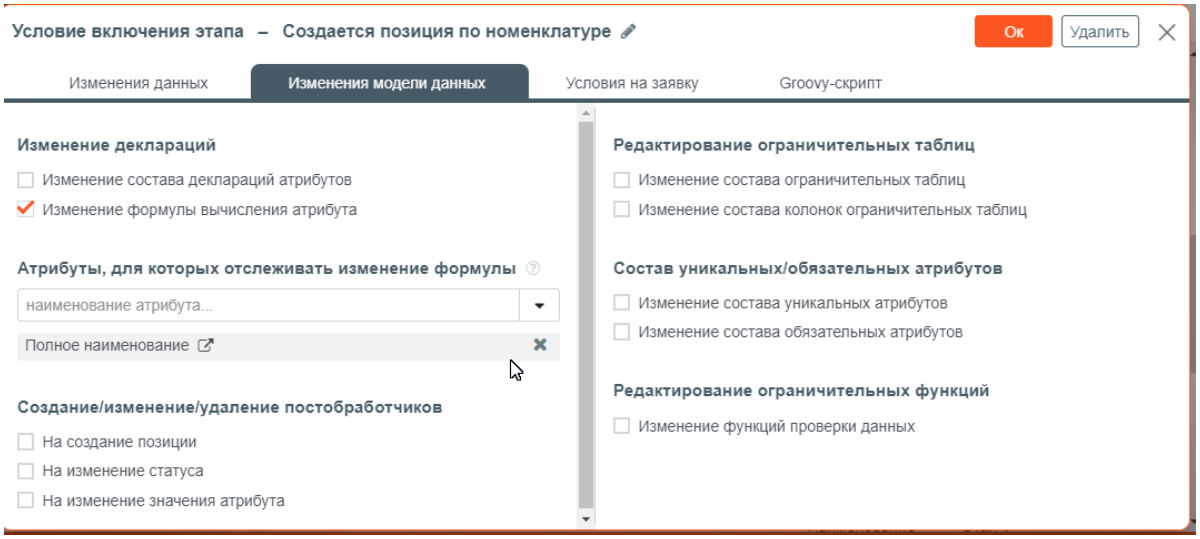


Рисунок 170 - Настройки условий на изменение модели данных

- **Создание / изменение / удаление постобработчиков** – группа, состоящая из следующих опций:
  - *На создание позиции* – опция, с помощью которой определяется создание / удаление / редактирование постобработчика на создание позиции;
  - *На изменение статуса* – опция, с помощью которой определяется создание / удаление / редактирование постобработчика на изменение статуса;
  - *На изменение значения атрибута* - опция, с помощью которой определяется создание / удаление / редактирование постобработчика на декларацию атрибута. Редактированием постобработчика, также считается постобработчик в созданной / удалённой декларации атрибута;

- **Редактирование ограничительных таблиц** - группа, состоящая из следующих опций:
    - *Изменение состава ограничительных таблиц* - опция, с помощью которой определяется добавление, изменение или удаление ограничения в виде ограничительной таблицы;
    - *Изменение состава колонок ограничительных таблиц* – опция, с помощью которой определяется добавление / удаление колонки в ограничительной таблице;
  - **Состав уникальных / обязательных атрибутов** содержит опции:
    - *Изменение состава уникальных атрибутов* – опция, с помощью которой определяется добавление, изменение или удаление ограничения на уникальный атрибут;
    - *Изменение состава обязательных атрибутов* – опция, с помощью которой определяется добавление, изменение или удаление ограничения на обязательный атрибут;
- **Редактирование ограничительных функций:**
    - *Изменение функции проверки данных* – опция, с помощью которой определяется добавление, удаление или редактирование ограничительной функции.
- на вкладке **Условия на заявку** (Рисунок 171) выполняются следующие настройки:
    - *Наличие связи с шаблоном заявки* – с помощью данной опции определяется наличие связи пакета изменения с одним из выбранных шаблонов заявки. Кроме проверки наличия связи с шаблоном заявки, обеспечивается возможность наложения условий на значения атрибутов заявки;
    - *Условия на значения атрибутов заявки* – для каждого из шаблонов можно выбрать необходимые атрибуты заявки и настроить условия сравнения. Предусмотрены следующие условия сравнения: «Равно», «Не равно», <Пустое значение>, <Не пустое

значение>. Для типа данных «Текст» и «Файл» доступны только два последних условия. Для множественных атрибутов, предусмотрены дополнительные условия: «содержит» (содержит все заданные значения в атрибуте заявки), «не содержит»;

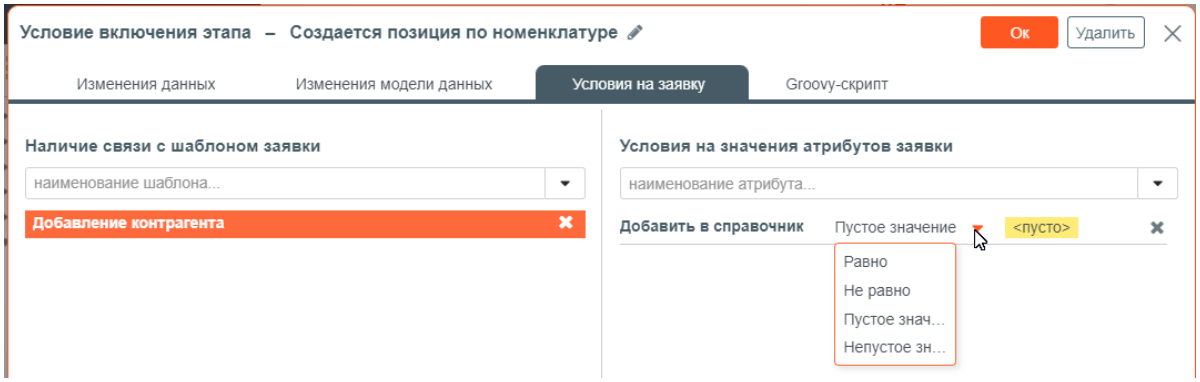


Рисунок 171 - Настройки условия на заявку для этапа маршрута согласования

·На вкладке **Groovy-скрипт** расположен стандартный редактор скрипта (Рисунок 172).

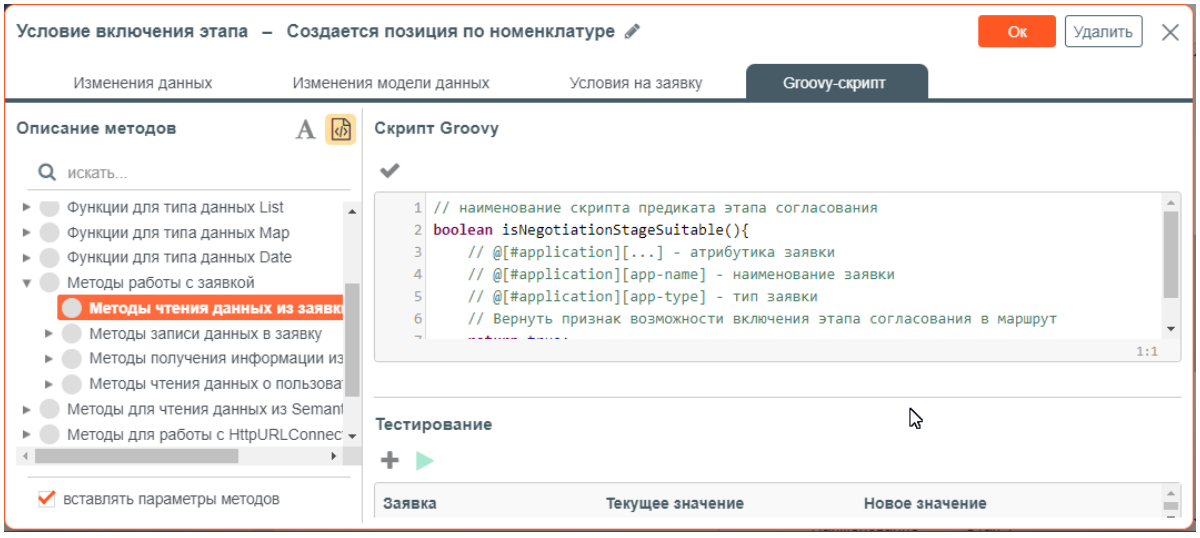


Рисунок 172 - Настройки условия включения этапа согласования в маршрут через скрипт Groovy

3.12.25 Сроки обработки и согласования заявок. Уведомления о сроках

При выполнении настроек экспертов НСИ и этапов согласования заявки (Рисунок 173) необходимо указать сроки исполнения и согласования заявок, а также контроля этих сроков. Контроль времени исполнения учитывает

настройки приоритетов заявок, рабочих и выходных дней, а также режим работы экспертов НСИ.

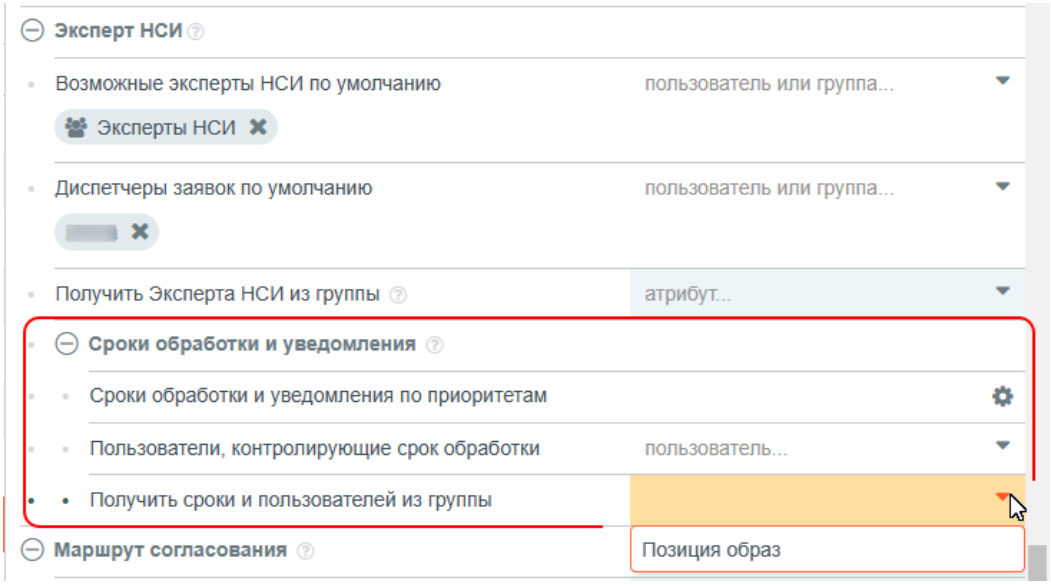


Рисунок 173 - Настройка сроков обработки и уведомления

В разделе **Эксперт НСИ** нужно указать следующие значения:

- **Сроки обработки и уведомления по приоритетам** (Рисунок 174) – время в минутах / часах / днях, в течение которого заявка по данному шаблону должна быть обработана исполнителем, а также время, при наступлении которого исполнитель и лица, контролирующие исполнение, получают уведомление о приближении регламентного времени окончания обработки. Настройка единицы измерения времени выполняется через список, появляющийся при наведении курсора на ЕИ в поле с обозначением приоритета;

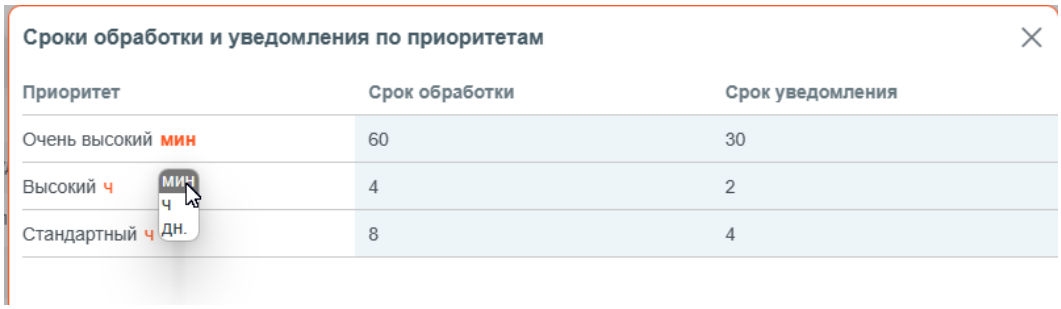


Рисунок 174 - Настройка времени согласования и уведомления по приоритетам в шаблоне заявки

- **Пользователи, контролирующие срок обработки** – один или несколько пользователей или групп пользователей в заявках, которые выбираются из выпадающего списка. Пользователи, контролирующие срок обработки, получают уведомления о приближении окончания сроков обработки заявок, а также о просроченных заявках и принимают административные решения в отношении экспертов НСИ;
- **Получить сроки и пользователей из группы** – если для шаблона используется хотя бы один атрибут заявки – ссылка на справочную группу или позицию, то список контролирующих пользователей и сроки обработки заявки могут быть получены из неё. Группа выбирается из выпадающего списка атрибутов заявки.

В разделе **Этапы согласования** выполняются настройки следующих значений:

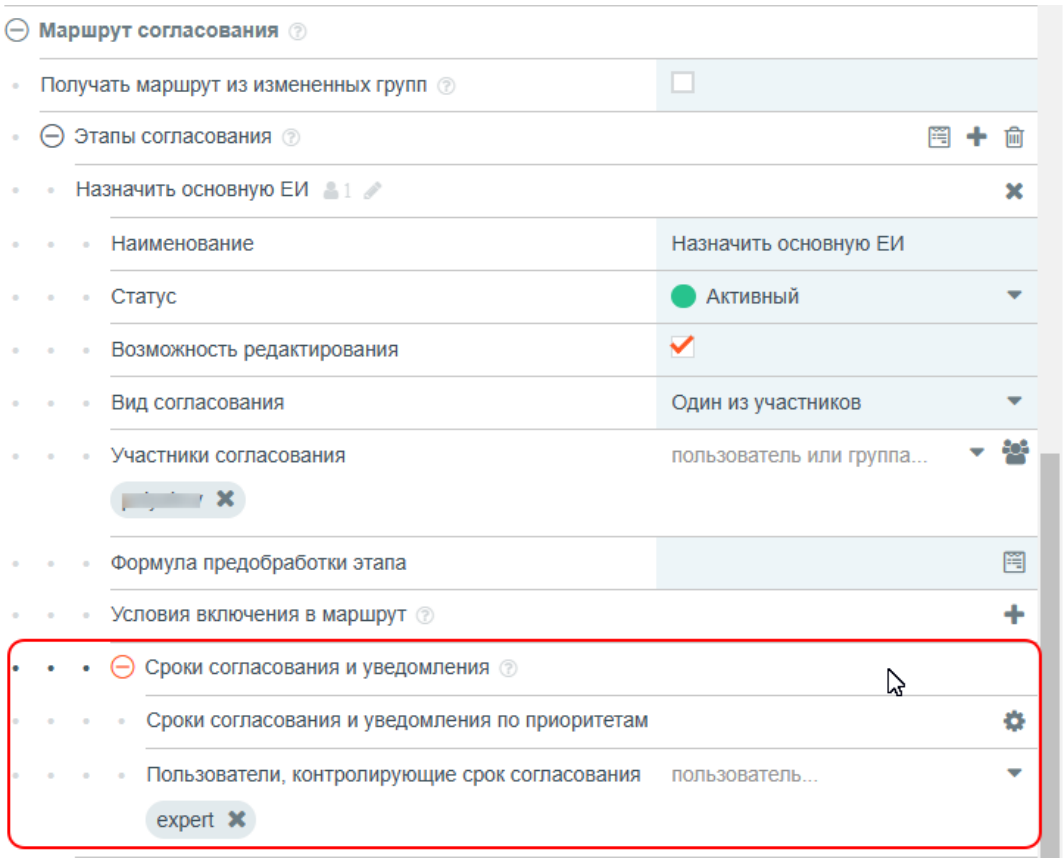


Рисунок 175 - Настройка сроков согласования заявок

- **Сроки согласования и уведомления по приоритетам** – время в минутах / часах / днях, в течение которого заявка по данному шаблону должна быть согласована , а также время, при наступлении которого согласующий и лица, контролирующие сроки согласования, получают уведомление о приближении регламентного времени окончания согласования. Настройка единицы измерения времени выполняется через список, появляющийся при наведении курсора на ЕИ в поле с обозначением приоритета;
- **Пользователи, контролирующие срок согласования** – один или несколько пользователей, которые выбираются из выпадающего списка. Пользователи, контролирующие срок согласования, получают уведомления о приближении окончания сроков согласования заявок, а также о просроченных заявках и принимают административные решения в отношении согласующих;

В случае если этап обработки заявки не завершён в установленные сроки, то такая заявка перемещается на вкладку **Просроченные заявки**, а уведомление о просрочке поступает контролирующему лицу, если этонастроено в шаблоне

3.12.26 Аннулирование заявок

В Системе можно настроить как индивидуальные сроки аннулирования для каждого шаблона заявки, так и общие, распространяющиеся на все шаблоны заявок. Настройки выполняются в подсистеме **Конфигурация данных.** Аннулирование заявок позволяет избежать накопления в Системе «мусора» в виде созданных и не отправленных заявок, возвращенных заявок, а также заявок и, связанных с ними, пакетов изменений, которые не получили движения по различным причинам.

Для вызова диалога настройки сроков аннулирования заявок нужно нажать на основной панели инструментов кнопку ***Настройка сроков аннулирования заявок*** (Рисунок 176).

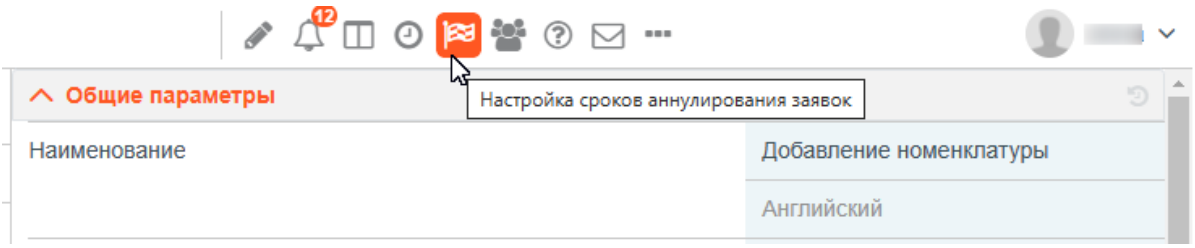
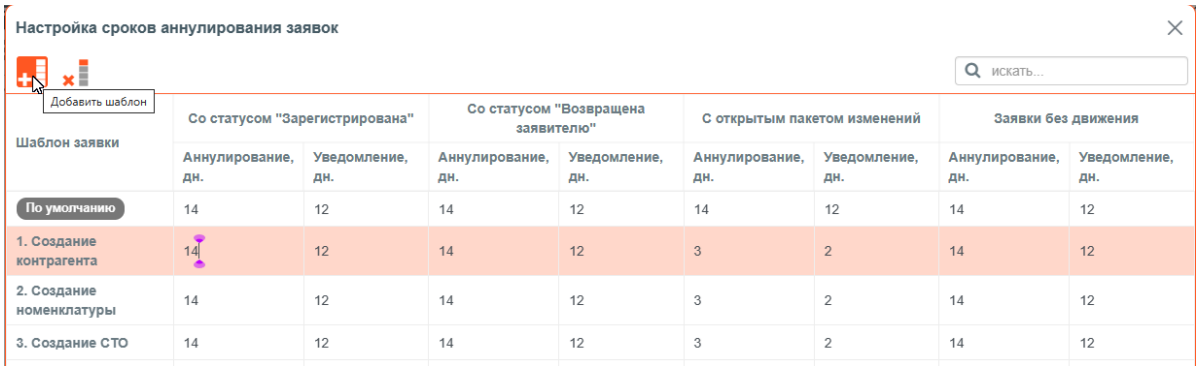


Рисунок 176 - Вызов настройки сроков аннулирования заявок

В диалоге настройки сроков аннулирования (Рисунок 177) доступны следующие виды аннулирования заявок:

- Аннулирование заявок со статусом «Зарегистрирована» – позволяет указать срок, который заявка может находится на статусе «Зарегистрирована»;
- Аннулирование заявок со статусом «Возвращена заявителю» – позволяет указать срок, который заявка может находится на статусе «Возвращена заявителю», при условии отсутствия связанного с ней неопубликованного пакета изменений;
- Аннулирование заявок с открытым пакетом изменений – позволяет указать срок, который заявка может находится без движения с неопубликованным пакетом изменений;
- Аннулирование заявок, по которым нет движения – позволяет указать какой срок может заявка находится без движения, при условии отсутствия связанного с ней неопубликованного пакета изменений.



Шаблон заявки	Со статусом "Зарегистрирована"		Со статусом "Возвращена заявителю"		С открытым пакетом изменений		Заявки без движения	
	Аннулирование, дн.	Уведомление, дн.	Аннулирование, дн.	Уведомление, дн.	Аннулирование, дн.	Уведомление, дн.	Аннулирование, дн.	Уведомление, дн.
По умолчанию	14	12	14	12	14	12	14	12
1. Создание контрагента	14	12	14	12	3	2	14	12
2. Создание номенклатуры	14	12	14	12	3	2	14	12
3. Создание СТО	14	12	14	12	3	2	14	12

Рисунок 177 - Диалог настройки сроков аннулирования заявок

При настройке сроков аннулирования заявок указывается два значения - срок аннулирования в днях, а также срок уведомления в днях, который определяет за сколько дней до момента аннулирования пользователь (заявитель, исполнитель, согласующий) будет предупрежден о наступлении события.

Строка *По умолчанию* определяет сроки аннулирования для всех заявок *Системы*.

Для настройки сроков аннулирования для конкретного шаблона заявки нужно, при помощи кнопки ***Добавить шаблон***, добавить шаблон заявки и заполнить необходимые сроки. Сроки, не заполненные для шаблона заявки, берутся из значений по умолчанию. Удаление индивидуально настроенных сроков для шаблона выполняется кнопкой ***Удалить шаблон***.

Ячейка быстрого поиска позволяет найти шаблон заявки в окне настройки сроков аннулирования.

3.12.27 Настройка видимости

В разделе **Настройка видимости** выбираются пользователи и роли, для которых данный шаблон будет доступен. Роли и пользователи выбираются из выпадающего списка.

3.12.28 Экспорт / импорт шаблона заявки через XML

В Системе существует возможность экспорта / импорта шаблона заявки через файл формата XML. При выполнении экспорта выполняется сохранение в XML следующих данных:

- классификационная группа для шаблона,
- настройки шаблона заявки,
- настройки видимости шаблона заявки,
- атрибуты заявки,
- атрибуты позиции,
- исключаемые атрибуты позиции,

- назначенные диспетчеры,
- эксперты НСИ,
- пользователи контролирующие срок обработки,
- группы пользователей МУЗ с их пользователями.

При выполнении импорта, сохраненные группы пользователей МУЗ, пользователи, роли и настройки видимости воссоздаются опционально. Остальные данные полностью заменяются в шаблоне заявки, если таковой существовал.

Для выполнения экспорта необходимо установить курсор на наименовании шаблона в таблице и вызвать из меню основной панели команду **Экспорт в XML** (Рисунок 178).

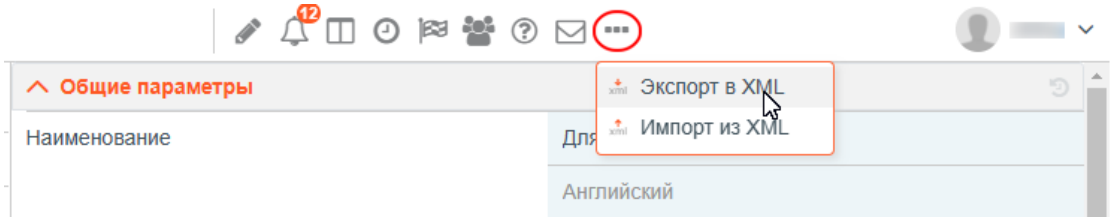


Рисунок 178 - Вызов команд экспорта и импорта в / из XML

Для выполнения импорта шаблона следует установить курсор на нужной группе шаблонов и вызвать из меню основной панели команду **Импорт из XML**. В открывшемся окне диалога (Рисунок 179) установить необходимые маркеры и нажать кнопку ***Импортировать***.

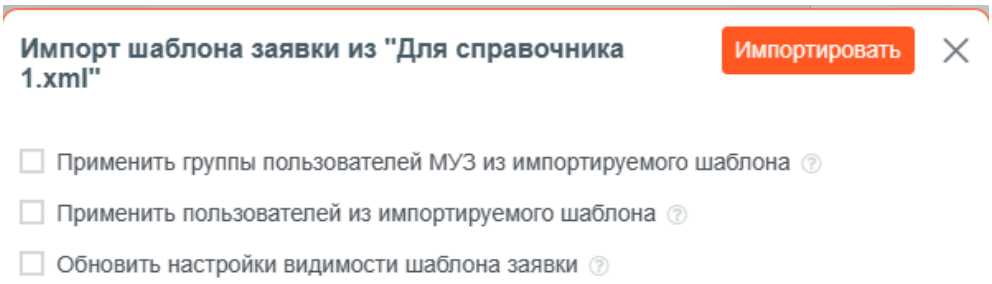


Рисунок 179 - Настройки импорта шаблона заявки из XML

3.13 Редактирование справочных данных

3.13.1 Управление пакетами изменений

Для внесения изменений в любые информационные объекты *Системы* пользователю необходимо обладать соответствующими правами (см. п. 3.28.1).

Любые изменения возможны только в режиме редактирования, который включается кнопкой ***Управление пакетами изменений*** -> ***Создать***, расположенной в верхней строке рабочего окна *Semantic MDM* (Рисунок 180).

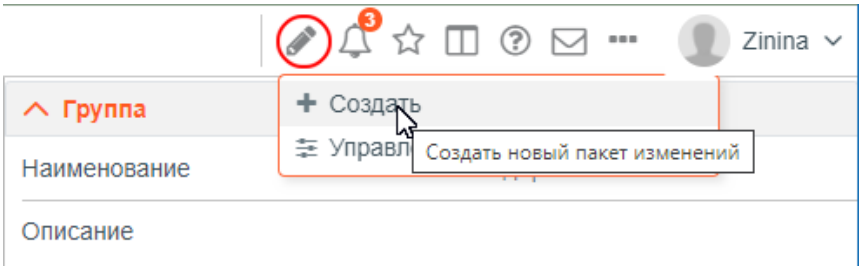


Рисунок 180 - Включение режима **Управление пакетами изменений**

После включения режима редактирования кнопка окрашивается в оранжевый цвет, а в меню кнопки становятся доступными команды (Рисунок 181).

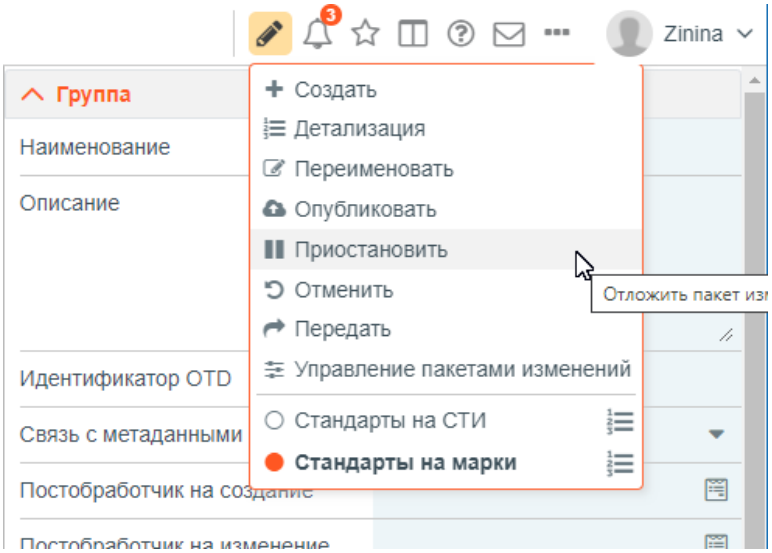


Рисунок 181 - Команды, доступные в режиме редактирования

Команды, доступные в режиме редактирования:

- **Детализация** – позволяет просмотреть в отдельном окне произведённые в данном пакете изменения объектов (см. п. 3.13.12);
- **Переименовать** - запускает окно **Информация о пакете изменений**, где в ячейке **Наименование** можно изменить название пакета, созданное по умолчанию (по умолчанию наименование пакета изменений соответствует названию справочной группы, в которой он был создан);
- **Опубликовать** – фиксирует изменения и открывает доступ всех пользователей *Системы* к изменённым данным;
- **Приостановить** – обеспечивает перерыв во внесении изменений без опубликования результатов. Работа с пакетом может быть возобновлена в любой момент;
- **Отменить** – отменяет все, произведённые в рамках текущего пакета, изменения и прекращает режим редактирования;
- **Передать** - открывает окно **Выберите пользователя**, где можно выбрать пользователя *Системы* для передачи ему пакета изменений. Передать можно только пакет изменений, не связанный с заявкой;
- **Управление пакетами изменений** – открывает дополнительное окно с историей всех пакетов изменений.

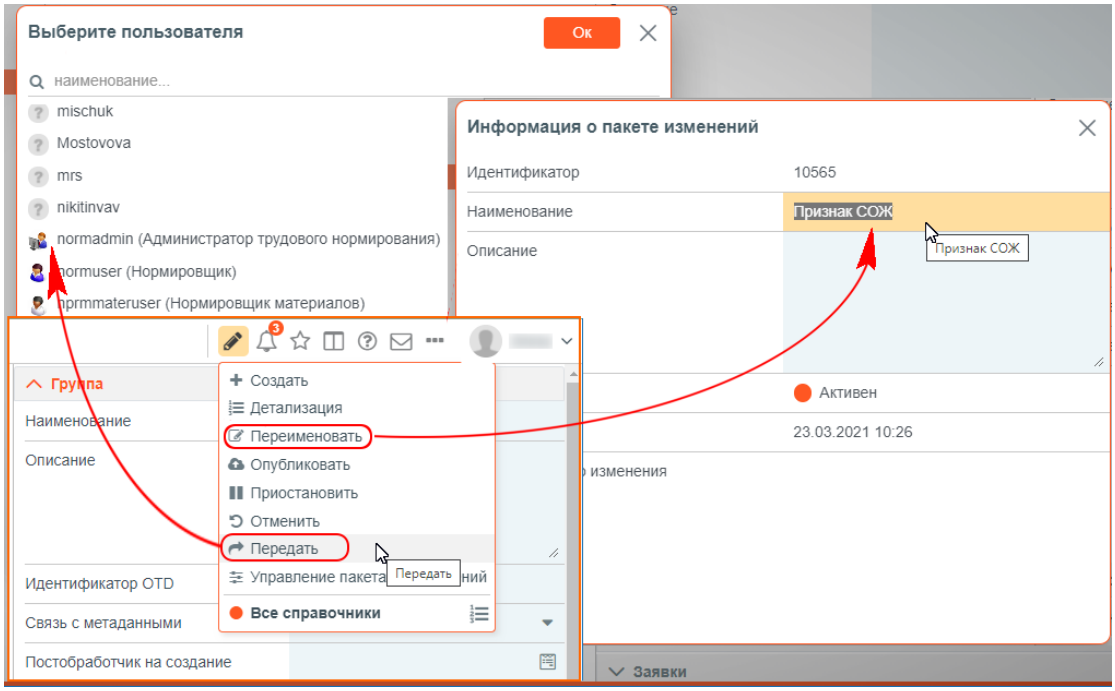


Рисунок 182 - Переименование и передача пакета изменений

Ниже команд для работы с пакетом изменений присутствует перечень пакетов, находящихся в работе. Текущий пакет изменений имеет соответствующий статус, отмеченный пиктограммой l. Для каждого пакета доступна кнопка ***Детализация*** , нажав на которую можно ознакомиться с подробностями о внесённых изменениях.

Если публикуемый пакет изменений не связан с заявкой, то пользователю будет выдано окно для ввода комментария к публикуемому пакету изменений.

В случае внесения изменений другим пользователем в группы, декларации, позиции или значения атрибута, для текущего пользователя в системе будет отображен признак блокировки данного элемента. При наведении курсора на признак блокировки, будет отображена подсказка с логином пользователя, который в данный момент выполняет изменение.

3.13.2 Согласование пакетов изменений

Для пользователей, у которых включена привилегия **Редактирование данных**, но выключена привилегия **Редактирование данных без заявки** автоматически включается режим редактирования, при котором в начале создается пакет изменений и выполняются изменения данных, после чего требуется выполнить его обязательное согласование в соответствии с настроенными маршрутами согласования для справочных групп, в которых были выполнены изменения.

Сценарий работы такого пользователя будет выглядеть следующим образом:

1. Пользователь нажимает кнопку ***Управление пакетами изменений*** -> ***Создать***.
2. *Система* создает автоматически связанную с ним заявку по следующему шаблону:
   - **Наименование заявки** – «Редактирование данных»;
   - **Тип заявки** – «Редактирование данных»;
   - **Заявитель** – текущий пользователь;
   - **Эксперт НСИ** – текущий пользователь;
   - **Маршрут согласования** – включено получение маршрута согласования из измененных групп;
   - **Срок обработки** – не устанавливается;
   - В атрибутах заявки имеется атрибут *Описание*, в котором пользователь может вписать поясняющий текст для согласующих.
3. Система переводит заявку в статус «В работе».
4. Пользователь выполняет изменение данных в рамках созданного пакета изменений и системной заявки. При необходимости заполняет поле *Описание* в заявке.
5. Пользователь отправляет пакет изменения на согласование по кнопке ***Управление пакетами изменений*** -> **На согласование**, либо отменяет заявку и пакет по кнопке ***Управление пакетами изменений*** -> **Отменить**.

3.13.3 Редактирование значений атрибутов в карточке объекта

Каждый справочный объект описывается набором характеристик – атрибутов. Состав атрибутов, описывающий конкретный объект, формируется на уровне родительского объекта (группы) и распространяется на все дочерние объекты, располагающиеся в таблице (на сетке) данных.

Для изменения значения атрибута необходимо:

- найти требуемый объект и установить на нем курсор в таблице;
- создать новый пакет изменений;
- перейти в карточку позиции;
- внести необходимые изменения. В зависимости от типа атрибута значение может вводиться вручную, в том числе копированием, создаваться в виде ссылки на значение из другого справочника (ветки справочника) или выбираться из выпадающего списка (предопределенных заранее или ранее введённых значений). В строковые атрибуты можно вводить URL-адреса, если значение атрибута начинается с: “http://”, “https://”, “ftp://”, “file://”, “\\”;
- если фон ячейки для ввода значения атрибута подсвечивается белым цветом, то это означает, что редактирование данного атрибута запрещено в соответствии с правами доступа текущего пользователя (Рисунок 183);

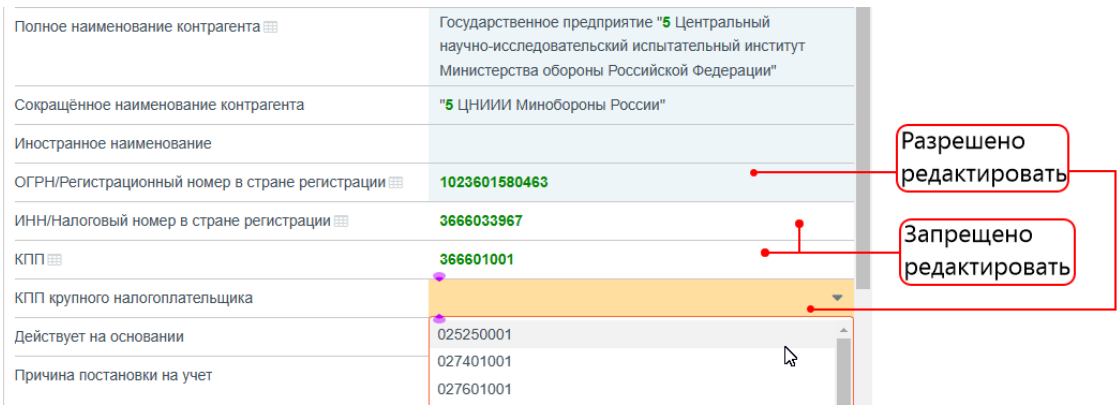


Рисунок 183 - Подсветка ячеек при отсутствии прав доступа на редактирование в режиме управления пакетами изменений

- для атрибутов с типом данных «строка» и признаком «множественный» можно вставить (скопировать) значения с разделителем «|» из буфера обмена. Если значение, вносимое в строковый атрибут превышает 500 символов, то оно будет автоматически обрезано при вводе;
- в значения атрибутов с типом данных «файл» и признаком «множественный» можно добавить сразу несколько файлов, указав их при выборе в *Проводнике*;
- если значение атрибута выбирается из выпадающего списка, но список значений отсутствует, то для него отображается системный элемент <отсутствуют значения для выбора>. Сортировка строковых значений по алфавиту в выпадающем списке значений выполняется при условии раскрытия полного списка значений. В случае выполнения поиска значения в выпадающем списке, сортировка выполняется по релевантности (Рисунок 184).

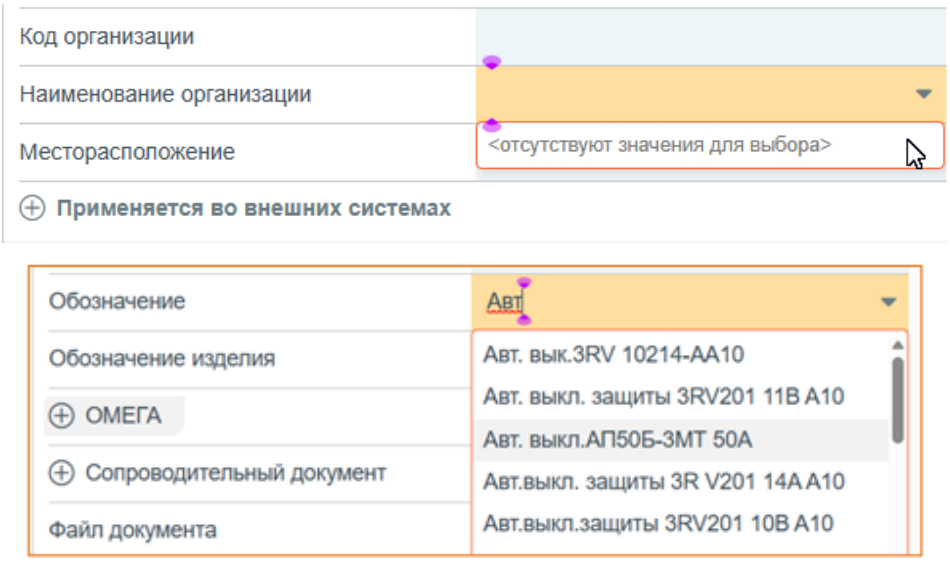


Рисунок 184 - Выбор значения атрибута из списка значений

- опубликовать_пакет изменений для фиксирования новых значений атрибутов в базе.

3.13.4 Редактирование значений атрибутов в таблице

Редактирование значений атрибутов позиций можно выполнять непосредственно в таблице. Для этого все редактируемые значения должны быть доступны на сетке данных (Рисунок 185). Редактировать можно непосредственно в таблице, в том числе, путем копирования / вставки списка значений из текстового файла или таблицы Excel.

Для редактирования данных в таблице следует:

- перейти в нужную справочную группу или отфильтровать поиском все необходимые объекты;
- выполнить настройку видимых столбцов на сетке данных;
- создать новый пакет изменений;
- выполнить двойной щелчок левой кнопкой мыши в нужной ячейке и ввести новое значение вручную, скопировать из текстового или табличного файла, выбрать ссылку на значение из другого справочника (ветки справочника) или из выпадающего списка (предопределенных заранее или ранее введённых значений). В строковые атрибуты можно вводить URL-адреса, если значение атрибута начинается с: «http://», «https://», «ftp://», «file://», «\\».

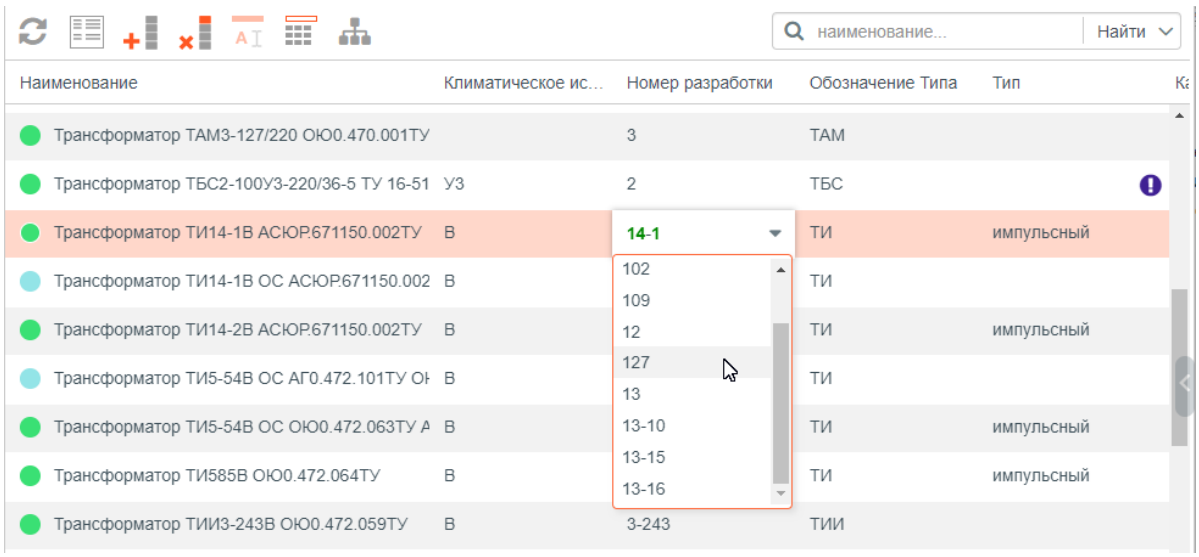


Рисунок 185 - Редактирование значений на сетке данных

3.13.5 Массовая вставка и редактирование значений атрибутов

Вставка и редактирование значений атрибутов объектов осуществляется только в режиме редактирования.

При необходимости задать или изменить значения какого-либо атрибута для группы объектов можно воспользоваться функционалом массового изменения значений атрибутов. Для этого необходимо:

- создать пакет изменений справочных данных;
- вызвать контекстное меню на заголовке столбца с атрибутом, требующим изменения. В случае, если на сетке таблицы не отображается нужный атрибут, следует настроить видимость колонок таблицы;
- выбрать из контекстного меню команду **Задать значение** (Рисунок 186) для задания нового значения выбранного атрибута у всей группы объектов;

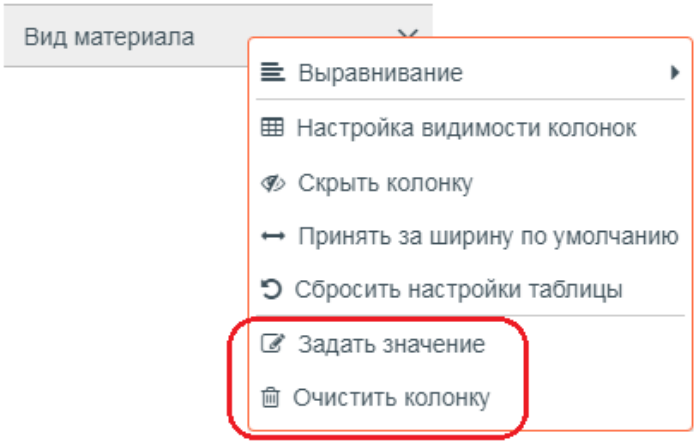


Рисунок 186 - Группа команд для массового редактирования значений атрибутов

- в открывшемся окне **Значение** (Рисунок 187) указать значение атрибута и нажать кнопку ***Ок***. Если у каких-либо объектов группы значения редактируемого атрибута присутствуют, то в поле введения значения по нажатию кнопки будет доступен выпадающий список значений. Для атрибутов типа *Файл* или *Изображение* можно указать расположение соответствующего файла, рисунка или ссылку на них;

- опубликовать пакет изменений для фиксирования изменённых значений атрибутов в базе.



а) б)

Рисунок 187 - Окно **Введите значение атрибута**

При необходимости удалить значения атрибутов у группы объектов воспользуйтесь командой **Очистить колонку** контекстного меню заголовка соответствующего столбца таблицы.

3.13.6 Редактирование позиций, созданных по ограничительным таблицам

Позиции, созданные по ограничительным таблицам, следует редактировать через них же. Позиции, создаваемые и/или проверяемые по ограничительным таблицам, в режиме управления пакетами изменений имеют в карточке позиции рядом с названием атрибута пиктограмму , которая позволяет перейти непосредственно в ограничительную таблицу.

Чтобы отредактировать позицию на сетке данных необходимо:

- найти требуемый объект и установить на нем курсор в таблице;
- создать новый пакет изменений;
- в панели инструментов таблицы позиций нажать кнопку ***Изменить позицию***;

Если позиция была создана с использованием ограничительной таблицы / таблиц, то в режиме управления пакетами изменений будет активна кнопка **Изменить позицию** (Рисунок 188). Во всех остальных случаях эта кнопка не активна.

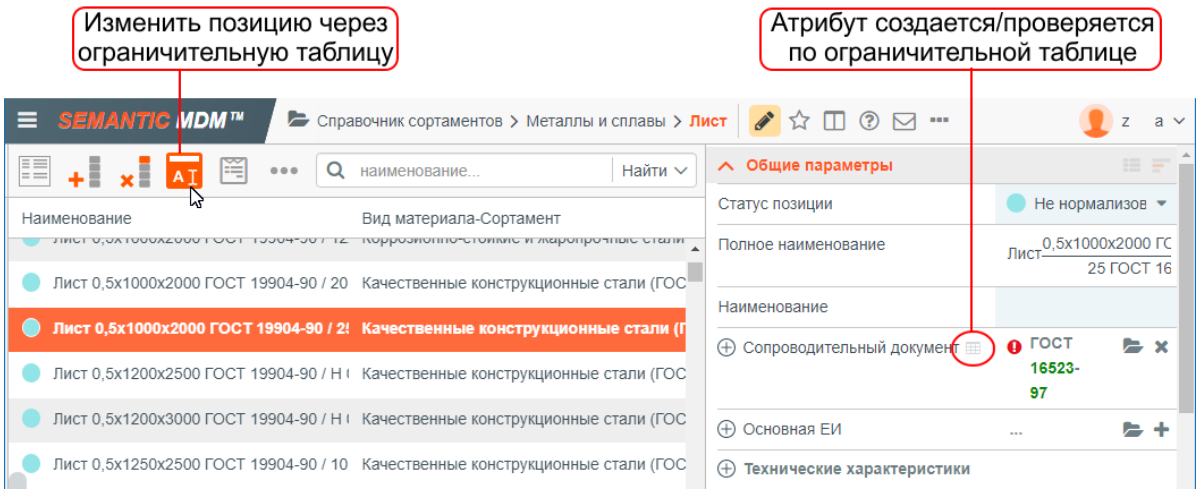


Рисунок 188 - Обращение к ограничительным таблицам из карточки позиции

- в открывшемся окне **Мастер генерации позиций** (Рисунок 189) внести необходимые изменения в значения атрибутов объекта;
- если позиция генерируется с использованием нескольких таблиц, то последовательно внести изменения (при необходимости) в каждую из них;

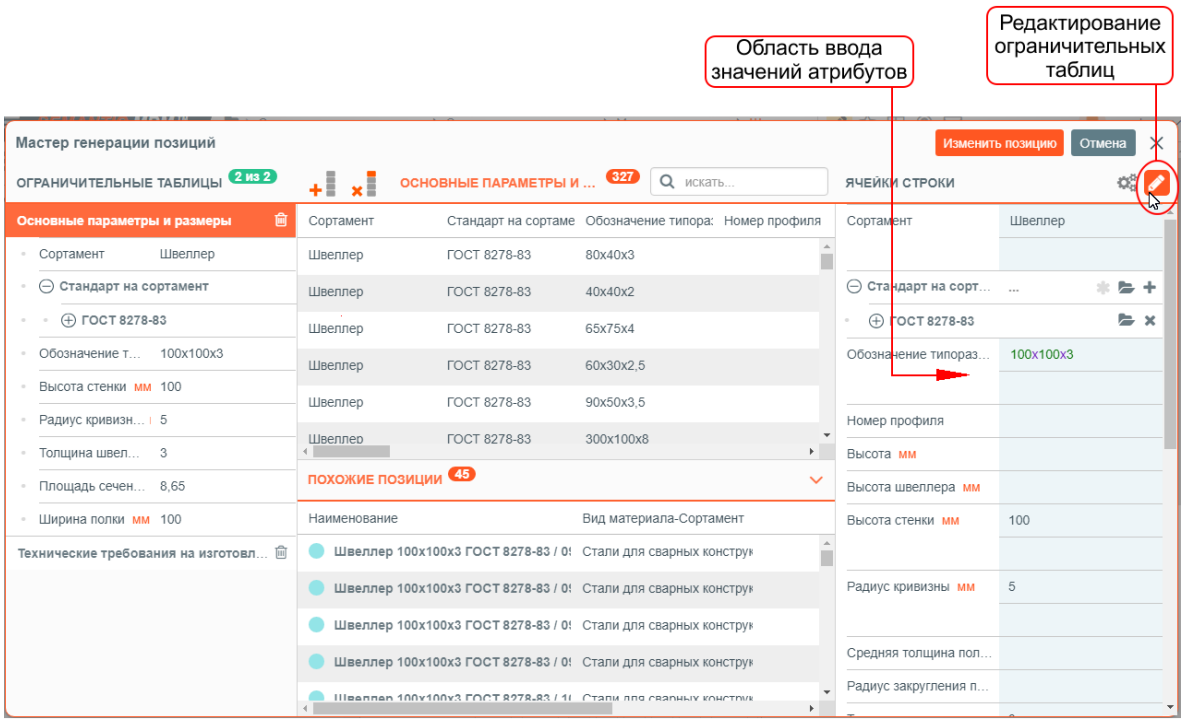


Рисунок 189 - Редактирование записей в ограничительной таблице

- после внесения изменений нажать кнопку ***Изменить позицию***. В результате чего изменения будут отображены в карточке позиции.

При внесении изменений в значения атрибутов позиции без обращения к ограничительным таблицам возможно возникновение ошибок, в результате которых запись не сможет получить статус Нормализована. Несоответствие позиции ограничительным таблицам обозначается пиктограммой  в строке на сетке данных.

При редактировании значений в ограничительных таблицах, внесённые данные автоматически проверяются на соответствие другим возможным ограничениям – связанным ограничительным таблицам, уникальным и обязательным атрибутам, а также функциям проверки данных. Значения атрибутов, не соответствующие любым типам ограничений, не позволят опубликовать пакет изменений в случае присвоения позициям статуса **Нормализована** (см. п. 3.14.20).

3.13.7 Редактор связей

Редактор связей открывается из карточки позиции и доступен только для множественных связей агрегации/композиции.

Редактор связей открывается только для одной выбранной множественной связи. Перемещение по дереву связей возможно только сверху вниз.

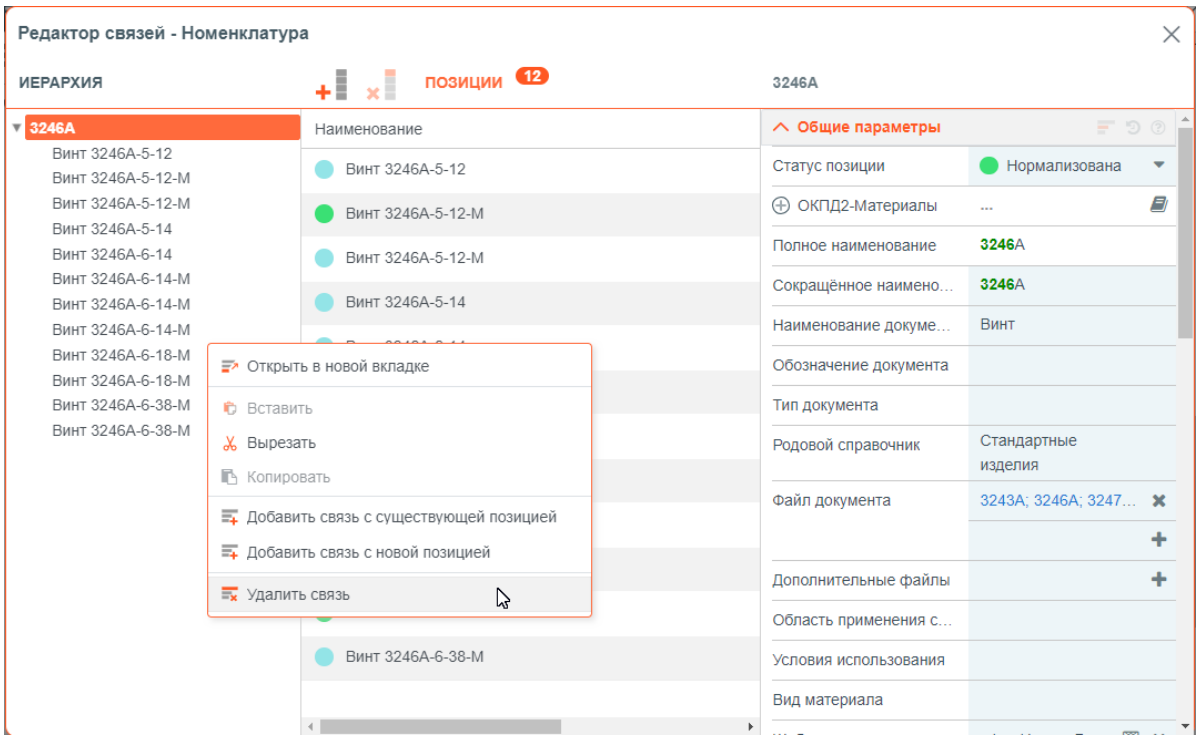


Рисунок 190 - Редактирование связей для заполнения значений атрибута

В режиме редактирования в редакторе можно добавить новую связь с существующей позицией или с новой позицией. При добавлении новой связи в присоединяемой позиции будет автоматически заполнено значение атрибута, который определяет эту связь. Например, при добавлении в нормативный документ новой позиции номенклатуры, в этой позиции будет автоматически заполнено значение атрибута *Сопроводительный документ*.

3.13.8 Декларативная совместимость позиций

Настройка декларативной совместимости выполняется в карточке позиции в разделе **Совместимости** в режиме редактирования. Совместимости предназначены для создания и настройки семантических связей между различными объектами НСИ. Очевидным примером семантической связи является, например, указание аналогов для материалов или режущего инструмента. Кроме этого, совместимости могут устанавливаться и для более сложных видов связей. Создание нового вида совместимости выполняется в представлении **Конфигурация данных** или может быть выполнено с использованием диаграммы связей совместимости, которая вызывается

соответствующей кнопкой в разделе **Совместимости**.

Для создания семантической связи в позиции следует:

- Перейти в режим редактирования;
- Установить курсор на нужной позиции и перейти в карточку;
- Развернуть раздел **Совместимости** (Рисунок 191) и выбрать из выпадающего списка нужный вариант семантической связи. Количество настраиваемых совместимостей не ограничено!;
- В строке с добавленным вариантом семантической связи нажать кнопку ***Добавить***;

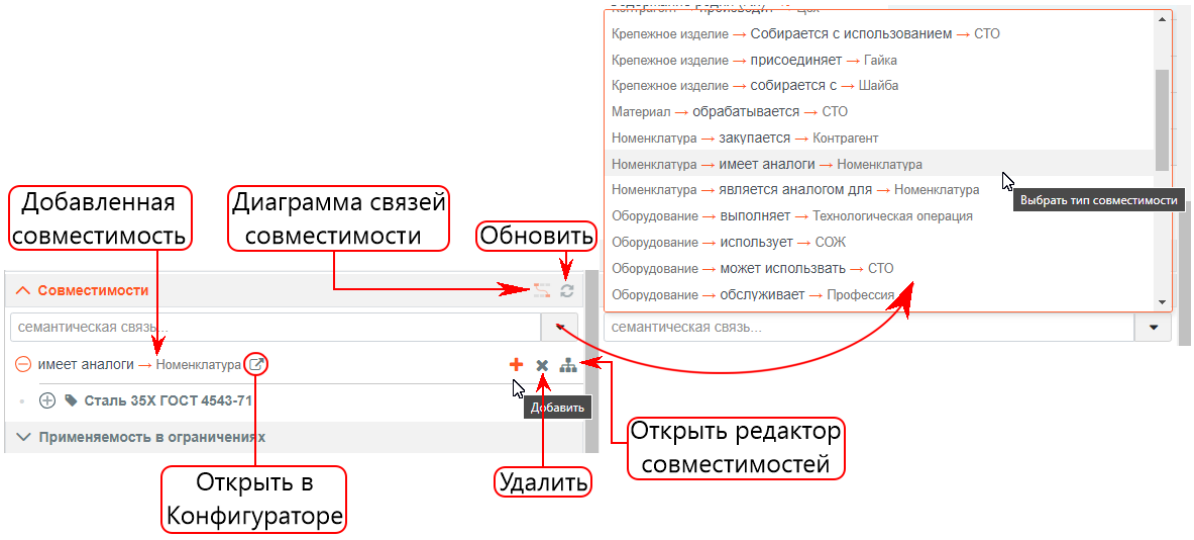


Рисунок 191 - Добавление новой совместимости в карточке позиции

- В открывшемся дополнительном окне со справочниками найти нужную позицию (можно использовать любые виды поиска) или несколько позиций, установить курсор на требуемом объекте и нажать кнопку ***Выбрать*** (внизу справа). Для выбора нескольких позиций нужно держать нажатой кнопку <Ctrl> при выборе объектов.

При создании совместимости между позициями после нажатия на кнопку ***Ок*** в окне **Подтверждение**, в карточке появится связанная позиция или позиции, каждая из которых может быть удалена из списка кнопкой ***Удалить*** в строке с названием позиции.

Для работы с совместимостями также можно использовать **Редактор совместимостей** (Рисунок 192), окно которого вызывается кнопкой ***Открыть редактор совместимостей***. Редактор позволяет:

- отображать большое количество собственных и унаследованных связей совместимостей текущей позиции / группы с другими позициями и группами;
- выполнять поиск среди совместимых позиций;
- выполнять фильтрацию по собственным / унаследованным от группы связям.

Кнопки для добавления и удаления совместимостей располагаются над списком совместимых объектов. Рядом располагается указание на количество собственных и унаследованных совместимостей, которые могут использоваться для быстрой фильтрации позиций в таблице - для этого необходимо щелкнуть левой кнопкой мыши на нужном признаке.

Кнопка  Удалить все собственные обеспечивает удаление собственных совместимостей, при этом совместимости, унаследованные от справочной группы (см. п. 3.13.9) затронуты не будут.

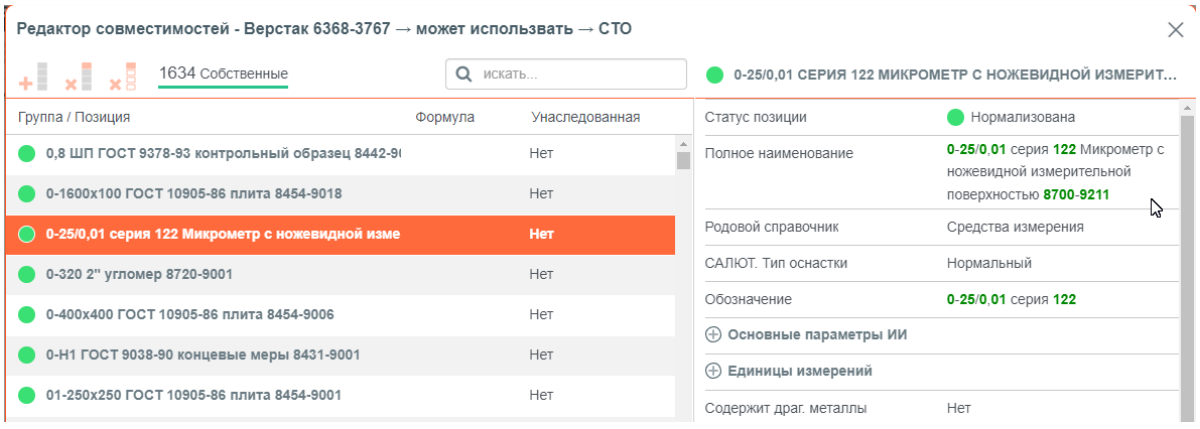


Рисунок 192 - Окно **Редактор совместимостей**

Кнопка **Диаграмма связей совместимости** вызывает окно **Связи совместимости**, где можно изменить существующие и настроить новые

семантические связи. Описание последовательности настройки связей приведено в п. 3.22.

В *Модуле импорта и репликации* поддерживается импорт и экспорт совместимостей, с помощью которого можно организовать:

- импорт совместимостей оборудования, оснастки и технологических операций между собой из PDM-системы;
- экспорт аналогов и замен в систему организации коммерческих закупок, в которой важно понимать какие номенклатурные позиции на какие могут быть заменены.

3.13.9 Вычисляемая совместимость справочных групп

Настройка совместимости выполняется в карточке группы в разделе **Совместимости** в режиме редактирования. Совместимости предназначены для создания и настройки семантических связей между различными справочными группами и позициями этих групп, которые наследуют настроенные совместимости. Образцом семантической связи является, например, указание технологического оснащения, принадлежащего структурной единице производства - цеху или участку. Кроме этого, совместимости могут устанавливаться и для более сложных видов связей. Создание нового вида совместимости выполняется в представлении **Конфигурация данных** или может быть выполнено с использованием диаграммы связей совместимостей (см. п. 3.22), которая вызывается соответствующей кнопкой в разделе **Совместимости**.

В *Semantic MDM* для справочных групп выполняется настройка вычисляемых совместимостей. Вычисляемые совместимости могут возникать между любыми двумя группами справочников.

Для создания семантической вычисляемой связи (Рисунок 193) необходимо:

- Перейти в режим редактирования;
- Установить курсор на нужной группе и перейти в карточку;

- Развернуть раздел **Совместимости** и выбрать из выпадающего списка нужный вариант семантической связи. Количество настраиваемых совместимостей не ограничено!;
- В строке с добавленным вариантом семантической связи нажать кнопку ***Добавить***;

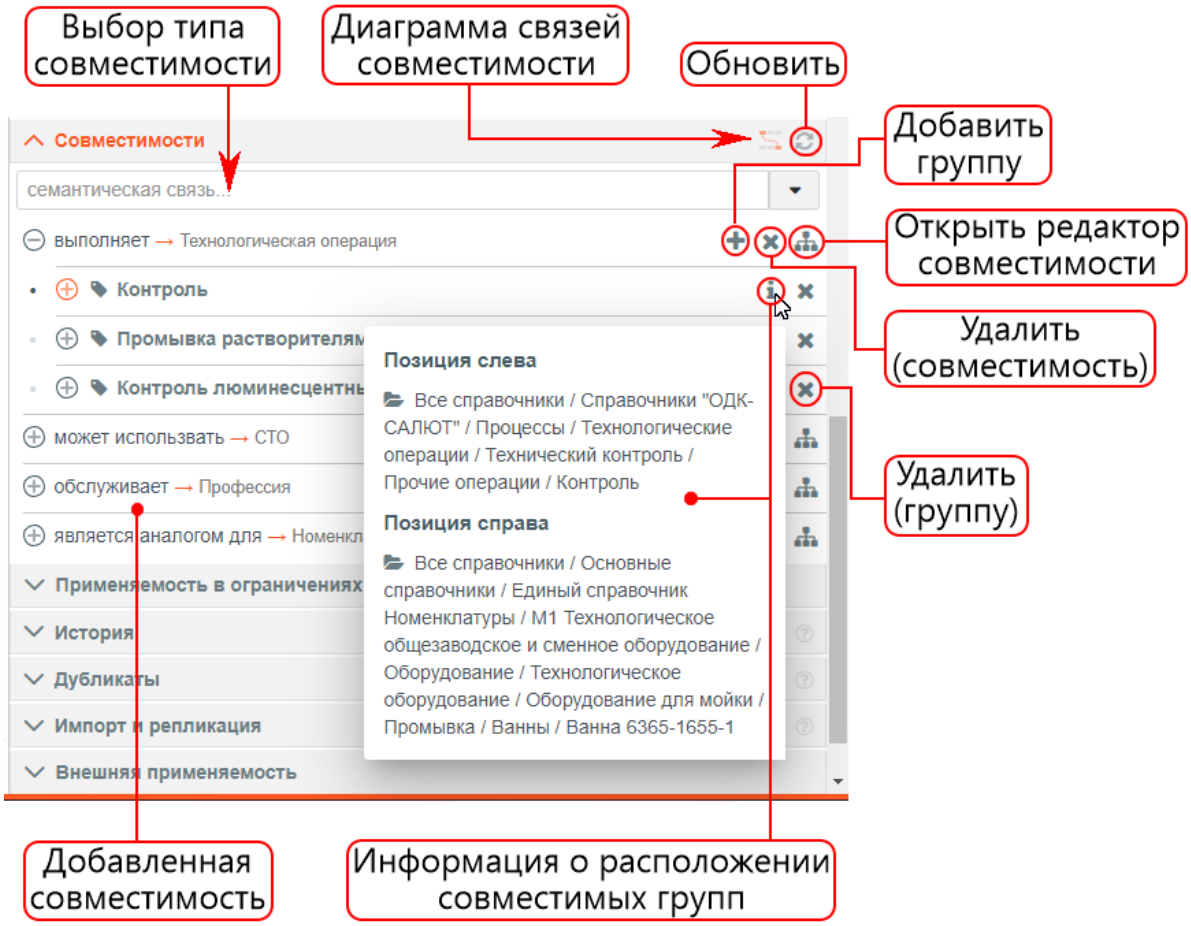


Рисунок 193 - Добавление новой совместимости в карточке объекта (группы или позиции)

- В открывшемся дополнительном окне со справочниками найти интересующую группу (можно использовать любые виды поиска), установить курсор на требуемом объекте и нажать кнопку ***Выбрать*** (внизу справа). Для выбора нескольких групп нужно держать нажатой кнопку <Ctrl> при выборе объектов.

При создании совместимости между группами после нажатия на кнопку ***Ок*** в окне **Подтверждение**, она сразу определяется как вычисляемая и пользователю предлагается выполнить настройки сравнения атрибутов,

которые позволяют определить совместимость конкретных позиций совместимых справочных групп. Эта настройка производится в окне **Редактор вычисляемой совместимости** (Рисунок 194) на вкладке **Настройка**. Без этой настройки все выбранные объекты считаются несовместимыми.

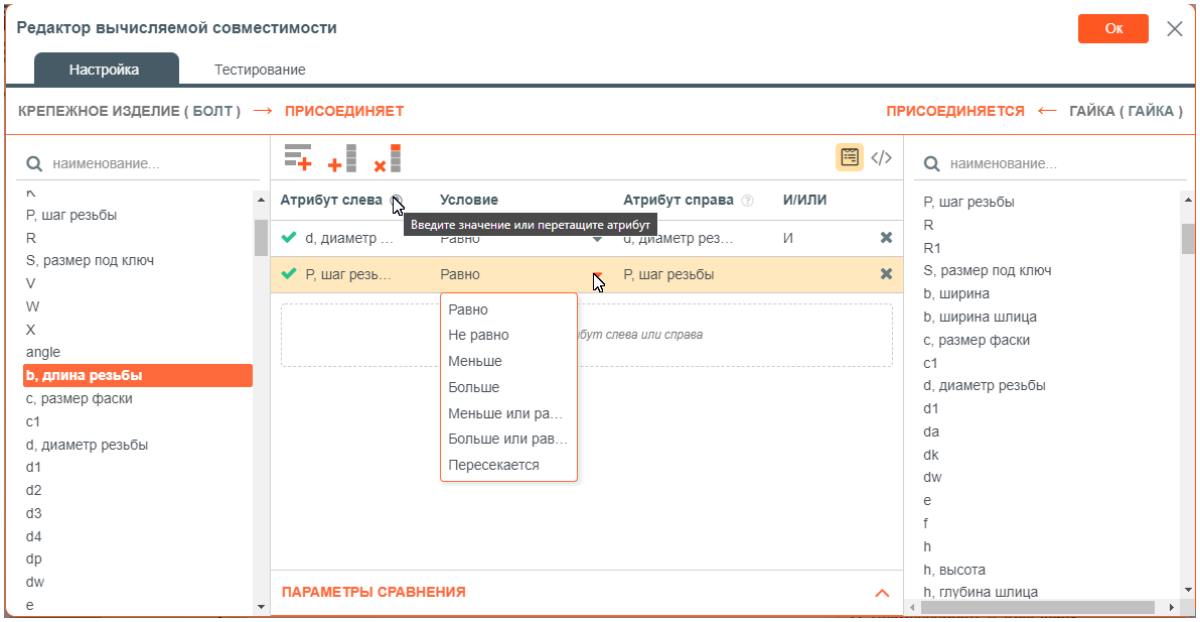


Рисунок 194 - Настройка совместимости объектов через условия сравнения атрибутов позиций

Добавление сравниваемых атрибутов производится путём их перетаскивания (Drag&Drop) из списка атрибутов в ячейку *Перетащите атрибут слева или справа*, расположенную в центре окна редактора. Сравниваемый атрибут добавляется в появившуюся строку в столбец **Атрибут слева** или **Атрибут справа**. Над таблицей с атрибутами располагаются кнопки, позволяющие добавить  или удалить  условие совместимости, а также добавить группу условий  . Добавление условия по кнопке и перетаскивание атрибутов являются взаимозаменяемыми.

Пользователь может настраивать различные условия сравнения атрибутов позиций справа и слева. Кроме стандартных условий сравнения, в зависимости от типа самого атрибута, доступны следующие:

- Сравнение строк через маппинг значений – например, если ищется труба, через которую можно подать «пар», то в качестве замены

подойдут трубы, через которые можно прогнать: «вода», «масло», «нефть»;

- Отклонение в большую или меньшую сторону на заданное значение – например, если определяются аналоги для прутка диаметром «6 мм», то подойдут прутки такого же или чуть большего диаметра «6 – 10 мм»;
- Настройка режима сравнения пустых значений с непустыми – например, если для прутка слева не указана точность проката, то в качестве аналога подойдут прутки как с обычной точностью «Б», так и с повышенной точностью «В». Если же точность указана, то подойдут прутки только указанной точности;
- Настройка режима сравнения пустых значений – например, можно настроить сравнение пустых значений и определить, следует ли считать два атрибута с пустыми значениями равными или нет;
- Пересечения по числовым диапазонам – например, для постоянного резистора подобрать аналоги в переменных резисторах;
- Для атрибутов с массивом значений, определить условия вхождение одного массива в другой или пересечения массивов значений – например, контрагентов можно считать аффилированными (совместимыми), если их владельцы пересекаются.
- Условие **И/ИЛИ** доступно, если для сравнения выбрано более одного атрибута:
- Условие **И** означает, что позиции считаются совместимыми при соблюдении обоих (нескольких) условий сравнения;
- Условие **ИЛИ** означает, что позиции считаются совместимыми при соблюдении хотя бы одного из условий сравнения.

По-умолчанию для атрибутов сравнения всегда устанавливается условие И.

На панели **Параметры сравнения** (Рисунок 195) (расположена снизу списка атрибутов) можно задать дополнительные параметры сравнения для

атрибутов, не имеющих заполненных значений - пустых. На панели доступны два режима сравнения с выпадающими значениями:

- Режим сравнения пустых значений:
  - <пусто>=< пусто> - если слева и справа пустые значения атрибутов, то считать позиции совместимыми по этому критерию;
  - <пусто><>< пусто> - если слева и справа пустые значения атрибутов, то считать позиции несовместимыми по этому критерию;
- Режим сравнения пустого значения в непустым:
  - Стандартное сравнение - соответствует режиму, установленному в режиме сравнения пустых значений, а непустые в соответствии с настройками критериев;
  - <пусто>=< любое значение> - если слева пустое значение, а справа любое значение атрибута, то считать позиции совместимыми по этому критерию;
  - <любое значение>=< пусто> - если слева любое значение, а справа пустое значение атрибута, то считать позиции совместимыми по этому критерию.

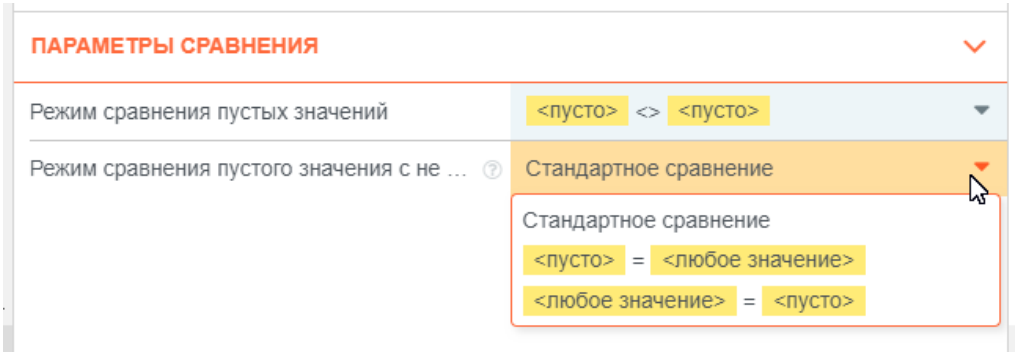


Рисунок 195 - Настройка сравнения атрибутов с пустыми значениями

Все сравниваемые атрибуты делятся на простые критерии сравнения, которые не входят ни в одну группу условий и критерии, входящие в группу условий. Простые критерии используются для выполнения предварительного отбора совместимых позиций. Простыми критериями сравнения являются критерии, которые удовлетворяют следующим условиям:

- не входят ни в одну группу условий;
- используются простые условия сравнения: =, >, <, >=, <=, пересекаются, за исключением <>;
- если выбрано условие сравнения =, то для него определен режим сравнения: <пусто> <> <пусто>;
- если выбрано условие сравнения =, то для него используется стандартный режим сравнения пустого значения с непустым значением;
- если выбрано условие сравнения =, то для него отсутствует маппинг значений.

Простые критерии сравнения отмечаются специальной пиктограммой , расположенной в строке сравнения слева.

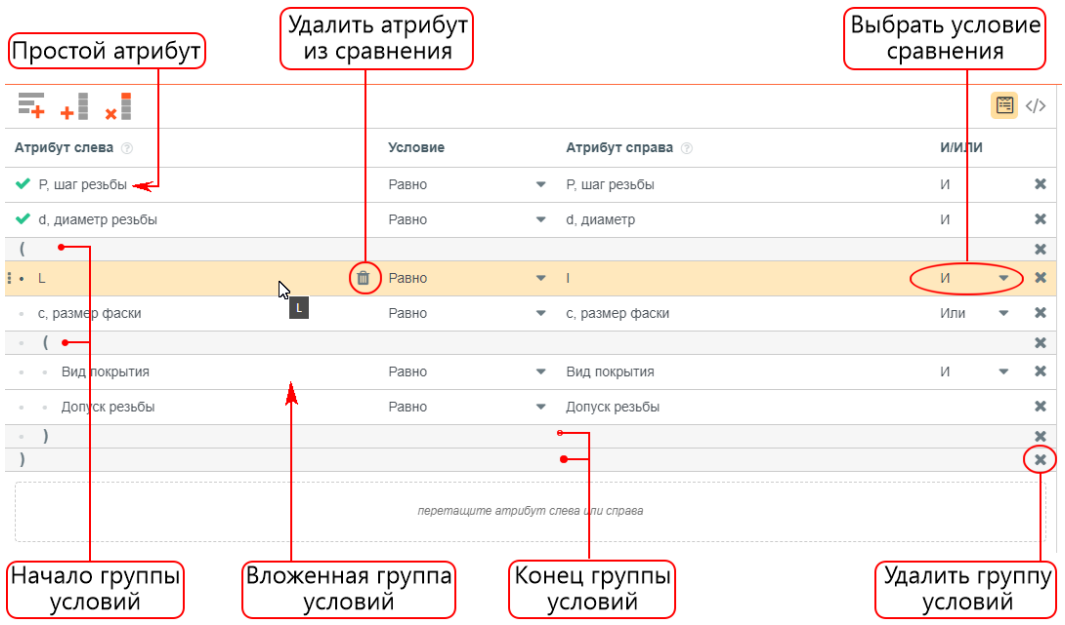


Рисунок 196 - Использование групп условий в вычисляемой совместимости

Для добавления группы условий (Рисунок 196) нужно нажать кнопку ***Добавить группу условий***. Группа условий выглядит как две строки светло-серого цвета, на верхней располагается открывающая скобка **(**, а на нижней - закрывающая **)**. Атрибуты из групп слева и справа могут быть

добавлены в группу при помощи перетаскивания (Drag&Drop), в том числе из ранее созданных простых критериев.

Группы условий могут быть вложенными, при этом следует учитывать, что группа условий - это список критериев сравнения совместимых между собой через И. При использовании вложенной группы условий для них существует выбор И или ИЛИ для совмещения вложенных групп с вложенными условиями.

Логика составления условий в целом выглядит следующим образом: “общие условия” И ((группа условий 1) ИЛИ (группа условий 2) ИЛИ (группа условий N)), где общие условия - это список критериев сравнения состыкованных между собой через И, который должен содержать хотя бы один простой критерий сравнения, позволяющий не выполнять полный перебор.

Если в настройках вычисляемой совместимости отсутствуют простые критерии в общих условиях, то система не позволит создать такую вычисляемую совместимость и при попытке сохранить ее, выдать сообщение об ошибке (Рисунок 197).

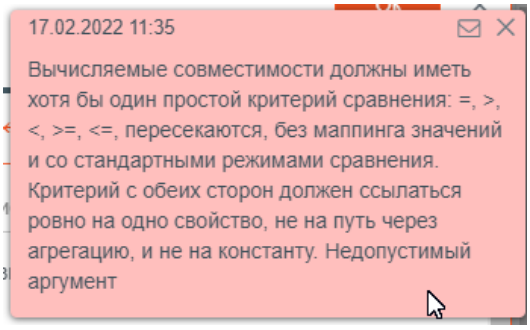


Рисунок 197 - Сообщение об ошибке в настройках вычисляемой совместимости

При удалении группы условий критерии сравнения не удаляются а перемещаются в список простых критериев сравнения.

Для того, чтобы открыть **Редактор вычисляемой совместимости** (Рисунок 198) для редактирования ранее настроенных совместимостей нужно нажать на пиктограмму , расположенную в строке с наименованием

требуемой справочной группы или аналогичную пиктограмму в окне **Редактора совместимостей**.

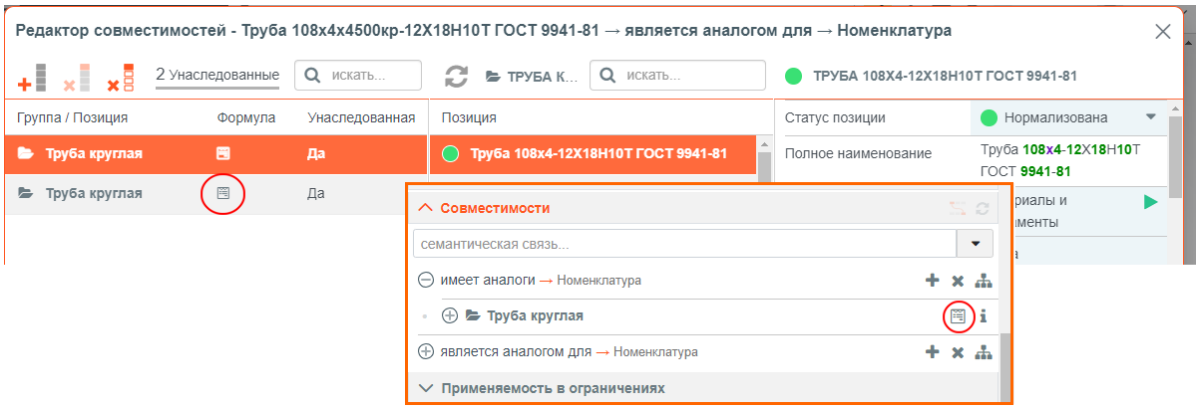


Рисунок 198 - Вызов **Редактора вычисляемой совместимости**

В **Редакторе вычисляемой совместимости** на вкладке **Тестирование** (Рисунок 199) можно выполнять тестирование набора условий вычисления совместимостей, в результате *Система* запоминает выбранные комбинации пар позиций, на которых проводилось тестирование.

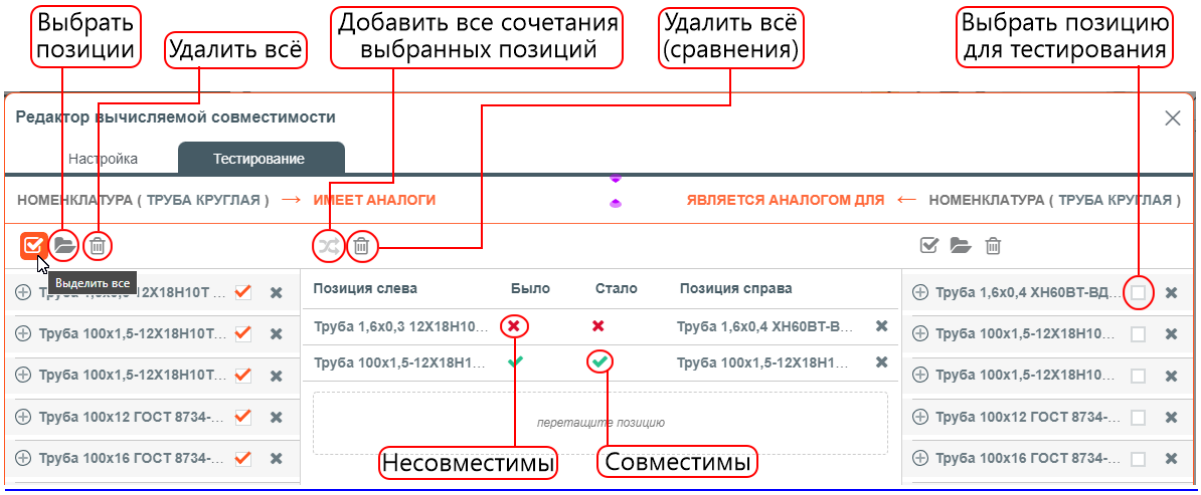


Рисунок 199 - Тестирование настроенных условий совместимости и создание связей

Для тестирования созданного набора условий на вкладке **Тестирование**, используя кнопки ***Выбрать позиции***, выбрать из справочных групп нужные позиции. Маркерами отметить отдельные позиции для тестирования либо выбрать все позиции из списка, нажав на кнопку ***Выделить всё*** и нажать

кнопку ***Добавить все сочетаний выбранных позиций***. Если позиции проходят по настроенным условиям совместимости, то в столбце **Стало** появится зеленая пиктограмма ✔, а сами позиции будут записаны в раздел **Совместимости** в карточки соответствующих позиций. Если условия сравнения не выполняются, то в столбце появится красная пиктограмма ✖, а совместимость установлена не будет.

Кнопка ***Диаграмма связей совместимости*** вызывает окно **Связи совместимости**, где можно изменить существующие и настроить новые семантические связи.

В *Модуле импорта и репликации* поддерживается импорт и экспорт совместимостей, с помощью которого можно организовать:

- импорт совместимостей оборудования, оснастки и технологических операций между собой из PDM-системы;
- экспорт вычисленных аналогов и замен в систему организации коммерческих закупок, в которой важно понимать какие номенклатурные позиции на какие могут быть заменены.

3.13.10 Управление историческими данными

Исторические данные, их создание, хранение и использование предназначено для ведения позиций и атрибутов, состояние и значение которых зависит от времени. Эта зависимость может носить разный характер - это может быть период действия определенного значения атрибута или позиции в целом, а может быть изменение значения атрибута (характеристики позиции) во времени в связи с внешними причинами, например, изменениями во внешней информационной системе.

Ведение периода действия позиции позволяет:

- выполнить поиск актуальных позиций на заданную дату;
- отобразить иерархический классификатор, актуальный на указанную дату. Получение актуальной структуры классификатора на заданную дату выполняется через расширенный поиск, при просмотре

структуры классификатора;

- отобразить список композируемых позиций, актуальных на заданную дату.

Ведение историчности значения атрибута позволяет:

- просматривать карточку позиции со значениями актуальными на заданную дату;
- выполнять быстрый поиск с учётом исторических значений;
- выполнять расширенный поиск с учётом исторических значений;
- выполнять поиск дубликатов с учётом исторических значений.

Для управления историческими данными пользователь должен иметь привилегию *Управление историчностью данных*, а в *Системе* должна быть активирована функция **Исторические данные** в подсистеме **Настройки**. Для просмотра исторических данных никакая привилегия не требуется (Рисунок 200).

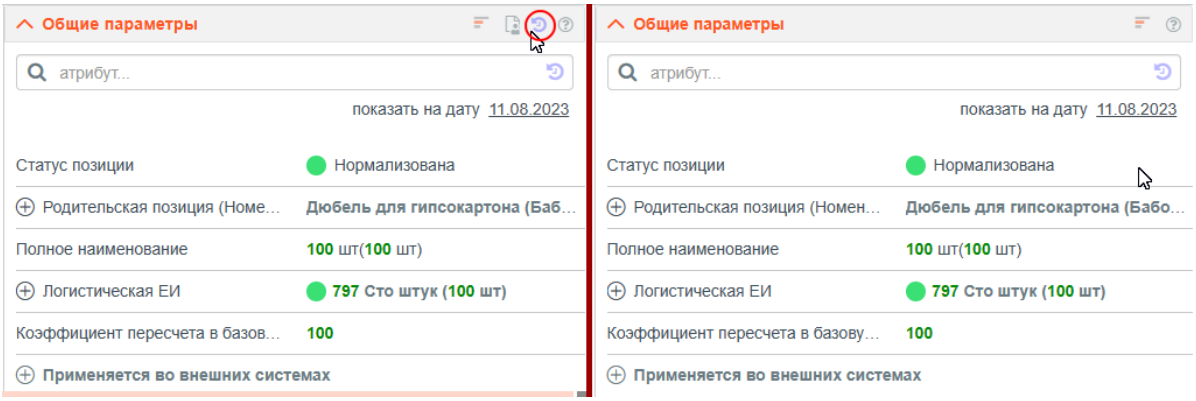


Рисунок 200 - Управление историчностью: справа пользователь с привилегией *Управление историчностью данных*, слева - без неё

Для настройки периода действия выбранной позиции (Рисунок 201) необходимо выполнить следующие действия:

- включить режим управления пакетами изменений;
- нажать на панели **Общие параметры** карточки позиции кнопку ***Настройка историчности*** ;

- в системных атрибутах *Дата начала действия* и *Дата окончания действия* указать даты, используя ввод с клавиатуры или выбор из календаря;
- опубликовать пакет изменений.

Для позиции, срок действия которой закончился, статус будет отображаться в виде специального значка, в том числе на сетке данных, цвет которого соответствует статусу обычной позиции (Рисунок 202).

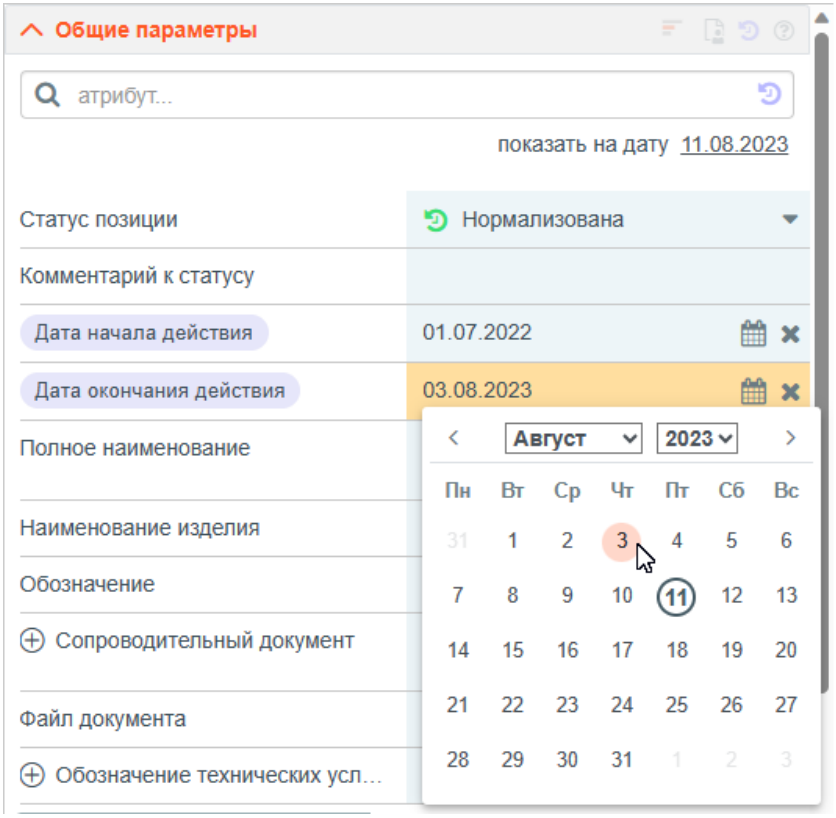


Рисунок 201 - Настройка периода действия позиции

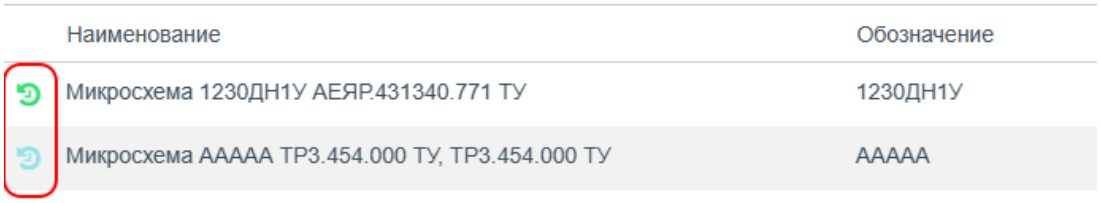


Рисунок 202 - Вид пиктограммы статуса у позиции с закончившимся сроком действия

Для атрибутов с типом данных: *Строка, Целое число, Число с плавающей точкой, Логический тип, Многострочный текст, Файл*, а также в атрибутах связи можно определять исторические значения, действительные до

заданного числа. Исторические значения могут быть заданы как для единичного, так и для множественного атрибута. Для множественного атрибута, в качестве исторического значения должен быть указан полный массив значений, который был действителен до заданной даты.

Ввод исторических значений выполняется в карточке позиции через специальное окно , вызываемое всплывающей кнопкой - пиктограммой, справа от значения атрибута.

Для ввода исторического значения атрибута (Рисунок 203) необходимо выполнить следующие действия:

- включить режим управления пакетами изменений;
- нажать на панели **Общие параметры** карточки позиции кнопку ***Настройка историчности*** ;
- в интересующем атрибуте нажать на всплывающую пиктограмму исторических значений ;
- в открывшемся окне **История значений <наименование атрибута>** нажать кнопку Добавить;
- в ячейке **Действительно до** указать дату окончания действия вводимого или выбираемого (для атрибутов связи) значения. Дата может быть введена с клавиатуры или путем выбора из выпадающего календаря;
- в ячейке **Значение** ввести с клавиатуры или выбрать из выпадающего списка (для атрибутов связи, при вводе предлагаемые значения фильтруются по уже введенным символам) необходимое значение;
- если до указанной даты атрибут не имел значений следует нажать кнопку <пусто>, чтобы установить факт отсутствия значений;
- при вводе исторических данных для атрибута связи справа доступна карточка агрегируемой / композируемой позиции, для атрибутов которой также можно вызвать новое окно **История значений**;
- после ввода необходимых дат и значений закрыть окно и опубликовать пакет изменений.

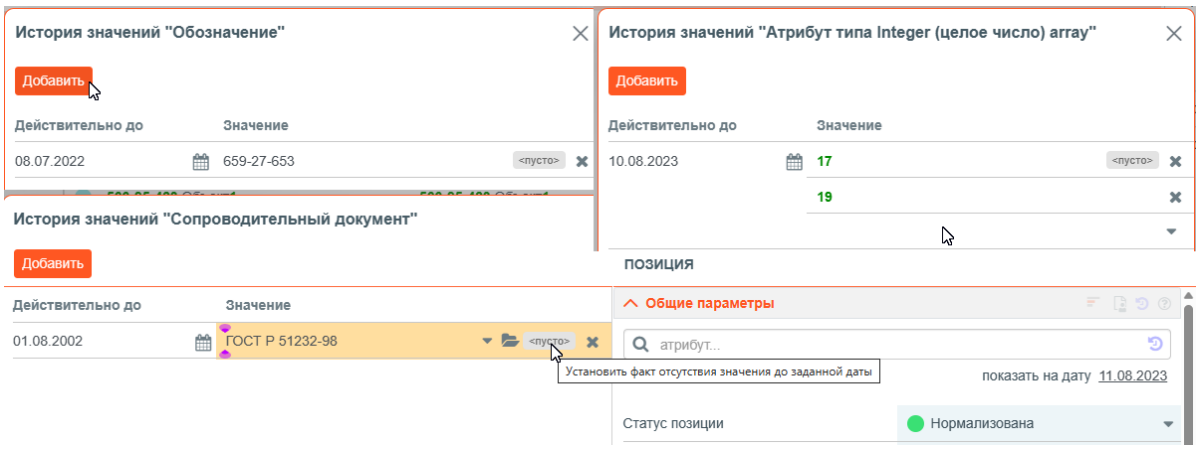


Рисунок 203 - Ввод исторических значений для простого атрибута (вверху слева), для множественного простого атрибута (вверху справа) и для атрибута связи (внизу)

Исторические значения атрибутов отображаются во всплывающем окне в карточке позиции при наведении курсора на пиктограмму исторических значений в соответствующем поле.

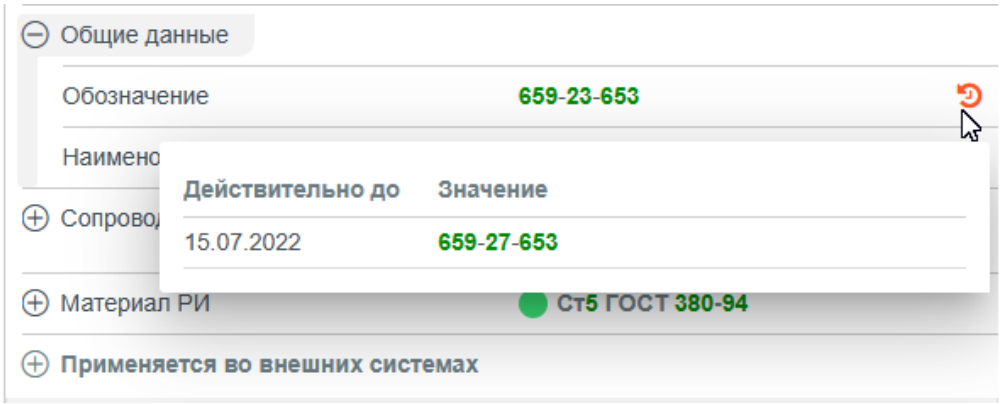


Рисунок 204 - Просмотр исторического значения атрибута

Добавление исторических значений можно выполнить также при просмотре детализации активного пакета изменений, путём нажатия на кнопку ***Записать старое значение как историческое*** справа от наименования атрибута, для которого было изменено значение (

Рисунок 205).

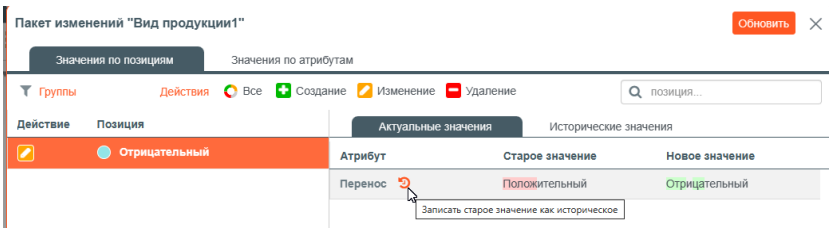


Рисунок 205 - Установление исторического значения атрибута через детализацию активного пакета изменений

Исторические значения могут выгружаться и загружаться из файлов форматов Excel, CSV и XML. Для этого предусмотрены соответствующие настройки в диалогах выгрузки (Рисунок 206).

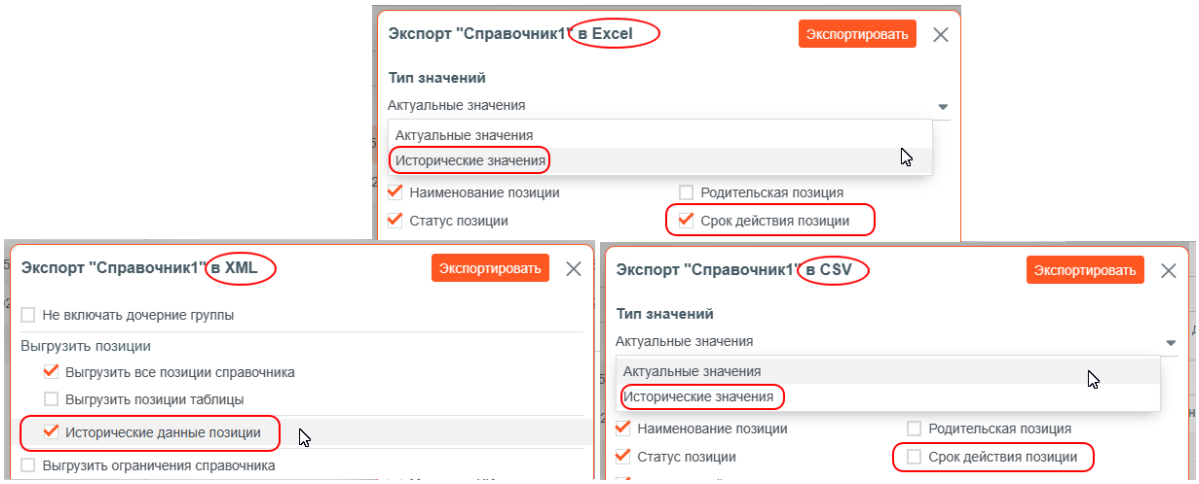


Рисунок 206 - Настройка выгрузки исторических данных в диалогах формирования файлов Excel, CSV и XML

3.13.11 История изменений справочных данных

Все изменения в *Системе* сохраняются и могут быть просмотрены через сравнение версий текущей позиции и / или детализацию пакета изменений. Сведения об изменениях значений любого атрибута объекта доступны в разделе **История** (Рисунок 207) в карточке позиции. Раздел доступен только для пользователей с привилегией *Просмотр истории*. В истории дополнительной пиктограммой обозначаются изменения, выполненные по заявке. При наведении курсора на пиктограмму, во всплывающем окне отображаются сведения о заявке. По щелчку на пиктограмме заявки

открывается вкладка с подсистемой **Журнал заявок**, а курсор позиционируется на интересующей заявке.

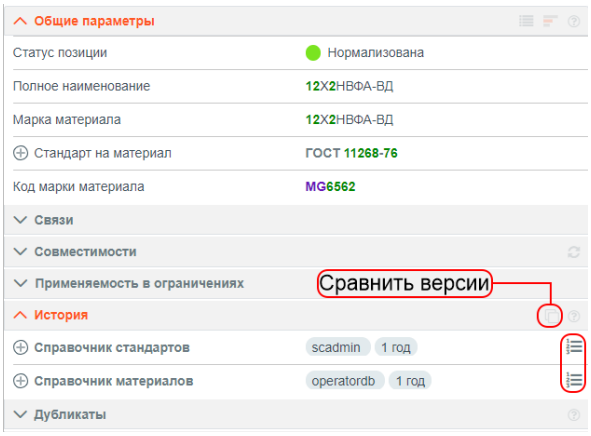


Рисунок 207 - Раздел История с пакетами изменений и сведениями о заявке

Для того, чтобы посмотреть изменения по текущему объекту следует:

- установить курсор на объекте в таблице данных;
- перейти в раздел **История** в карточке позиции и:
  - нажать кнопку ***Сравнить версии*** и, в открывшемся окне **Сравнение версий позиции** (Рисунок 208), ознакомится с предыдущими версиями объекта. Для сравнения могут быть выбраны только атрибуты с различающимися значениями и / или только основные атрибуты. Над колонкой с атрибутами располагается строка, в которой указывается статус позиции, срок, когда производилось изменение и пользователь, создавший пакет изменений. При необходимости любая из версий может быть скрыта кнопкой Скрыть версию (при повторном вызове сравнения скрытые версии будут вновь показаны), а также может быть вызвана детализация пакета изменений по выбранной версии;

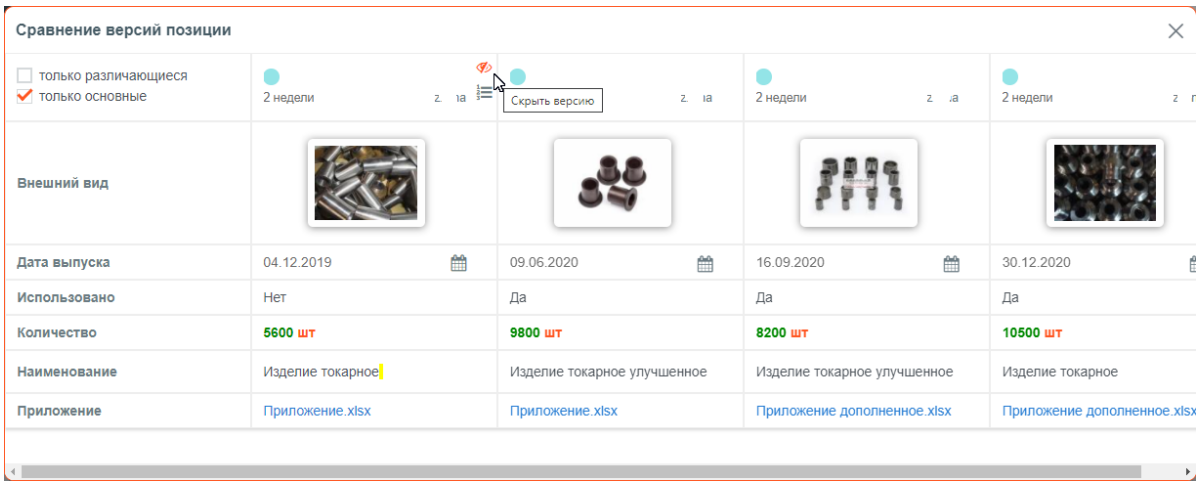


Рисунок 208 - Сравнение версий справочной позиции

- выбрать интересующий пакет изменений (каждый пакет содержит имя пользователя, проводившего изменения и срок проведения изменений, при наведении курсора на который становится доступна дата и время опубликования пакета);
- нажать кнопку ***Детализация*** и, в открывшемся окне **Пакет изменений**, ознакомиться с интересующей информацией.

Кроме этого, доступ к истории изменения справочных данных возможен со стороны управления пакетами изменений через команду **Управление пакетами изменений** (Рисунок 209).

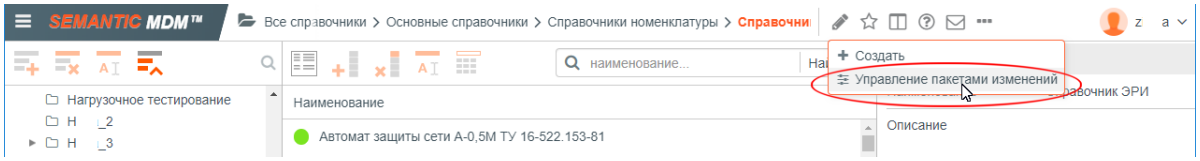


Рисунок 209 - Доступ к команде **Пакеты изменений**

3.13.12 Детализация пакета изменений

При обращении к команде **Управление пакетами изменений** открывается дополнительное окно **Пакеты изменений**, где в таблице представлены все пакеты изменений выполненные ранее, в том числе, приостановленные и активные. В таблице можно выполнить сортировку по

столбцам, быстрый поиск по значениям столбцов, расширенный поиск по датам, а также отфильтровать пакеты по статусу. Фасетный поиск разрешен для всех столбцов, кроме *Идентификатор*, *Заявка* и *Возможность активации*.

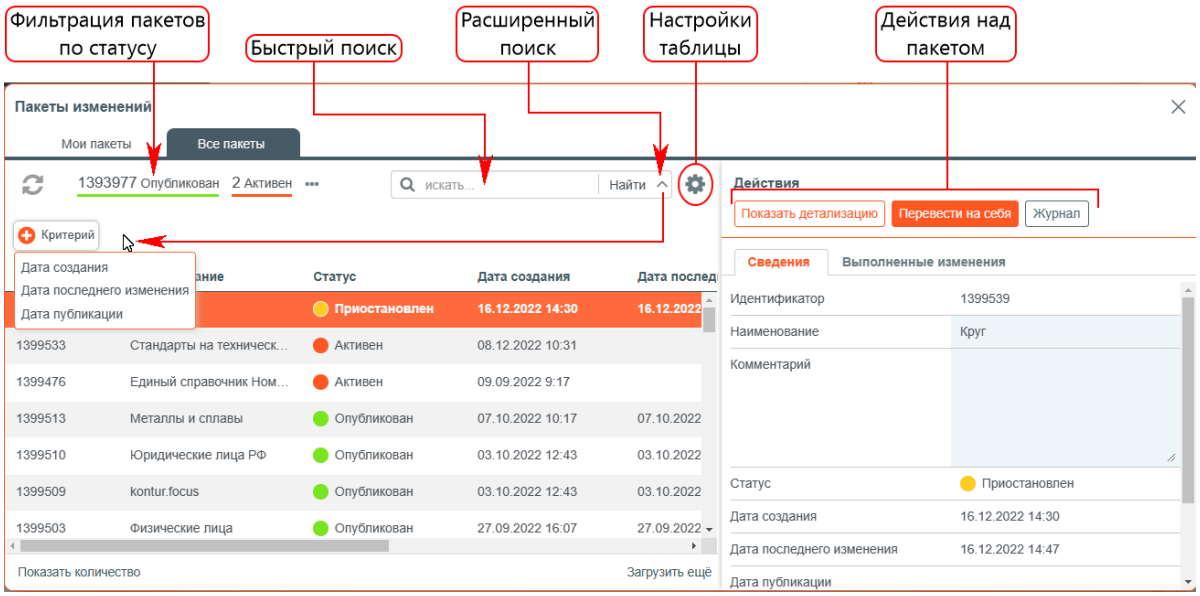


Рисунок 210 - Окно **Пакеты изменений**

Количество видимых столбцов в таблице пакетов изменений регулируется настройкой. Для таблицы с пакетами изменений доступны следующие колонки (Рисунок 211).

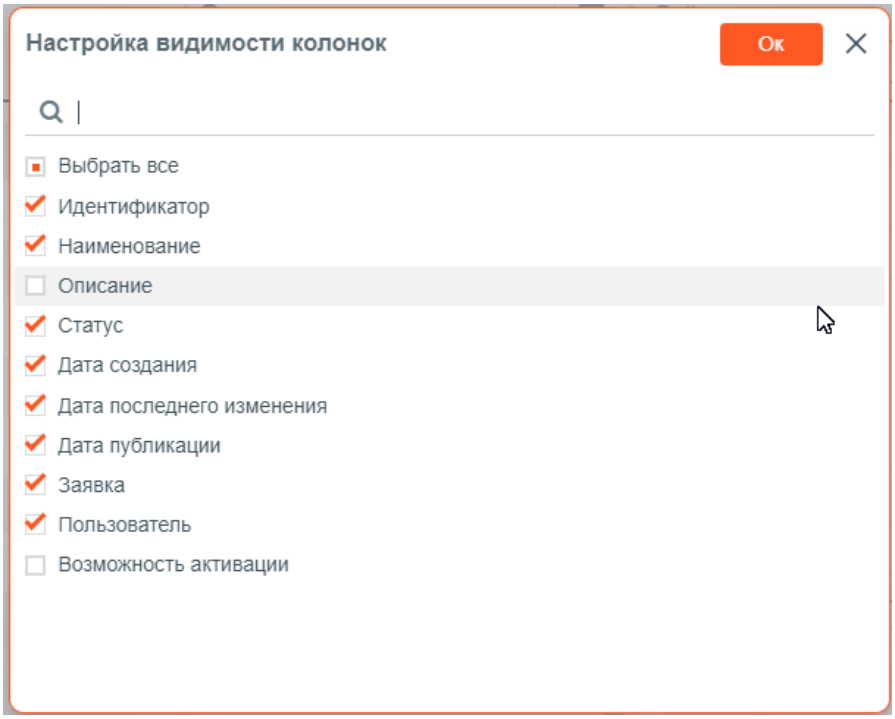


Рисунок 211 - Настройка видимости колонок в таблице истории пакетов изменений

Расширенный поиск в окне **Пакеты изменений** включает в себя три возможных критерия - *Дата создания, Дата последнего изменения, Дата публикации* (Рисунок 212). Указание даты может производиться с помощью перемещения специального маркера на временной шкале, вручную с клавиатуры, путем выбора значений из календаря.

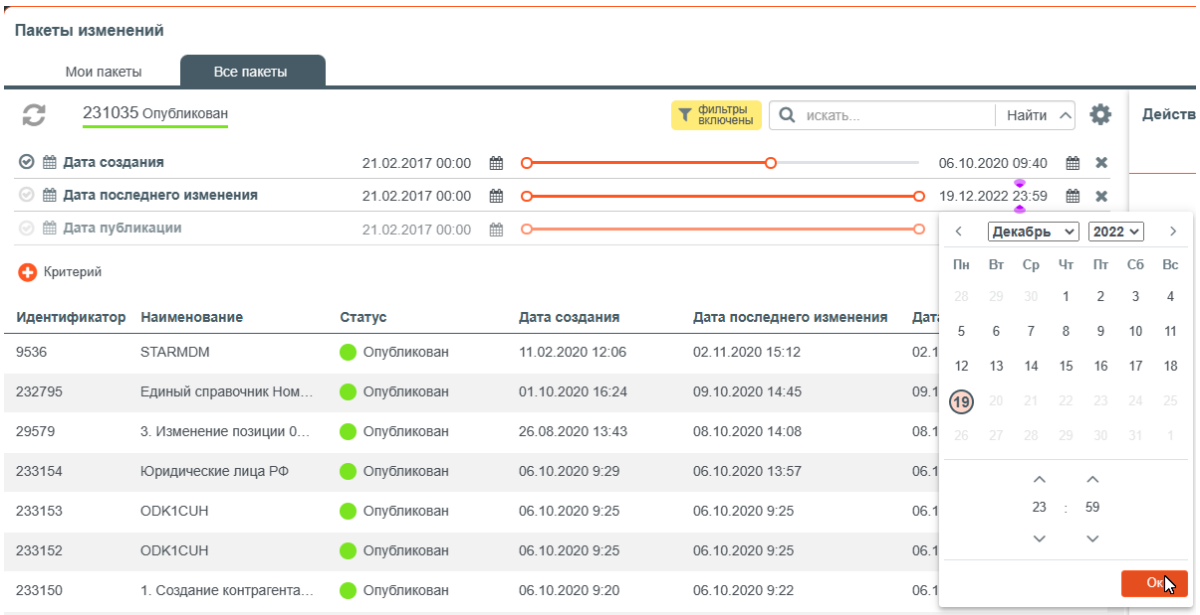


Рисунок 212 - Настройка дат в расширенном поиске

В окне присутствуют две вкладки - **Мои пакеты** и **Все пакеты**. На вкладке **Мои пакеты** расположены пакеты изменений созданные текущим пользователем. На вкладке **Все пакеты** показаны все пакеты изменений. Здесь в разделе **Действия** (Рисунок 213) располагаются дополнительные кнопки, доступность которых зависит от статуса пакета изменений, способа его создания (по заявке или без нее), а также от того, кем был создан пакет. С учетом этого пользователю доступны следующие действия:

- **Показать детализацию** - открывает отдельное окно, в котором можно ознакомиться с подробностями внесённых изменений;
- **Перевести на себя** - команда позволяет пользователю с правами Менеджера заявок переместить пакет изменений со статусом *Приостановлен* или *Активен* на свою вкладку для работы с ним;
- **Передать** - команда позволяет пользователю с правами Менеджера заявок передать пакет изменений со статусом *Приостановлен* другому эксперту НСИ. После активизации команды эксперт выбирается из окна диалога **Выберите пользователя**;
- **Активировать** - команда позволяет активировать ранее приостановленный пакет изменений, если он расположен на вкладке **Мои пакеты**;
- **Опубликовать** - команда позволяет опубликовать активный пакет изменений, если он расположен на вкладке **Мои пакеты**;

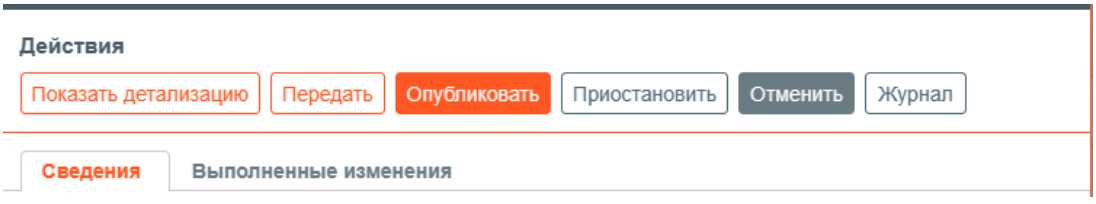


Рисунок 213 - Действия над пакетом изменений на вкладке **Мои пакеты**

- **Приостановить** - команда позволяет приостановить активный пакет изменений, если он расположен на вкладке **Мои пакеты**;

- **Отменить** - команда позволяет отменить приостановленный или активный пакет изменений, если он расположен на вкладке **Мои пакеты**;
- **Журнал** - открывает **Журнал работы пользователя**, где можно посмотреть изменения проведенные в пакете изменений, с точки зрения работы пользователя.

В правой части окна **Пакеты изменений** кроме команд разрешенными действиями располагается панель из двух вкладок (Рисунок 214):

- **Сведения** – на вкладке расположена информация о наименовании пакета (обычно совпадает с названием справочной группы, в которой был создан), статусе, датах создания и публикации, наличии заявки, по которой выполнялись изменения, а также описание пакета;
- **Выполненные изменения** - на вкладке располагается перечень объектов, которые были изменены. Характер внесенных изменений указывается пиктограммой перед наименованием.

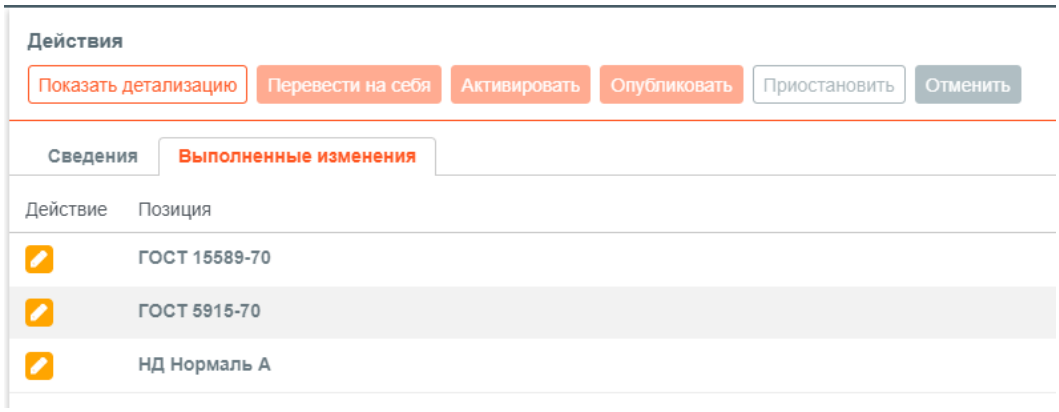


Рисунок 214 - Вкладка **Выполненные изменения** окна **Пакеты изменений**

Действия выполненные над объектами обозначаются специальными пиктограммами, расположенными в одноименной колонке.

Таблица 16. Пиктограммы пакета изменений

Пиктограмма	Состояние
	Изменение
	Удаление
	Создание

Функция детализации проведённых изменений (команда **Показать детализацию**) также может быть вызвана со стороны справочного объекта. Для получения доступа к детализации необходимо перейти в раздел **История**.

В окне детализации пакета изменений состав доступных вкладок зависит от того, какие объекты справочной системы подвергались изменениям.

Над вкладками располагаются команды доступа к фильтрации объектов в пакете изменения по виду выполненного действия (**Все**, **Создание**, **Изменение**, **Удаление**) или по справочной группам (**Группы**), в которых были произведены изменения объектов, при этом можно выделить родительскую группу и увидеть изменения, выполненные во всех подчиненных группах.

В детализации пакета изменений используется подсветка латиницы, чисел, концевых и двойных пробелов для более точного контроля выполненных изменений в строковых атрибутах.

При детализации пакетов изменений на отдельных вкладках буду представлены изменения вносимые в следующие объекты *Semantic MDM*:

- **Значения по позициям** (Рисунок 215) – в этом режиме будут показаны все позиции, которые подверглись редактированию в рамках рассматриваемого пакета. Для них будут указаны измененные атрибуты, а также старые и новые значения этих атрибутов. Щелчок

левой кнопки мыши по наименованию позиции позволит перейти в тот справочник, где она располагается;

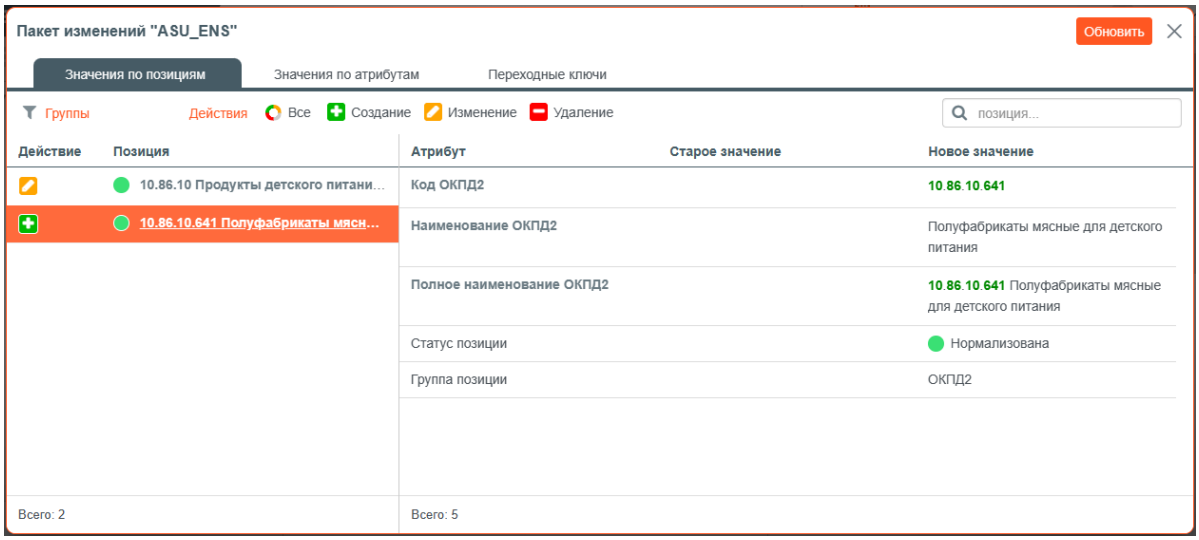


Рисунок 215 - Окно детализации пакета изменений на вкладке **Значения по позициям**

- **Значения по атрибутам** (Рисунок 216) – в этом режиме даётся полный перечень атрибутов, которые изменялись в текущем пакете. Установка курсора на наименовании атрибута позволяет посмотреть для каких позиций изменялись его значения, а также старые и новые значения. При наведении курсора на любую строку она временно расширяется, если значения в ней по длине не вписываются в размер столбца

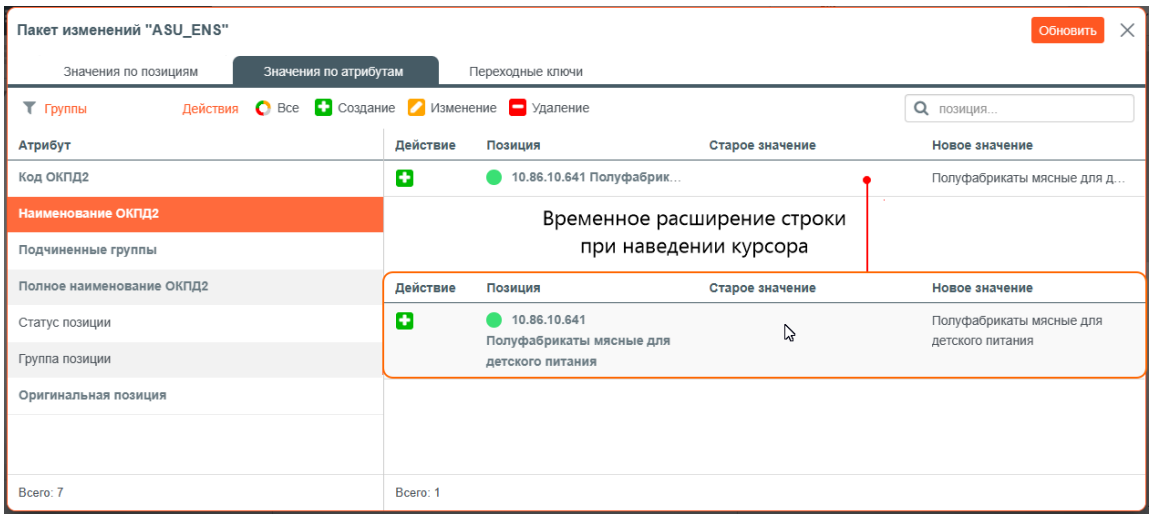


Рисунок 216 - Окно детализации пакета изменений на вкладке **Значения по атрибутам**

При просмотре детализации пакета изменений используется подсветка разницы между старым и новым значением для строковых атрибутов, текста и формул. Подсветка старого значения выполняется светло-красным цветом, а нового - светло-зеленым (Рисунок 217).

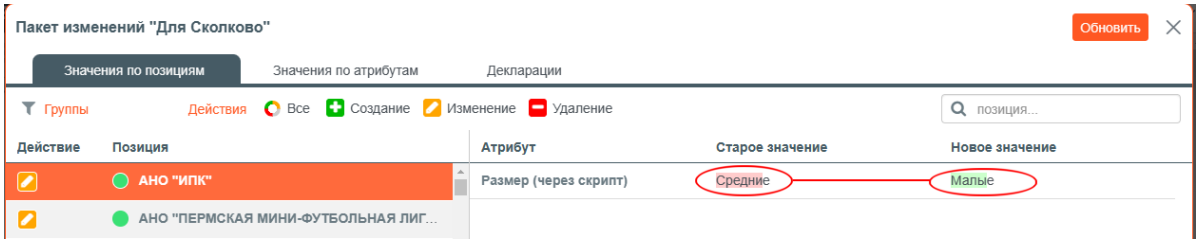


Рисунок 217 - Подсветка изменений между старым и новым значениями

- вкладка **Декларации** (Рисунок 218) содержит информацию по добавлению /изменению / удалению деклараций. В правой части таблицы располагаются подробные сведения об изменениях, внесенных в настройки атрибута. На вкладке выполняется сортировка данных по пути к группе;

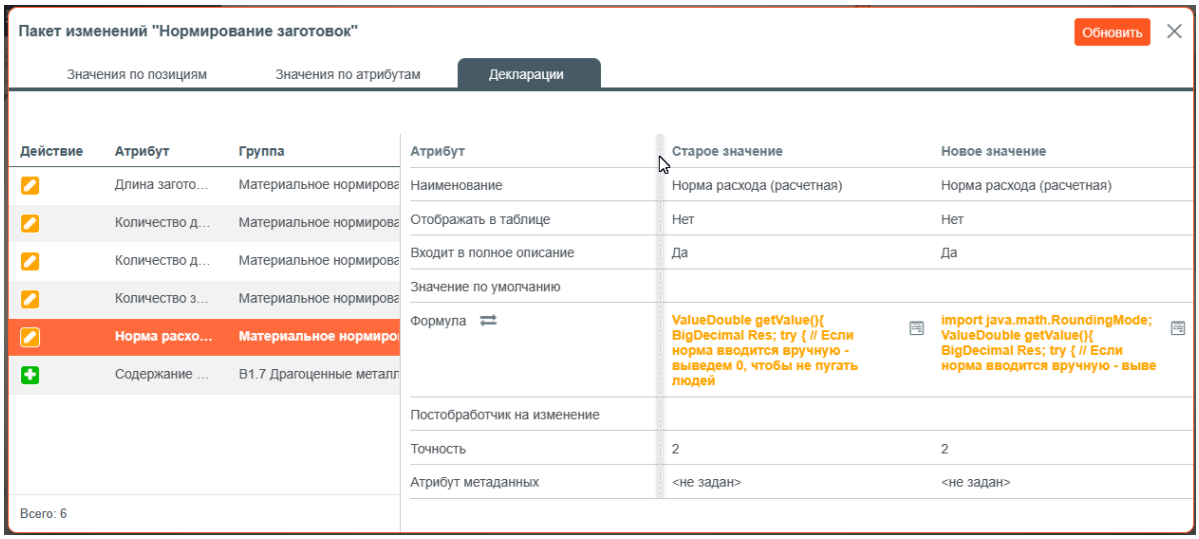


Рисунок 218 - Пакет изменений с детализацией по декларациям

- вкладка **Группы** (Рисунок 219) содержит информацию по добавлению / изменению / перемещению / удалению справочной группы. В правой части можно ознакомится с детальным представлением о внесенных

изменениях. На вкладке выполняется сортировка по наименованию групп;

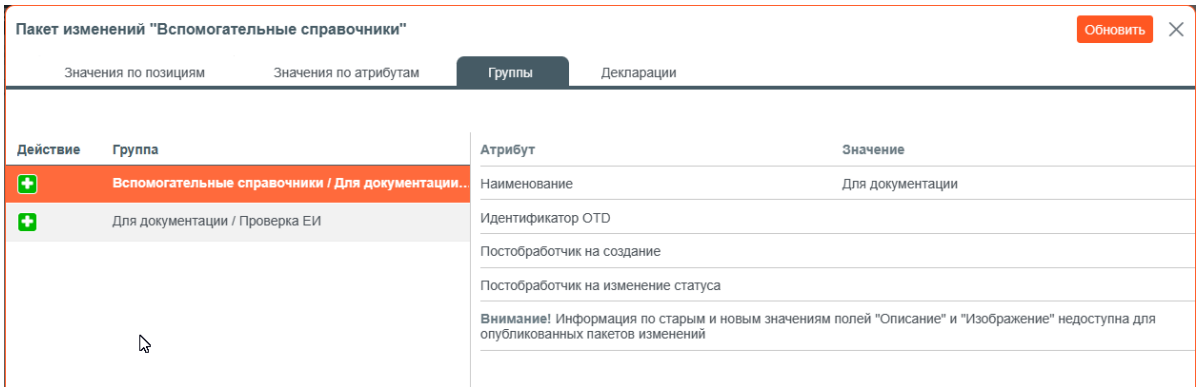


Рисунок 219 - Пакет изменений с детализацией по справочным группам

- вкладка **Функции проверки данных** (Рисунок 220) содержит информацию по добавлению / изменению / удалению ограничительных функций, настроенных для справочных групп;

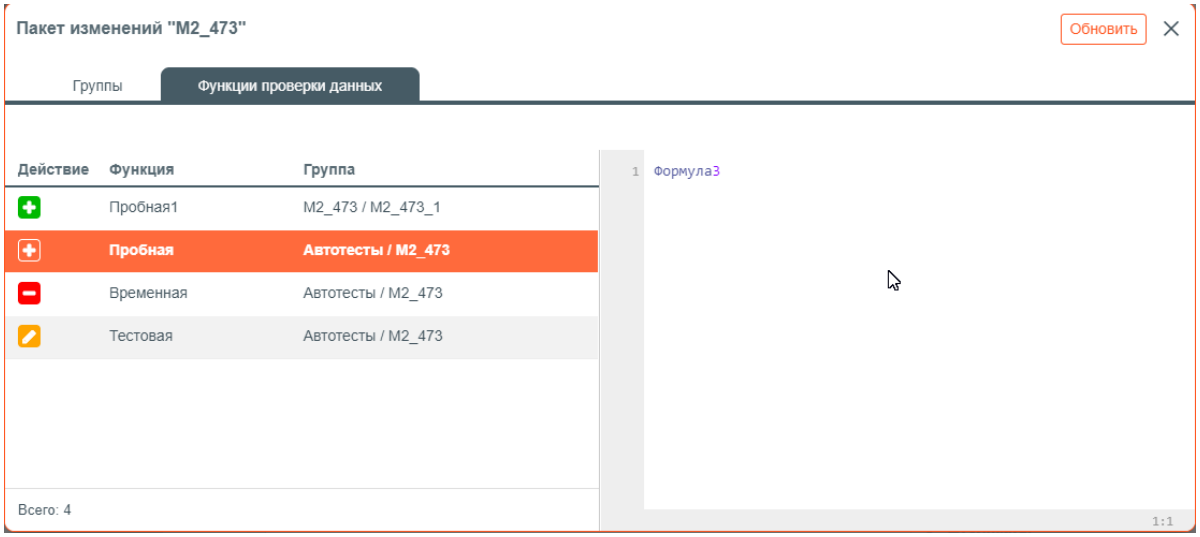


Рисунок 220 - Пакет изменений с детализацией
по функциям проверки данных

- вкладка **Обязательные/Уникальные** (Рисунок 221) содержит информацию по добавлению / удалению обязательных или уникальных атрибутов. В таблице указывается наименование справочной группы, для которой выполнялась настройка, а также наименования атрибутов, назначенных обязательными и / или уникальными;

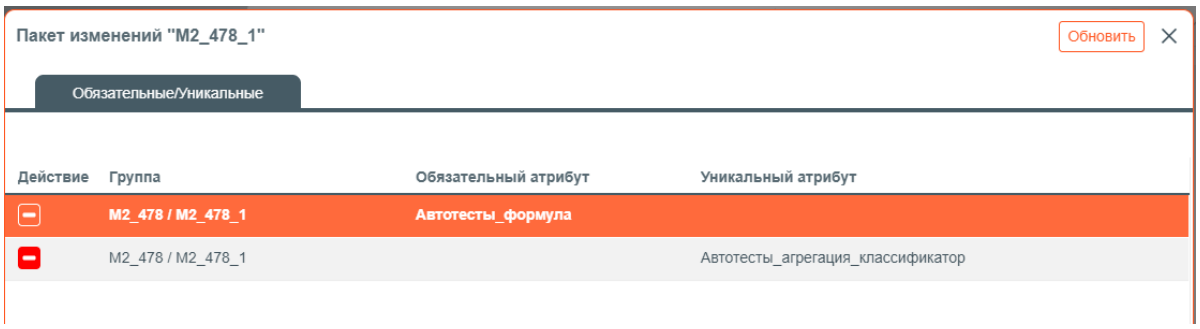


Рисунок 221 - Пакет изменений с детализацией
по обязательным / уникальным атрибутам

- вкладка **Ограничительные таблицы** (Рисунок 222, Рисунок 223) содержит информацию по изменению / созданию / удалению ограничительных таблиц. Содержимое вкладки будет зависеть от того опубликован пакет изменений или нет. Для опубликованного пакета изменений в таблице указываются сведения о выполненном действии, наименование таблицы, наименование справочной группы, в которой изменялась ограничительная таблица. Для неопубликованного пакета изменений в таблице будут представлены добавленные / измененные / удаленные строки ограничительной таблицы. Измененные значения выделяются жирным шрифтом. При наведении курсора на измененное значение во всплывающем окне (хинте) будет отображено старое значение и новое значение. На вкладке доступна сортировка по наименованию ограничительной таблицы;



Рисунок 222 - Опубликованный пакет изменений с детализацией по ограничительным таблицам

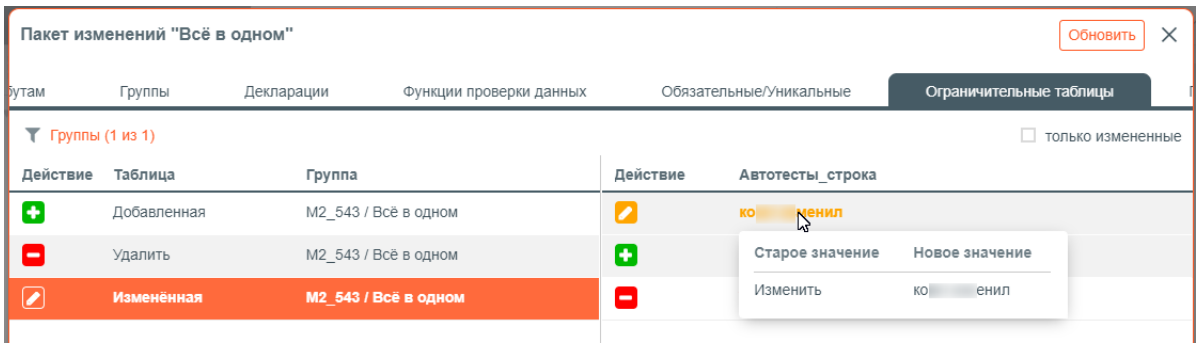


Рисунок 223 - Активный пакет изменений с детализацией
по ограничительным таблицам

- вкладка **Переходные ключи** (Рисунок 224) содержит информацию по добавленным / удаленным переходным ключам. В таблице указываются: наименование системы (в скобках указан идентификатор системы), идентификатор объекта во внешней системе, дубликат - принадлежит ли данный переходной ключ позиции - дубликату, позиция - наименование позиции, для которой был добавлен или удален переходной ключ, группа - наименование справочной группы, где находится позиция с данным переходным ключом.

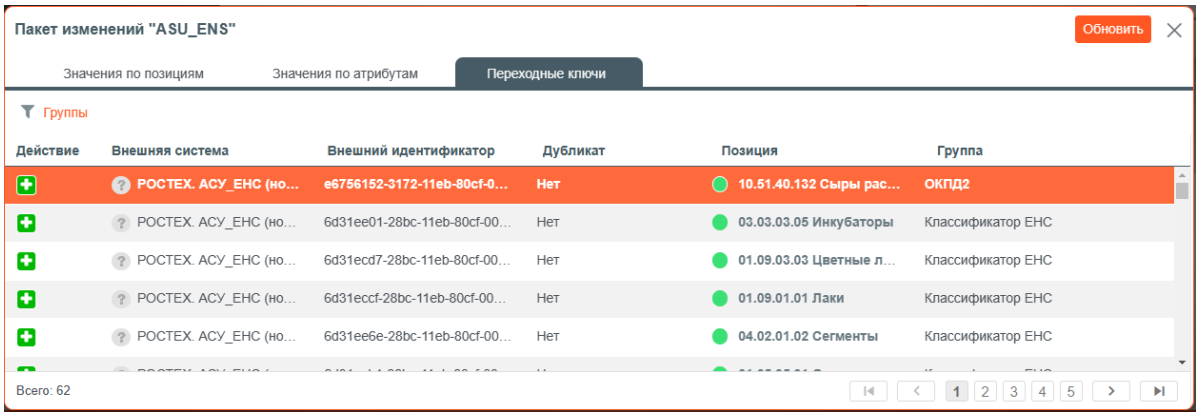


Рисунок 224 - Пакет изменений с детализацией по переходным ключам

Кнопка ***Обновить*** в правом верхнем углу окна пакета изменений предназначена для обновления сведений по транзакции, что особенно актуально, если просматриваемый пакет ранее находился в статусе приостановленного и в него могли быть внесены изменения.

3.14 Создание новых групп и позиций

Система обладает широкими возможностями по созданию справочных групп. Наполнение новых групп справочными объектами может происходить за счёт их создания вручную, с помощью ограничительных таблиц, путем загрузки из файлов формата XML и XLS, а также путём перемещения уже существующих объектов из одних групп в другие.

Процесс создания новых справочных групп или новых позиций ведётся в рамках работы с пакетами изменений.

3.14.1 Создание новой справочной группы

Для создания новых веток в дереве справочников необходимо входить в группу пользователей, обладающих правами на редактирование данных. Настройка групп и прав пользователей осуществляется системным администратором Semantic MDM.

Для создания новой группы (ветки) в дереве справочников необходимо:

- перейти в режим редактирования создав новый пакет изменений;
- установить курсор на нужной группе справочников или на корневом элементе дерева;
- нажать кнопку ***Добавить группу*** (Рисунок 225) или вызвать из контекстного меню одноименную команду;

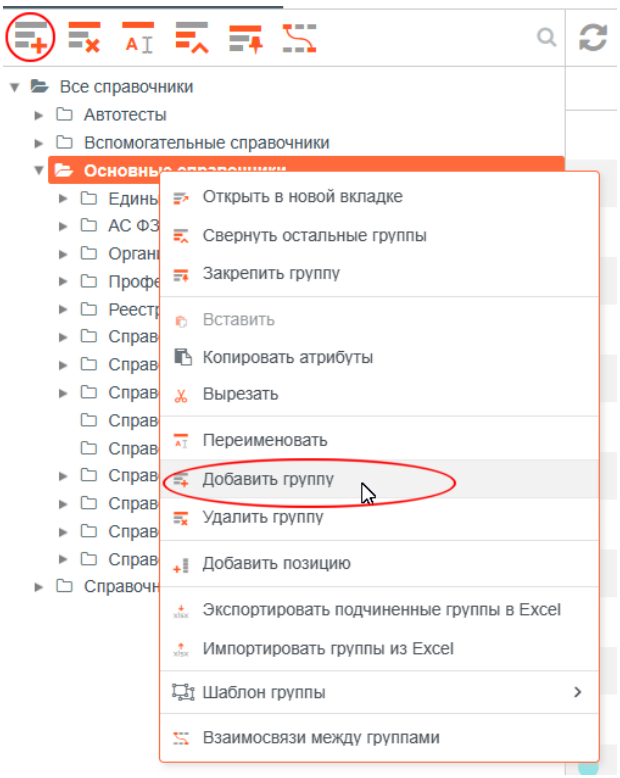


Рисунок 225 - Добавление новой группы

- перейти в карточку позиции и заполнить сведения о наименовании новой группы и дать её описание (не обязательно) (Рисунок 226);

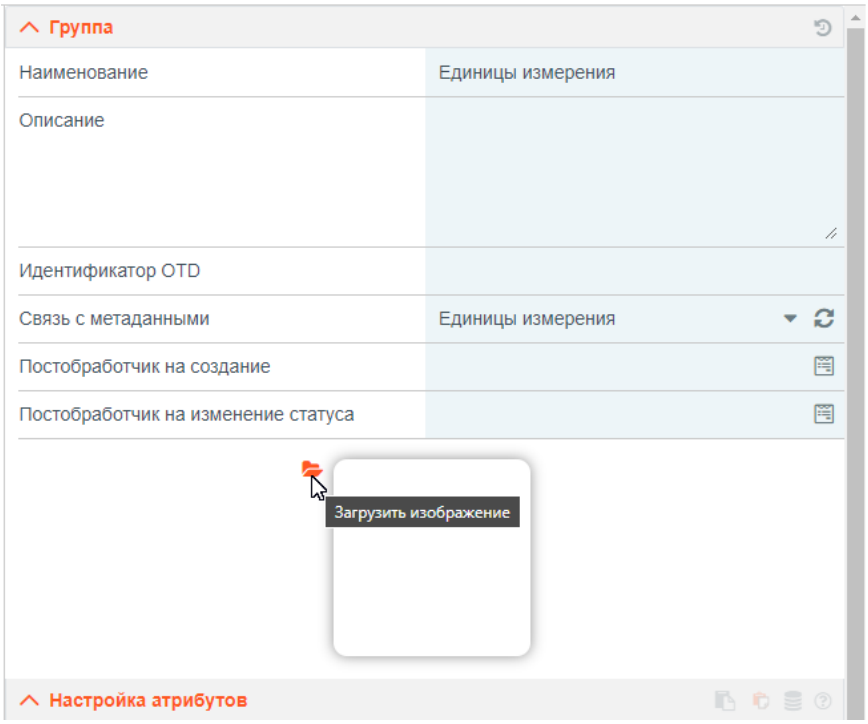


Рисунок 226 - Описание справочной группы

- указать идентификатор OTD, если он применяется;
- в ячейке **Связь с метаданными** можно выбрать один из системных справочников - *Единицы измерения* либо *Пользователи*. Функционал используется только для создания пользовательских справочников на основе системных! Если в списке ничего нет, то это означает, что пользовательские справочники уже были созданы. Для обычных справочников не заполняется;
- **Постобработчик на создание** - указывается скрипт, применяемый для постобработки позиций, создаваемых в справочной группе. Скрипт GROOVY создается в редакторе функций постобработки, который вызывается кнопкой  в этой ячейке;
- **Постобработчик на изменение статуса** - указывается скрипт, применяемый для постобработки позиции, для которой был изменен статус. Скрипт GROOVY создается в редакторе функций постобработки, который вызывается кнопкой  в этой ячейке (см. п. 3.14.8).

Если предполагается, что справочник будет доступен со стартовой страницы *Системы*, то ему нужна пиктограмма, которую следует загрузить из файла (JPEG, PNG, BMP) объёмом не более 5 Мб. Для этого нужно нажать на кнопку ***Загрузить изображение*** и указать файл с изображением.

Далее следует перейти к выбору атрибутов группы и настройкам прав доступа.

3.14.2 Перемещение и удаление групп

Для того, чтобы удалить справочную(ые) группу(ы) следует:

- перейти в режим редактирования создав новый пакет изменений;
- установить курсор на нужной группе справочников или выделить несколько групп (в пределах одной родительской группы) щелчком левой кнопки мыши при нажатой клавише <Ctrl>;

- нажать кнопку ***Удалить группу*** на панели инструментов дерева или вызвать из контекстного меню одноименную команду. Для нескольких выделенных групп следует использовать команду контекстного меню **Удалить выбранные** (Рисунок 227).

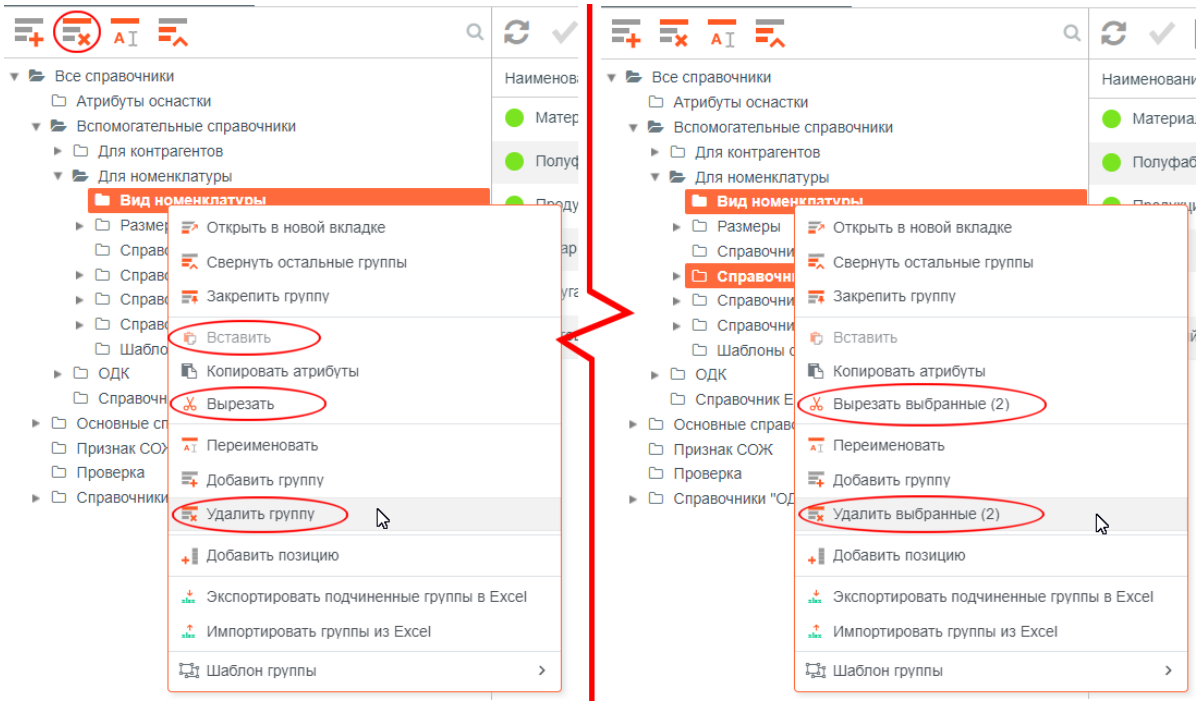


Рисунок 227 - Удаление и перемещение справочных групп

Для того, чтобы переместить справочную(ые) группу(ы) следует:

- перейти в режим редактирования создав новый пакет изменений;
- установить курсор на нужной группе справочников или выделить несколько групп (в пределах одной родительской группы) щелчком левой кнопки мыши при нажатой клавише <Ctrl>;
- вызвать из контекстного меню **Вырезать** или **Вырезать выбранные** (если перемещается несколько групп);
- установить курсор на корневом элементе дерева или на новой родительской группе и выбрать в контекстном меню команду **Вставить**. Выбранные группы будут перемещены в новое расположение.

3.14.3 Регистрация новых справочников

Создание новых справочных групп выполняется экспертами НСИ, однако, для доступа к ним со стартовой страницы они должны быть зарегистрированы в представлении **Конфигурация данных** в группе **Справочники**.

Для того, чтобы объединить справочные группы в единый справочник и открыть к нему доступ со стартовой страницы следует:

- установить курсор на группе **Справочники** в представлении **Конфигурация данных** и перейти на сетку данных;
- нажать кнопку ***Добавить***, расположенную над областью таблицы или выбрать команду **Добавить справочник** в контекстном меню (Рисунок 228);

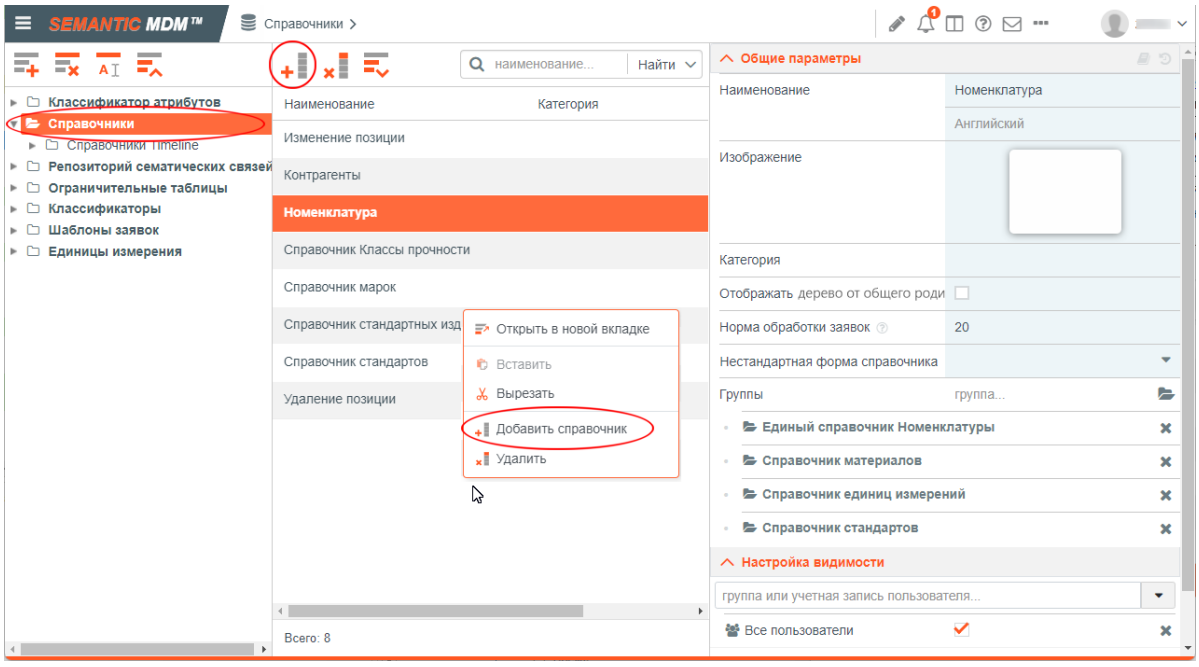


Рисунок 228 - Регистрация справочника

- перейти в карточку группы и заполнить необходимые ячейки:
  - **Наименование** – название справочника на русском и английском (опционально) языке;

- **Изображение** – можно добавить изображение в виде пиктограммы. Можно использовать файлы форматов BMP, PNG, JPEG размером не более 5 Мб;
- **Категория** – название общей группы справочников на стартовой странице, в которую будет входить текущий;
- **Отображать дерево от общего родителя** - при включении данной опции будет построено дерево, начиная с общего родителя всех выбранных справочных групп. Без неё - дерево будет содержать только выбранные и их дочерние группы;
- **Норма обработки заявок** - указывается количество заявок, которые *Эксперт НСИ* должен обработать за один рабочий день по данному справочнику. Значение используется при расчете SLA и KPI работы подразделения НСИ и отдельных сотрудников;
- **Нестандартная форма справочника** - позволяет выбрать из выпадающего списка альтернативную реализацию представления справочника, которая загружается из расширения, зарегистрированного в *Системе*;
- **Группы** – выбрать справочные группы, которые будут входить в создаваемый справочник;

  - в настройках видимости выбрать группы или отдельных пользователей, которые могут видеть справочник.

3.14.4 Декларация атрибутов для позиций справочной группы

Объекты в справочной группе описываются набором атрибутов - декларациями, которые выбираются на уровне группы. Атрибуты создаются в представлении **Конфигурация данных** или могут быть созданы непосредственно при настройке группы, для чего используется специальное всплывающее окно **Атрибуты**. В *Semantic MDM* атрибуты являются

универсальными (сквозными), то есть один и тот же атрибут может использоваться в любом количестве справочных групп.

В разделе **Настройка атрибутов** карточки позиции выполняется выбор (декларация) и настройка атрибутов для объектов, входящих в группу.

После создания новой справочной группы в разделе **Настройка атрибутов** может присутствовать один и более атрибутов, которые нельзя удалить и содержание которых наследуется из родительской группы. При наведении курсора на значок в правой части строки с именем атрибута, можно узнать из какой группы унаследованы его настройки. Если это необходимо, то настройки унаследованного атрибута могут быть изменены. Для этого следует щёлкнуть левой кнопкой мыши на и нажать кнопку ***Перекрыть*** (Рисунок 229), а затем отредактировать необходимые свойства декларации.

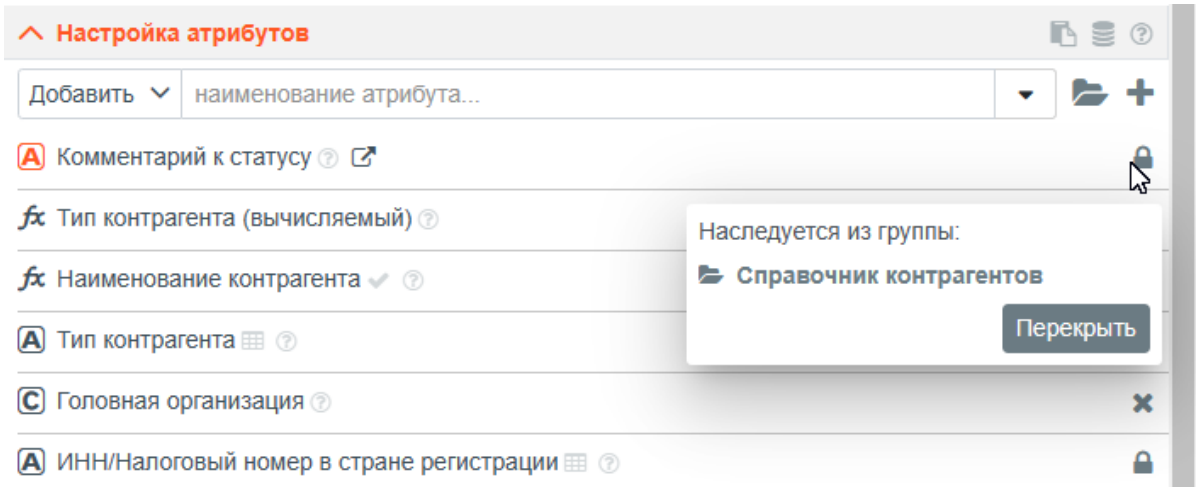


Рисунок 229 - Унаследованный атрибут группы

Рядом с наименованием каждого из унаследованных или собственных атрибутов располагаются служебные пиктограммы, характеризующие данный атрибут и выступающие в качестве кнопок в режиме управления пакетами изменений. Пиктограммы (Таблица 17) позволяют узнать тип атрибута, наличие заполненных значений, перейти к диалогам с дополнительными настройками и редакторам.

Таблица 17. Пиктограммы типа атрибута

Пиктограмма	Значение
	Декларация простого атрибута
	Декларация атрибута связи (агрегации или композиции)
	Декларация атрибута, значение которого вычисляется по формуле
	Значение атрибута является наименованием позиции (значение выводится на сетке данных в колонку **Наименование**). Признак может быть установлен только для одного атрибута в справочной группе.
	Кнопка Выбрать позицию. Доступна для атрибутов связи типа Агрегация
	Значения отсутствуют. Пиктограмма появляется рядом с декларацией атрибута, для которого в справочной группе не задано ни одного значения
	При наведении курсора на данную пиктограмму всплывает окно с наименованием атрибута, его типом, источником и описанием.
	Пиктограмма-кнопка (меняет цвет на красный при наведении курсора). Позволяет открыть атрибут в подсистеме **Конфигурация данных**.
	Кнопка ***Редактор формул***. Вызывает редактор для разработки шаблона строки или скрипта Groovy, используемых при вычислении значения атрибута.
	Кнопка ***Ссылка на формулу***. Открывает одноименное окно диалога для выбора справочной группы, в которой располагается шаблон вычисления значения атрибута.
	Кнопка вызывает окно с календарем в поле атрибута типа *Дата* для заполнения значения путем выбора даты.
	Атрибут используется в ограничительных таблицах
	Атрибут унаследован от родителя. Во всплывающем окне указывается наименование группы откуда унаследован атрибут.
	Кнопка ***Показать таблицу.*** Открывает дополнительное окно для работы с композируемыми позициями текущей (композирующей, родительской) позиции справочника.
	Кнопка ***Создать***. Создает в карточке родительской позиции новую композируемую позицию для заполнения значений её атрибутов.
	Кнопка ***Открыть взаимосвязи***. Открывает **Редактор связей** для создания, удаления или изменения связей текущей позиции с композируемыми позициями.
	Кнопка настройки цветов для категорий и множественных агрегаций / композиций.
	Кнопка ***Автоматический запуск***. Доступна только для ячейки *Постобработчик на изменение*. Активизация данной кнопки определяет автоматическое срабатывание функции постобработки при изменении значения позиции в этой ячейке.
	Пиктограмма, показывающая наличие исторических значений у атрибута, которые показываются во всплывающей панели при наведении курсора на пиктограмму. В режиме управления пакетами

	изменений превращается в кнопку, вызывающую диалог для заполнения сведений об исторических значениях.

Для того, чтобы продекларировать новый атрибут нужно начать вводить название атрибута в ячейке **Наименование атрибута** при этом будет развернут выпадающий список зарегистрированных в *Системе* атрибутов, причём по мере ввода символов список атрибутов будет фильтроваться (Рисунок 230). Выбрать атрибут можно, не вводя данные, путём прокручивания списка с помощью колеса мыши.

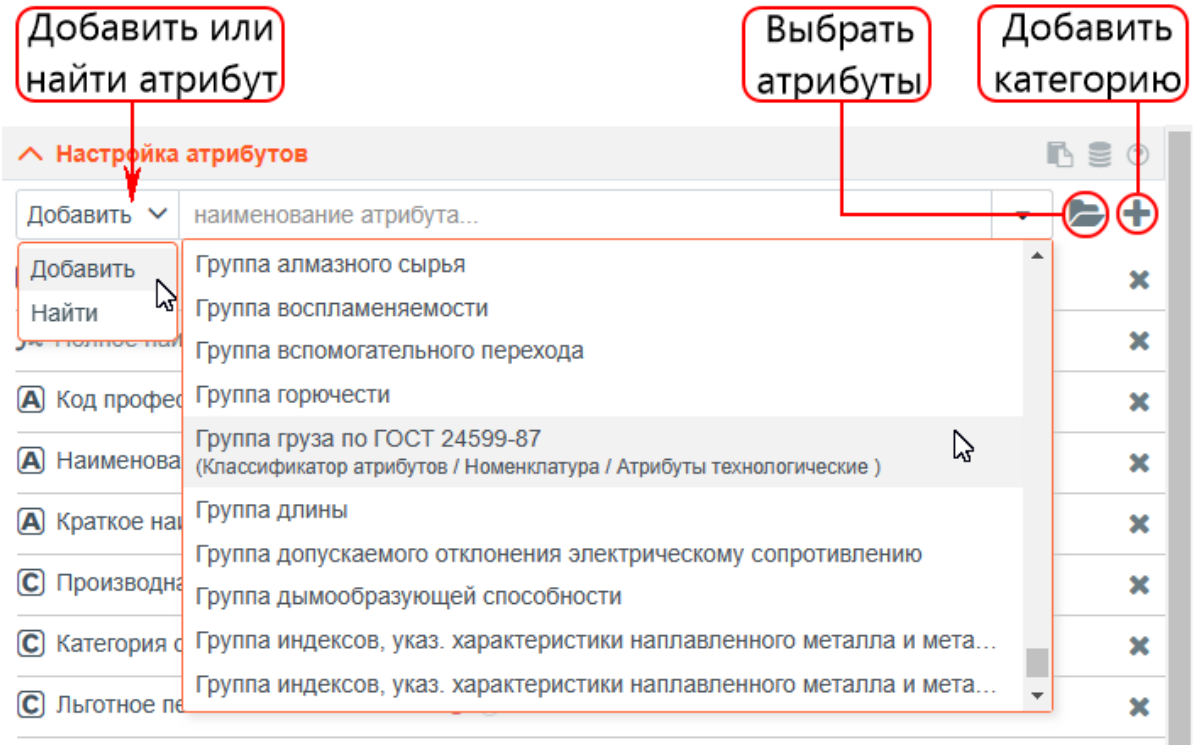


Рисунок 230 - Выбор нового атрибута

Слева от ячейки выбора атрибута находится переключатель которые позволяет не добавлять новую декларацию, а выполнять поиск атрибута среди уже продекларированных. Для этого из выпадающего списка переключателя нужно выбрать команду **Найти** (Рисунок 231). Если выбрана команда **Найти**, то в карточке остаются только те атрибуты (и их категории), которые соответствуют критерию поиска.

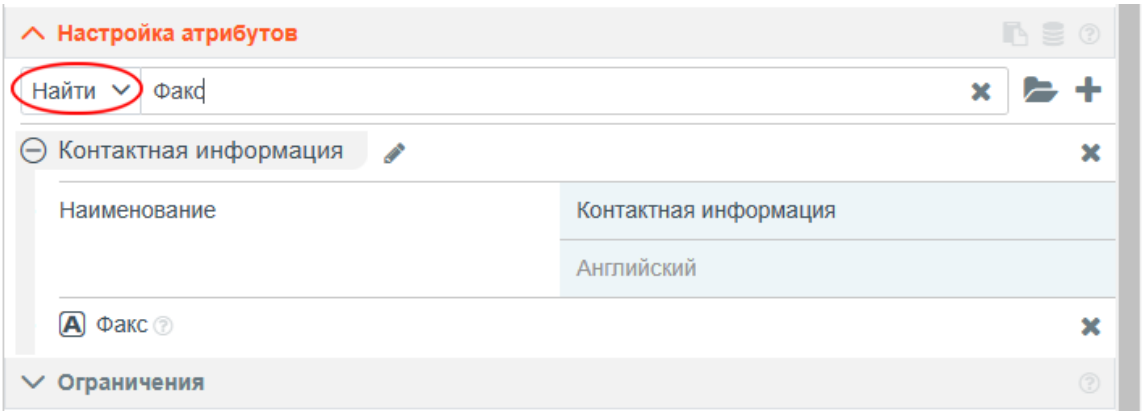


Рисунок 231 - Поиск атрибута в карточке справочной группы

Вторым вариантом выбора атрибута является обращение к **Классификатору атрибутов** (Рисунок 232), где они представлены в упорядоченном виде. Для этого нужно нажать кнопку ***Выбрать атрибуты*** и, в открывшемся окне **Атрибуты**, выбрать группу и атрибуты из её состава, установив маркер в ячейке слева от наименования, а затем нажать кнопку ***Ок***. Выбранные атрибут будут добавлены в список. Над списком атрибутов расположена ячейка быстрого поиска по всему классификатору. Для поиска необходимо начать вводить наименование атрибута. По мере ввода символов *Система* будет фильтровать отображаемые данные.

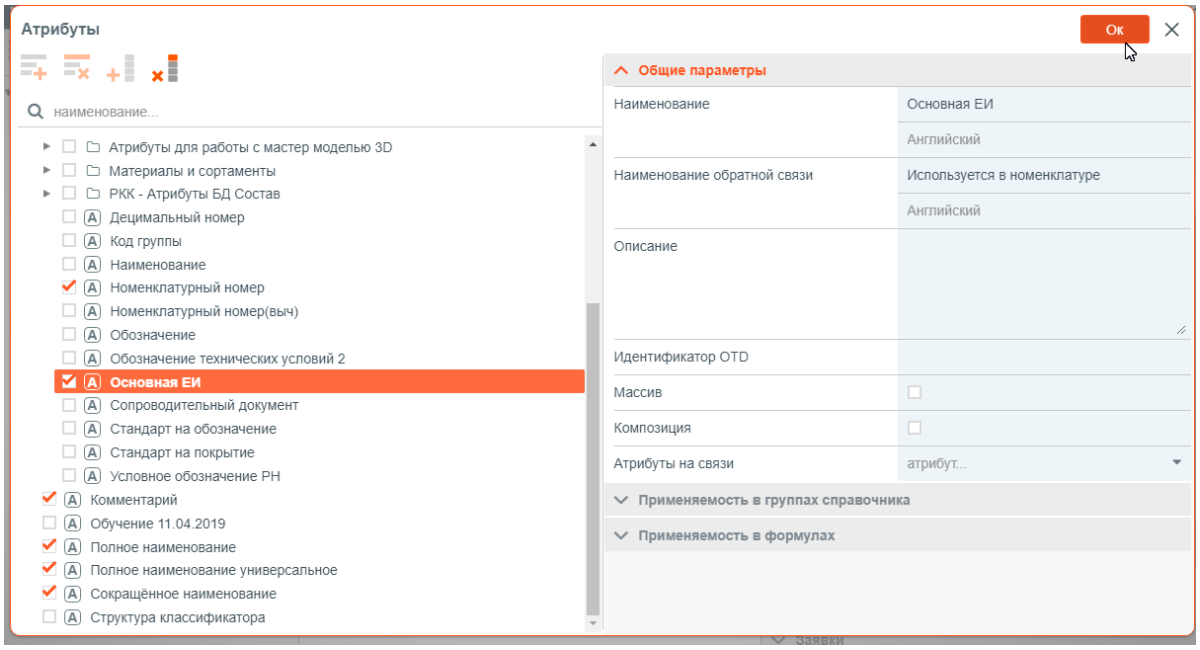


Рисунок 232 - Классификатор атрибутов

В этом же окне можно создать и затем выбрать новый атрибут.

Кнопка ***Добавить категорию*** позволяет объединить атрибуты в смысловые группы, например, как показано на рисунке ниже. Чтобы создать новую категорию следует нажать кнопку ***Добавить категорию***, затем заполнить ячейку **Наименование**.

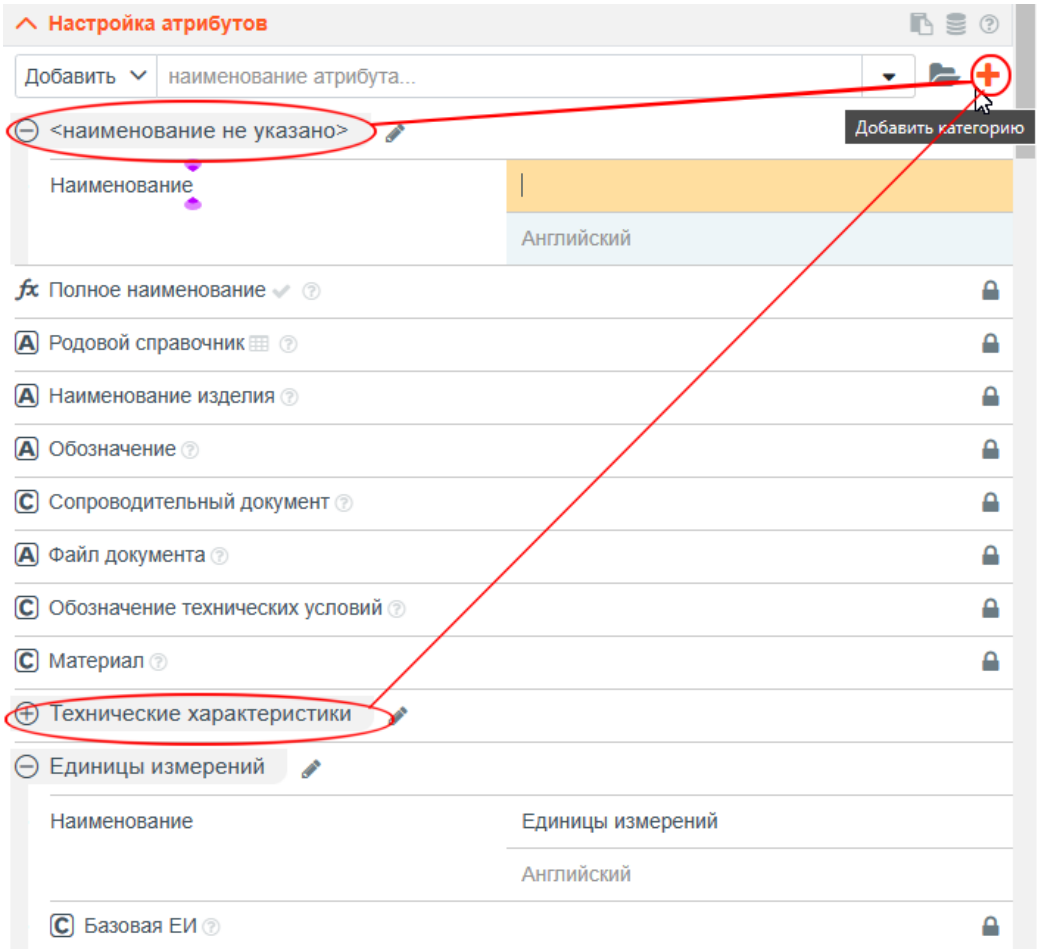


Рисунок 233 - Категории атрибутов

Чтобы добавить атрибуты в определённую категорию, их в начале нужно просто добавить в список атрибутов, а затем, зажав левую кнопку мыши на наименовании атрибута, перенести на название категории (функция Drag&Drop) (Рисунок 234). Для добавления в категорию сразу нескольких атрибутов необходимо выделить их в списке щелчком левой кнопки мыши с зажатой клавишей <*Ctrl*>. В процессе перемещения строка будет подсвечена розовым цветом, как показано ниже. При перемещении нескольких

выделенных атрибутов строки подсвечиваются светло-оранжевым цветом. Аналогичным образом можно извлечь атрибуты из категории.

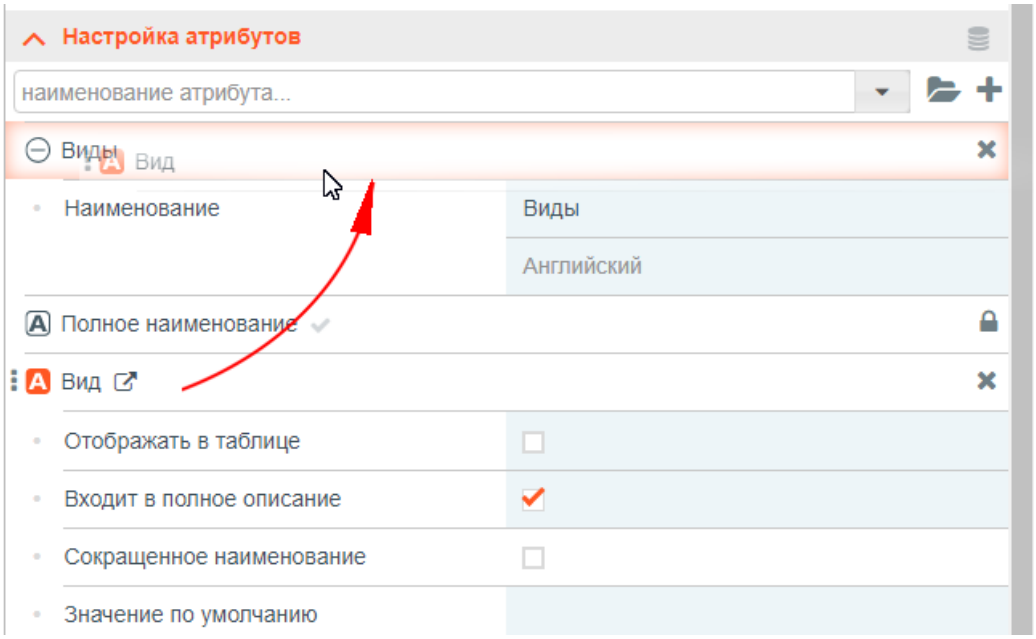


Рисунок 234 - Перемещение атрибута в категорию

Если при извлечении атрибутов из категории (разгруппировании) категория останется пустой, она будет автоматически удалена из списка.

Категории в карточке для наглядности подсвечиваются цветом (Рисунок 235). По умолчанию цвета настраиваются системным администратором, но при необходимости в любой справочной группе могут быть определены свои цвета для разных категорий. Для изменения цвета категории по умолчанию следует нажать кнопку **Настроить цвет** (кнопка доступна только пользователям административными правами), расположенную справа от наименования категории, и настроить цвет в открывшемся окне. Цвет может быть указан непосредственно курсором с использованием цветовой шкалы, цветового поля и шкалы прозрачности либо введен в виде формулы Hex / RGB A / HSL A. Переключение формулы цвета выполняется кнопкой справа от поля формулы. Кнопкой ***Ок*** необходимо подтвердить выполненные

настройки. Чтобы вернуться к базовому цвету следует в окне выбора цвета нажать кнопку ***По умолчанию***.

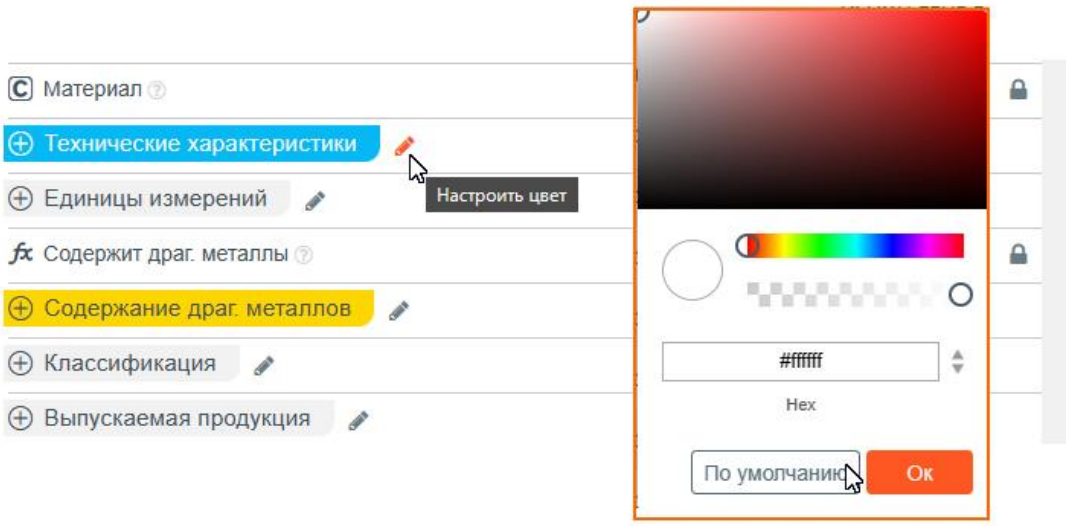


Рисунок 235 - Настройка подсветки категорий в карточке

3.14.5 Копирование атрибутов справочной группы

В некоторых случаях при разработке справочных групп возникает необходимость многократно использовать одни и те же атрибуты с одинаковыми настройками в разных ветках. Например, типоразмеры, классификационные коды, ссылки на другие справочники и т.п. Для сокращения времени, затрачиваемого на добавление и настройку таких атрибутов можно воспользоваться функцией копирования.

Чтобы скопировать атрибуты из одной группы в другую необходимо:

- перейти в режим редактирования создав новый пакет изменений;
- установить курсор на нужной группе (ветке) справочника, откуда предполагается копировать атрибуты;
- в контекстном меню выбрать команду **Копировать атрибуты** или нажать пиктограмму - кнопку ***Копировать атрибуты*** в разделе **Настройка атрибутов** карточки группы;

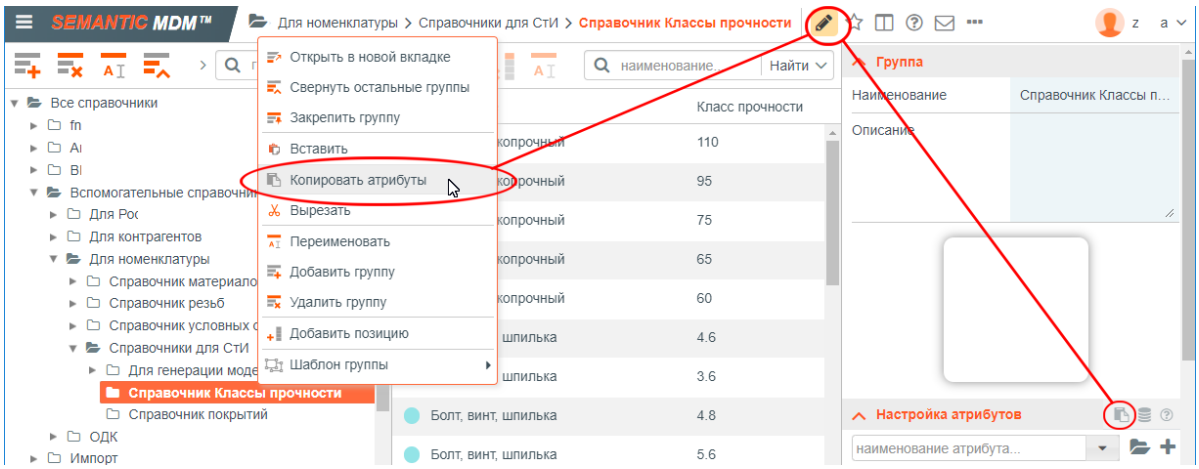


Рисунок 236 - Вызов команд **Копировать атрибуты** и **Вставить**

- установить курсор на в дереве на наименовании группы, в которую копируются атрибуты и выбрать из контекстного меню команду **Вставить** или нажать пиктограмму - кнопку ***Вставить атрибуты*** в разделе **Настройка атрибутов** карточки группы (пиктограмма появляется только при наличии скопированных атрибутов);
- в окне **Вставка атрибутов** (Рисунок 237) щелчком левой кнопки мыши снять маркеры у ненужных атрибутов и нажать кнопку ***Ок***.

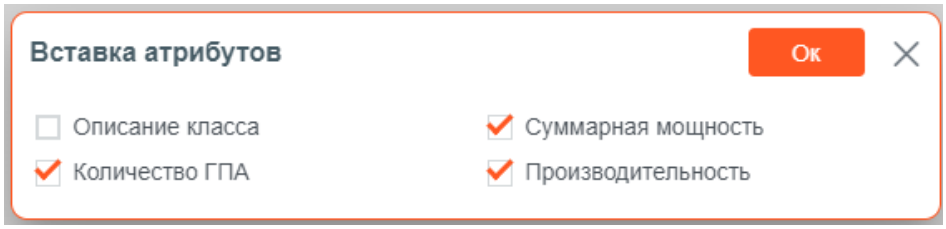


Рисунок 237 - Окно **Вставка атрибутов** для выбора копируемых атрибутов

Выбранные атрибуты будут скопированы в указанную группу вместе со всеми настройками, включая формулы и агрегации. Дальнейшая их настройка ведется по общим правилам, изложенным в п. 3.13.7.

3.14.6 Перемещение атрибутов

Атрибуты располагаются в списке в порядке их добавления. При создании новой категории она размещается в верхней части перед

унаследованными атрибутами. При необходимости положение любого атрибута и категории в списке может быть изменено с помощью функции Drag&Drop.

Чтобы переместить атрибут, нужно навести курсор на строку с наименованием атрибута, зажать левую кнопку мыши и потянуть атрибут на новое место. В процессе перемещения строка с атрибутом приобретёт розовый цвет. Для добавления в категорию сразу нескольких атрибутов необходимо выделить их в списке щелчком левой кнопки мыши с зажатой клавишей <*Ctrl*>. При перемещении нескольких выделенных атрибутов строки подсвечиваются светло-оранжевым цветом.

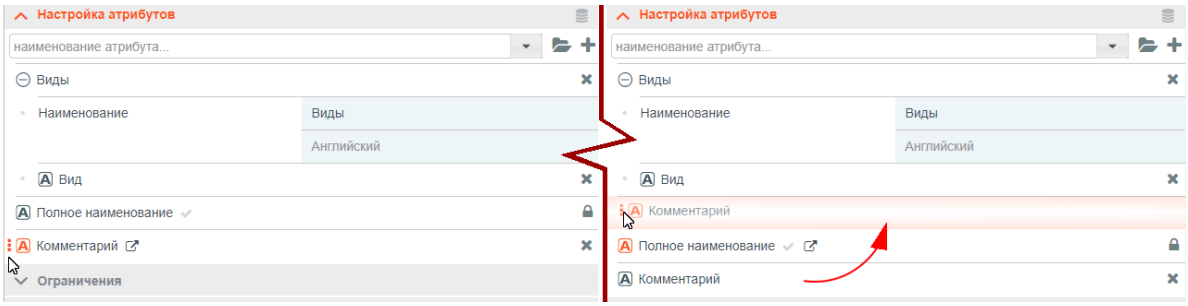


Рисунок 238 - Перемещение атрибута в списке

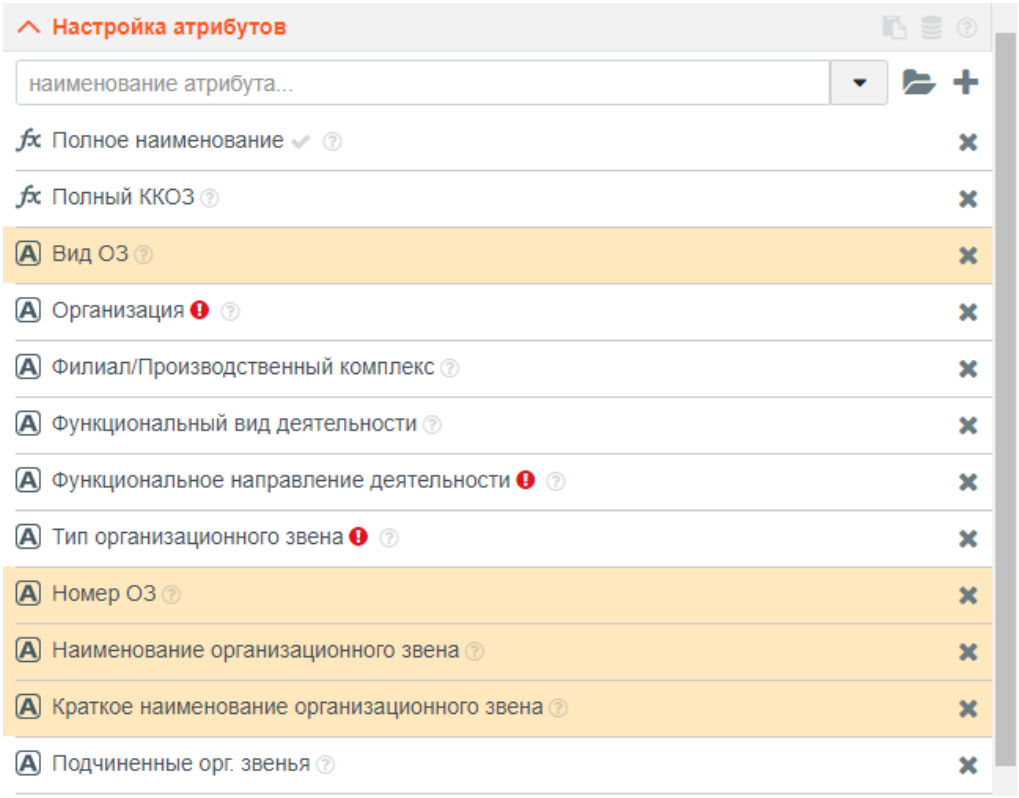


Рисунок 239 - Выбор нескольких атрибутов (деклараций) для перемещения

3.14.7 Настройка атрибутов группы

Тип каждого атрибута определяется при его создании и диктуется характеристикой, которую он описывает. При декларации атрибута в справочной группе любой атрибут получает набор стандартных настроек (Рисунок 240), которые определяют способ его отображения, значение по умолчанию, права доступа и т.д. О некоторых особенностях настроек отдельных типов атрибутов изложено в разделе «Классификатор атрибутов».

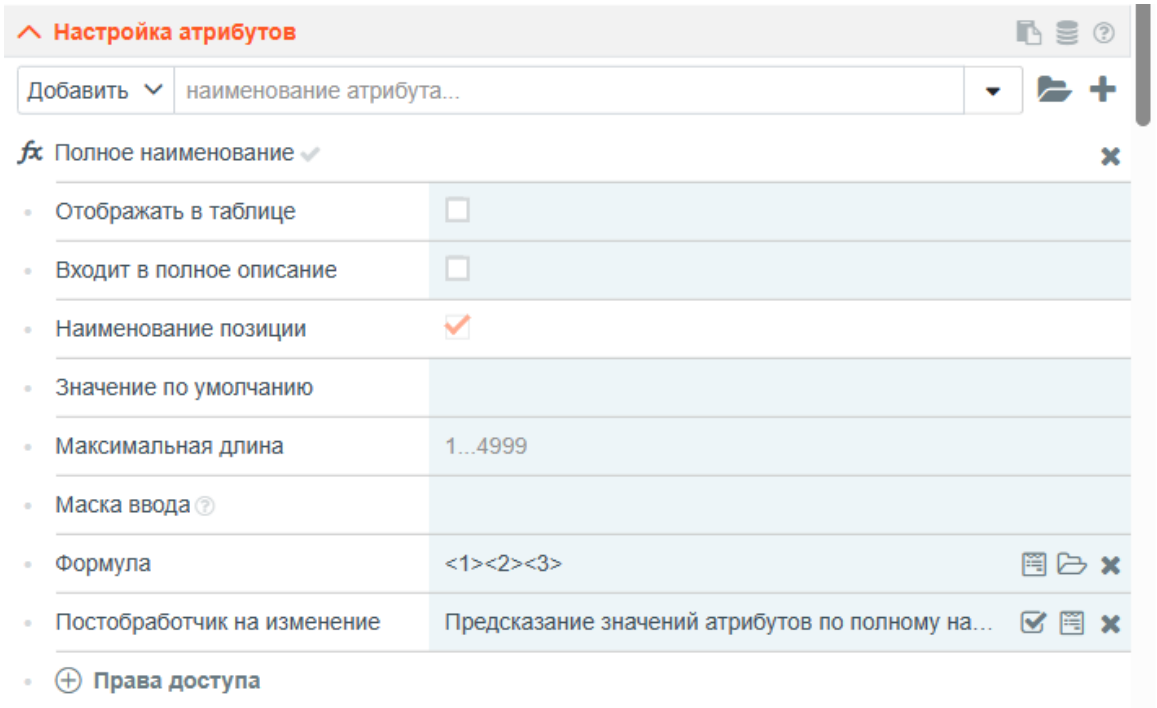
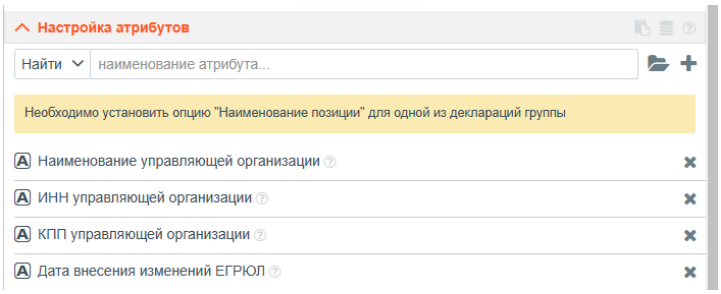


Рисунок 240 - Настройки атрибута справочной группы

После добавления атрибута следует выполнить необходимые настройки (Таблица 18). Количество настроек зависит от типа атрибута.

Таблица 18. Настройки атрибута

Настройки атрибута	Комментарий
Наименование	По умолчанию имя атрибута фиксируется при его создании в **Классификаторе атрибутов**, однако при декларировании атрибута в справочной группе его можно переименовать. Для этого в ячейке водится новое имя. Ввод наименования на английском языке не является обязательным
Отображать в таблице	При установке данного маркера атрибут выводится в таблицу данных сразу после создания объекта

Входит в полное описание	При установке данного маркера значение атрибута используется в полном описании позиции
Область	Настройка доступна только для атрибутов связи (агрегации / композиции). С помощью кнопки ***Добавить*** здесь необходимо указать справочник(и), значения атрибутов которого(ых) могут быть присвоены текущему атрибуту.
Наименование позиции	Установленный маркер означает, что значение этого атрибута является наименованием объекта НСИ, которое будет выведено в системную колонку **Наименование** в таблице данных. Возможность установки маркера определяется типом атрибута. Внутри группы возможен только один атрибут с таким свойством. Если ни одна из деклараций не имеет этой опции, то в карточке справочной группы в режиме управления пакетами изменений будет выведено следующее предупреждение  Предупреждение об отсутствии настройки «Наименование позиции» Системный администратор может отключить обязательное отображение системной колонки **Наименование**. В этом случае в таблице отображаются только те атрибуты, у которых в декларации установлена опция **Отображать в таблице**. Если ни одна декларация не имеет этой опции, то в таблице видна только системная колонка со статусом позиции.
Значение по умолчанию	Указанное в ячейке значение применяется для всех создаваемых в группе объектов по умолчанию. При необходимости оно может быть заменено другим
Максимальная длина	Поле доступно только для простых атрибутов с типом данных *Строка*. В ячейке можно указать максимальную разрешенную длину строкового значения. Если символов окажется больше разрешенного количества, то они будут обрезаны автоматически
Маска ввода	Ограничивает возможность произвольного ввода значений путём установки маски.
Формула	Открывает редактор формул, где можно ввести логические и иные формулы для вычисления значения данного атрибута. Кроме шаблона строки для определения значения может быть использован скрипт Groovy. Формула также может быть добавлена в виде ссылки.
Постобработчик на изменение	Специальная формула, разрабатываемая в **Редакторе функций постобработки** в виде скрипта Groovy, для выполнения обогащения позиции / группы новыми значениями атрибутов, для реализации функции *AfterInsert*, которая позволяет заполнить атрибуты позиции вычисленными значениями по умолчанию, а также создать и сразу заполнить композиции. Доступно только для вычисляемых строковых атрибутов, простых строковых атрибутов и агрегаций!

	Пиктограмма **Автоматический запуск** позволяет включить / выключить автоматический запуск постобработки при редактировании позиции. Если постобработчик не имеет автоматического запуска, то его выполнение запускается специалистом вручную по кнопке после редактирования значений.
Точность	В ячейке указывается количество цифр после запятой при указании значения атрибута типа «число с плавающей точкой».
Единица измерения	Из выпадающего списка выбирается единица измерения. Измеряемая сущность назначается при создании атрибута. Свойство доступно только для простых атрибутов с типом данных *Число с плавающей точкой*.
Атрибут метаданных	Определяет атрибут, как характеристику единицы измерения. Может относится к **Наименованию**, **Измеряемой сущности**, **Обозначению**, **Формуле пересчёта** или **Обратной формуле пересчёта**. Появляется только в случае, когда для группы настроена связь с метаданными.
Изменять период действия позиции	Свойство доступно только для простых атрибутов с типом данных *Дата*. Вы выпадающем списке свойства можно задать варианты: • *<не изменять>* - дата, указанная в атрибуте не влияет на историчность позиции; • *Дата начала действия* – дату, указанную в атрибуте считать датой начала периода действия позиции; • *Дата окончания действия* – дату, указанную в атрибуте считать датой окончания периода действия позиции.
Права доступа	Раздел, в котором определяются права пользователей на работу с данным атрибутом. Допускаемые разрешения: *Наследовать от родителя* (по умолчанию), *Нет доступа, Чтение, Создание, Редактирование*. Выбрать можно только один вариант.

3.14.8 Постобработчик для обогащения данных

Постобработчик для обогащения данных автоматически срабатывает при изменении значения атрибута / создании позиции или изменении статуса позиции, если это определено настройками или при ручном запуске по кнопке (вызывается в режиме редактирования из карточки позиции) (Рисунок 241, Рисунок 242). Назначение постобработчика производится при настройке деклараций. Он позволяет решать следующие задачи:

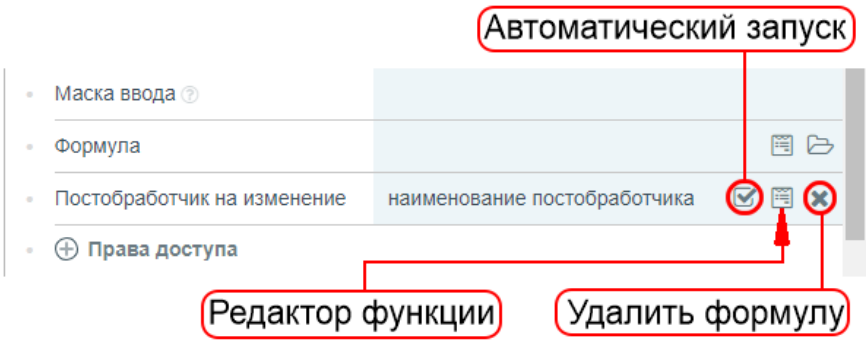


Рисунок 241 - Постобработчик в декларации атрибута

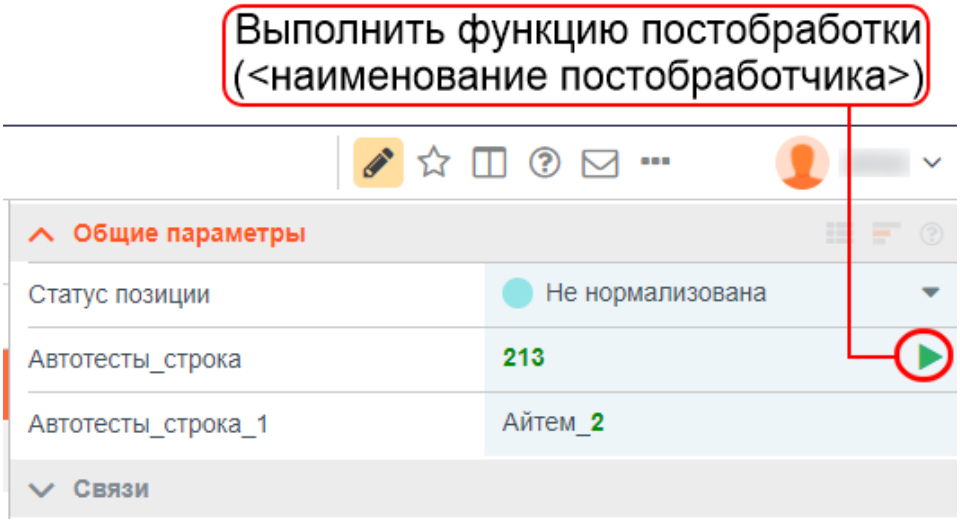


Рисунок 242 - Ручной вызов постобработчика по кнопке в карточке позиции

- Автоматическая генерация номенклатурного кода при создании новой позиции;
- Автоматическое заполнение атрибутов при создании позиции, в зависимости от текущего пользователя;
- Автоматическая генерация номенклатурного кода, после заполнения атрибута, с кодом номенклатурной группы и автоматическое изменение кода номенклатуры, при изменении кода номенклатурной группы;
- Автоматическое создание позиции во вспомогательном справочнике, по введенному значению в атрибуте типа «агрегация». Например, введен «ГОСТ 1050-88» в атрибут типа «агрегация», *Система* выполнила поиск, определила, что такой стандарт отсутствует, создала

его, заполнила нужный атрибут и подключила созданную позицию в атрибут агрегации;

- Автозаполнение атрибутов позиции, по введенному полному наименованию позиции и запуска пользователем постобработчика по кнопке;
- Автозаполнение атрибутов адреса, при вводе адреса одной строкой и запуска пользователем постобработчика по кнопке;
- Автозаполнение атрибутов контрагента, при вводе ИНН и запуска пользователем постобработчика по кнопке;
- Автоподбор классификационной группы по ОКПД2, ОКВЭД2, ЕКПС для позиции, после запуска постобработчика пользователем по кнопке для соответствующего атрибута классификации;
- Автозаполнение атрибутов , при вводе нового типоразмера пользователем и запуска постобработчика по кнопке. Например, вводится типоразмер листа «5х100х1000» и запускается постобработчик по кнопке, который разберет эту строку и заполнит атрибуты «Толщина», «Ширина», «Длина»;
- Массовая операция по преобразованию данных в позициях. Например, загружены позиции с не вычисляемым обозначением типоразмера. Далее, есть задача разобрать обозначение типоразмера по атрибутам, чтобы потом сделать атрибут «Обозначение типоразмера» вычисляемым. В этом случае, пишется постобработчик на атрибут *Обозначение типоразмера*, затем поиском и группой фильтруется таблица позиций и для столбца с атрибутом «Обозначение типоразмера» запускается постобработчик;

Автоматический запуск постобработчиков запрещен при выполнении:

- Импорта, через представление **Импорт и репликация**;
- Импорта из XML;
- Функции постобработчика – запрещено автоматическое срабатывание из одного постобработчика другого постобработчика, но при этом

остается возможность выполнить такой действие принудительно через вызов программной функции.

Постобработчик пишется в виде скрипта Groovy в специальном **Редакторе функций постобработки** (Рисунок 243), который вызывается кнопкой ***Редактор формул*** из ячейки **Постобработчик на изменение**.

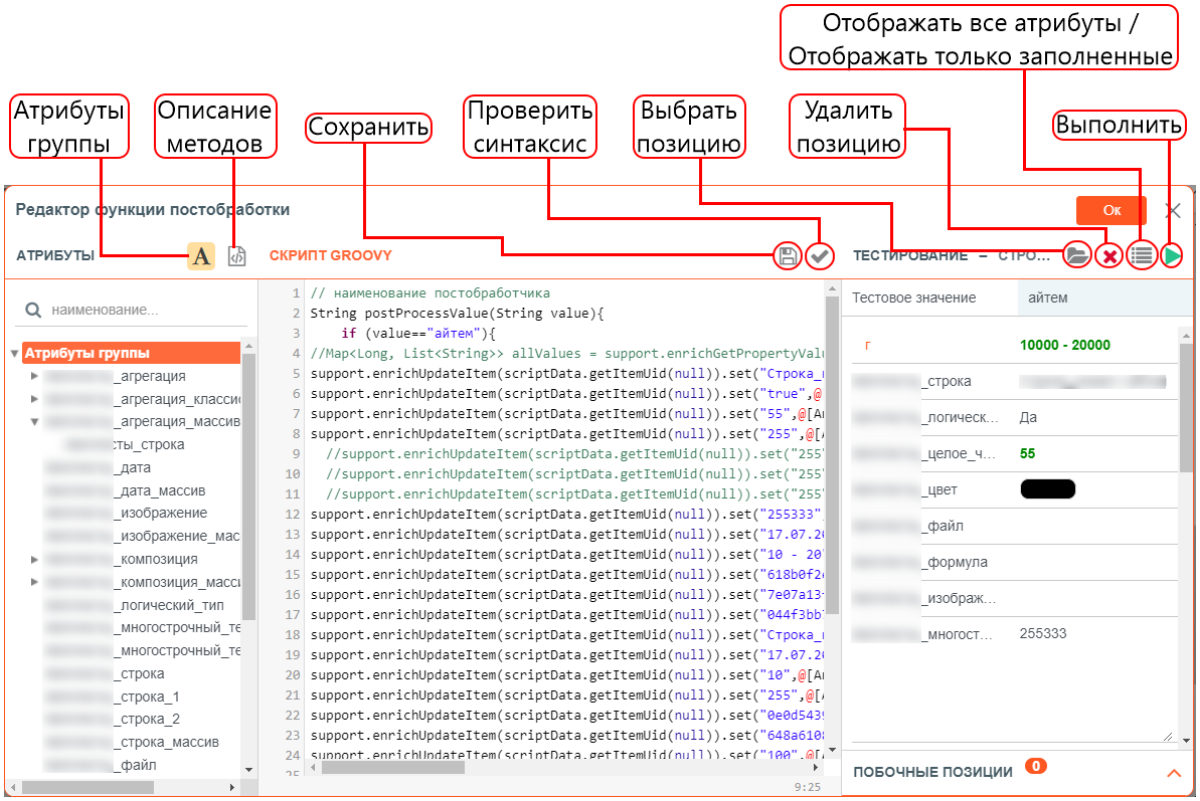


Рисунок 243 - Окно **Редактора функций постобработки**

В левой части окна редактора при активированной кнопке  ***Атрибуты*** располагается список доступных атрибутов текущей группы, а также атрибуты пользователя. Для того, чтобы включить атрибут в текст достаточно перетащить его (Drag&Dpop) из списка в поле скрипта в центре. Имя атрибута в тексте скрипта представляется как **@[Имя атрибута]**.

При переключении на описание методов  одноименной кнопкой пользователю становятся доступными все функции, методы, условия и циклы, применяемые в скриптах для обогащения данных, и их описания (во

всплывающих окнах - hint), которые появляются при наведении курсора на соответствующий элемент списка.

Для того, чтобы добавить нужную функцию в текст достаточно перетащить её (Drag&Dpop) из списка в поле скрипта. Функции добавляются со всем необходимым синтаксисом и пояснениями (комментариями). Если в тексте функции не достаточно данных или допущена ошибка, то в соответствующей строке появляется пиктограмма (Рисунок 244), при наведении на которую курсора, появляется подсказка с описанием допущенной ошибки.

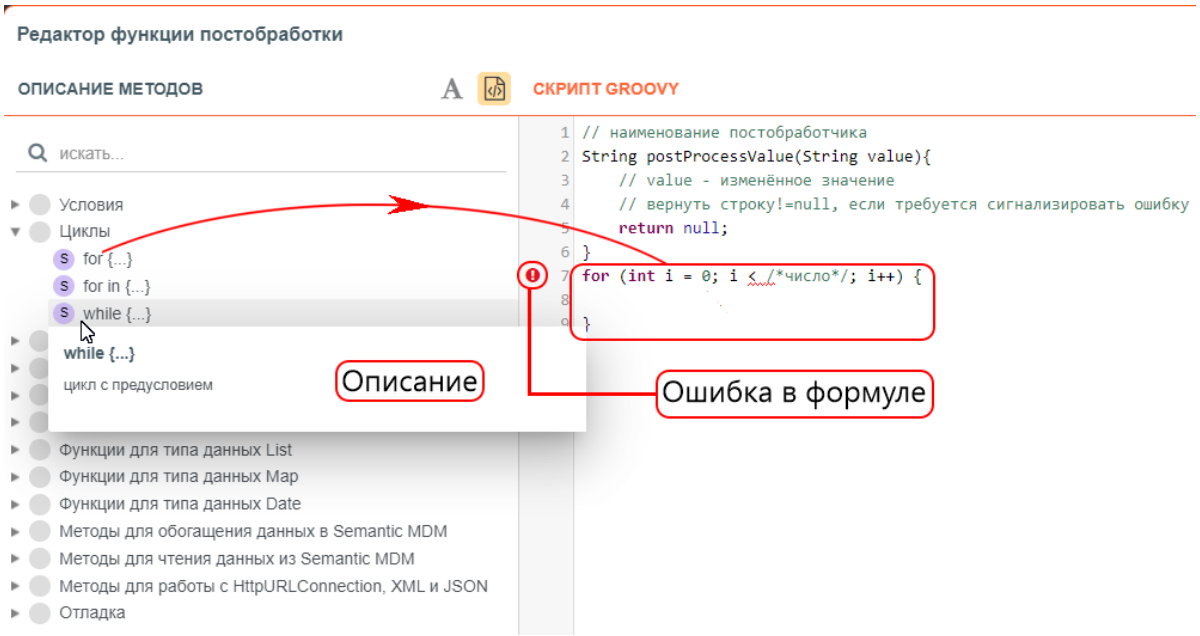


Рисунок 244 - Описание методов в скрипте GROOVY на обогащение данных (постобработчик на изменение)

Справа от текста скрипта находится поле для тестирования постобработчика, над которым расположены кнопки выбора позиции для тестирования, переключатель отображаемых атрибутов выбранной позиции и кнопка запуска процесса тестирования.

Редактор функций постобработки позволяет проводить тестирование разработанных скриптов, как на пустой новой позиции, так и на существующей. В *Системе* доступны методы, которые позволяют:

- читать/записывать значение из атрибута для позиций с заданным

UUID, с возможностью чтения атрибутов агрегированных и композированных позиций (включая множественные);

- создавать позиции в группе с заданным UUID;
- создавать композированные позиции для заданного родительской позиции (UUID) и атрибута композиции (UUID);
- удалять композированную позицию для заданной позиции (UUID) и атрибута композиции (UUID);
- выполнять поиск позиций в заданной группе по заданным атрибутам;
- работы с json/xml структурой через подключаемые библиотеки;
- работать с внешними веб-сервисами;
- вызывать программно другой постобработчик, назначенный на группу или декларацию атрибута;
- возвращать текст ошибки, для вывода в виде всплывающего сообщения;
- записать значение атрибута в базовой ЕИ для позиций с заданным UUID и идентификатором атрибута;
- записать ЕИ и значение в этой же ЕИ для позиций с заданным UUID и идентификатором атрибута.
- Редактор обеспечивает следующие возможности:
- Выбор функции / переменной / константы из выпадающего списка по нажатию <*Ctrl*>+<*Пробел*>;
- Выполнение поиска по коду с возможностью использования регулярных выражений по нажатию <*Ctrl*>+<*F*>;
- Выполнение быстрого комментирования / раскомментирования строк / группы строк по нажатию <*Ctrl*>+</>;
- Удаление строки целиком по <*Ctrl*>+<*D*>;
- Автоматическая установка правильного отступа, при переходе на новую строку;
- Подсвечивание парных скобок;
- Подсвечивание ошибок в редакторе скрипта;

- Промежуточное сохранение скрипта, по нажатию на кнопку ***Сохранить*** над областью скрипта;
- Проверка синтаксиса скрипта по нажатию на кнопку ***Проверить синтаксис*** (в правом верхнем углу над областью скрипта) .

Для проверки функций обогащения на панели справа отображается карточка проверяемой позиции в режиме «только чтение» (то есть пользователь не может изменить значения в карточке вручную). В карточке тестируемой позиции могут отображать все атрибуты, с возможностью переключения в режим отображения атрибутов со значениями.

По умолчанию, для тестирования функции, в карточке отображается пустая «виртуальная» позиция. В заголовке карточки доступны кнопки ***Выбрать позицию***, ***Удалить позицию***. По кнопке «Выбрать позицию» поднимается окно выбора позиции из справочника, в котором продекларирован атрибут с постобработчиком. Над карточкой располагается поле для ввода тестового значения атрибута и кнопка запуска функции ***Выполнить***. По нажатию на данную кнопку, система выполняет функцию постобработчика и обогащает выбранную позицию данными. Пользователь видит в карточке все изменения в позиции, с учетом изменений в агрегированных и композированных в неё позиций. При повторном запуске функции, состояние атрибутов позиции будет сброшено в исходное и выполнится повторное обогащение позиции данными.

Если после выполнения тестирования функции, появились позиции, никак не связанные с тестируемой, то перечень таких позиций отображается списком в разделе **Побочные позиции**, под карточкой атрибутов с пояснением что данные позиции были созданы функцией постобработчика, но никак не связаны с тестируемой позицией.

Постобработчик в дочерних группах отмечается пиктограммой как унаследованный, если он задан в родительской группе (Рисунок 245).

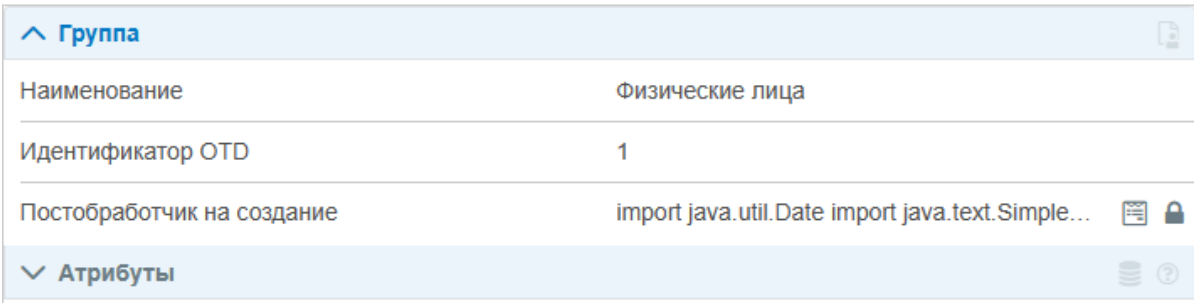


Рисунок 245 - Постобработчик, унаследованный от родительской группы

3.14.9 Настройка прав доступа на атрибуты и группы

На каждый декларируемый в группе атрибут, а также на саму справочную группу назначаются права доступа, которые определяют порядок использования справочных данных.

Настройку прав доступа может осуществлять пользователь с привилегией *Управление безопасностью,* а также пользователь с привилегией *Чтение безопасности*, но только в тех справочных группах, для которых ему установлен уровень *Полный доступ*.

Уровень доступа в декларации атрибута для пользователя и / или роли по умолчанию берется из настроек прав доступа справочной группы, в которой он продекларирован. В случае необходимости, унаследованные права доступа для пользователя и / или роли могут быть перекрыты на уровне декларации.

Для настройки прав доступа к декларации атрибута в справочной группе необходимо:

- установить курсор на нужном атрибуте в списке и перейти в раздел **Права доступа** (Рисунок 246) (включать режим управления пакетами изменений не требуется);

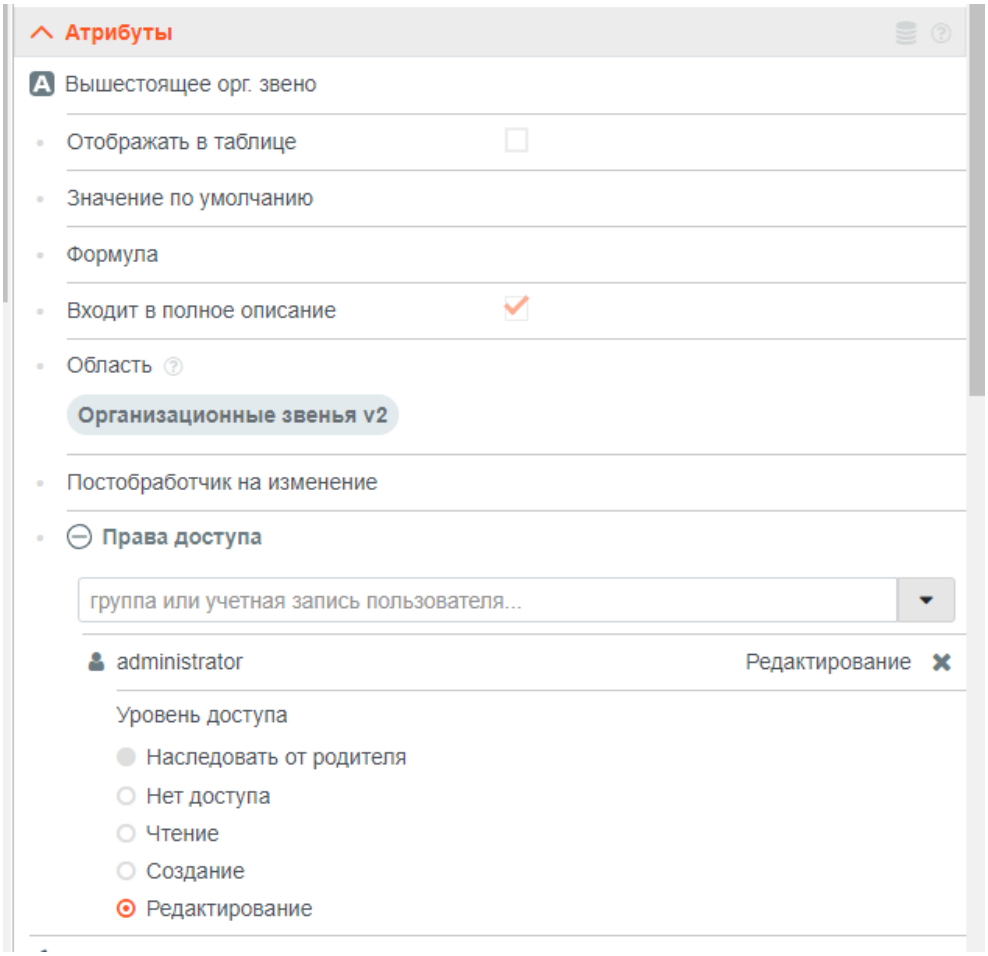


Рисунок 246 - Настройка прав доступа к атрибуту

- в строке выбора группы или учётной записи пользователя выбрать из выпадающего списка нужную группу или пользователя;
- установить маркер напротив требуемого разрешения:
  - *Наследовать от родителя* – права на атрибут наследуются из декларации атрибута родительской справочной группы.
  - *Нет доступа* – пользователи не могут видеть атрибут и его значения в данной справочной группе.
  - *Чтение* – пользователи могут видеть атрибут и его значения в данной справочной группе.
  - *Создание* – пользователи имеют право *Чтение* и могут редактировать значения атрибута в только что созданных позициях в текущем пакете изменений.
  - *Редактирование* – пользователи имеют право *Создание* и могут

редактировать значения атрибута для любых позиций в данной справочной группе.

- повторить добавление групп или пользователей нужное количество раз.

Настройка прав на справочную группу выполняется аналогичным способом в разделе **Права доступа**. При назначении прав доступа на справочную группу, *Система* автоматически использует их для определения уровня доступа к каждому из атрибутов, если для атрибута не назначены иные права доступа. Например, при назначении на справочную группу уровня доступа *Редактирование*, пользователь автоматически получает уровень доступа *Редактирование* и на все атрибуты этой группы, если для них не настроен иной уровень доступа.

Для настройки прав на справочную группу (Рисунок 247) следует:

- в карточке группы развернуть раздел **Права доступа**;
- в строке выбора группы или учётной записи пользователя выбрать из выпадающего списка нужную группу или пользователя;

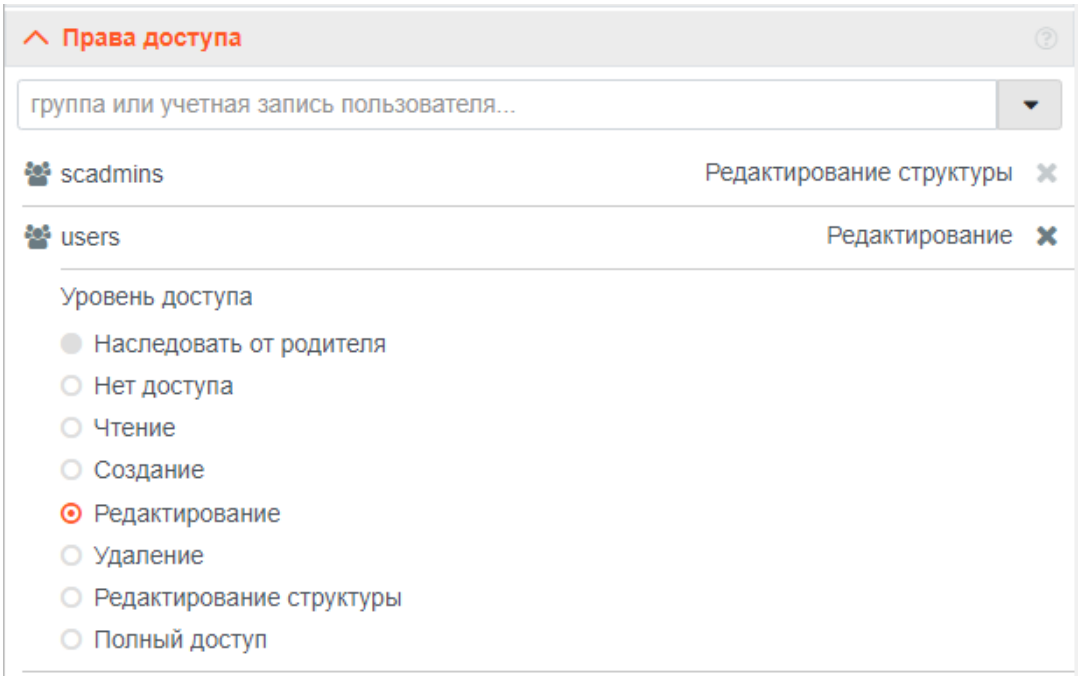


Рисунок 247 - Настройка прав доступа к справочной группе

- установить маркер напротив требуемого разрешения:
  - *Наследовать от родителя* – права на текущую справочную группу наследуются из родительской справочной группы;
  - *Нет доступа* – пользователи не могут видеть данную справочную группу;
  - *Чтение* – пользователи могут видеть данную справочную группу и все ее позиции;
  - *Создание* – пользователи имеют право *Чтение* и могут создавать новые позиции в данной справочной группе;
  - *Редактирование* – пользователи имеют право *Создание* и могут редактировать любые позиции в данной справочной группе, но не могут удалять ранее созданные позиции. Удаление позиций, созданных в текущем пакете изменений разрешено;
  - *Удаление* – пользователи имеют право *Редактирование* и могут удалять любые позиции в данной справочной группе;
  - *Редактирование структуры* – пользователи имеют право *Удаление* и могут редактировать в данной справочной группе:
    - список подчиненных справочных групп,
    - список деклараций атрибутов,
    - список ограничений,
    - настраивать совместимости,
    - настраивать заявки,
    - настраивать поиск дубликатов;
  - *Полный доступ* – пользователи имеют право *Редактирование структуры* и могут назначать права доступа другим ролям и пользователям.
- повторить добавление групп или пользователей нужное количество раз.

Если на уровне справочной группы для пользователя установлено право на чтение, а на какой-либо атрибут этой группы назначены права на

редактирование и запись, то этот атрибут он сможет редактировать, а все остальные видеть только в режиме просмотра.

Права доступа для справочных групп и деклараций атрибутов также могут быть настроены с использованием Матрицы доступа (см. п. 3.28.7), расположенной в подсистеме **Администрирование**.

3.14.10Настройка ограничений

Для проверки правильности и целостности вводимой в *Систему* информации предусмотрены различные способы контроля. Эти способы описываются в карточке группы любого справочника, кроме классификаторов, в разделе **Ограничения**.

В Системе применяются четыре вида ограничений (Рисунок 248):

- Ограничительные таблицы;
- Обязательные атрибуты;
- Уникальные атрибуты;
- Функции проверки данных.

Ограничительные таблицы – специальные таблицы, которые устанавливают зависимость значений одних атрибутов от значений других. Например, зависимость разрешённых значений диаметров проката от применяемого ГОСТа или ТУ. Могут использоваться не только для проверки вводимых данных (имеют пиктограмму), но также для генерации новых позиций (имеют пиктограмму). Позиции, несоответствующие ограничительным таблицам, не могут быть опубликованы, если им установлен статус *Нормализована*.

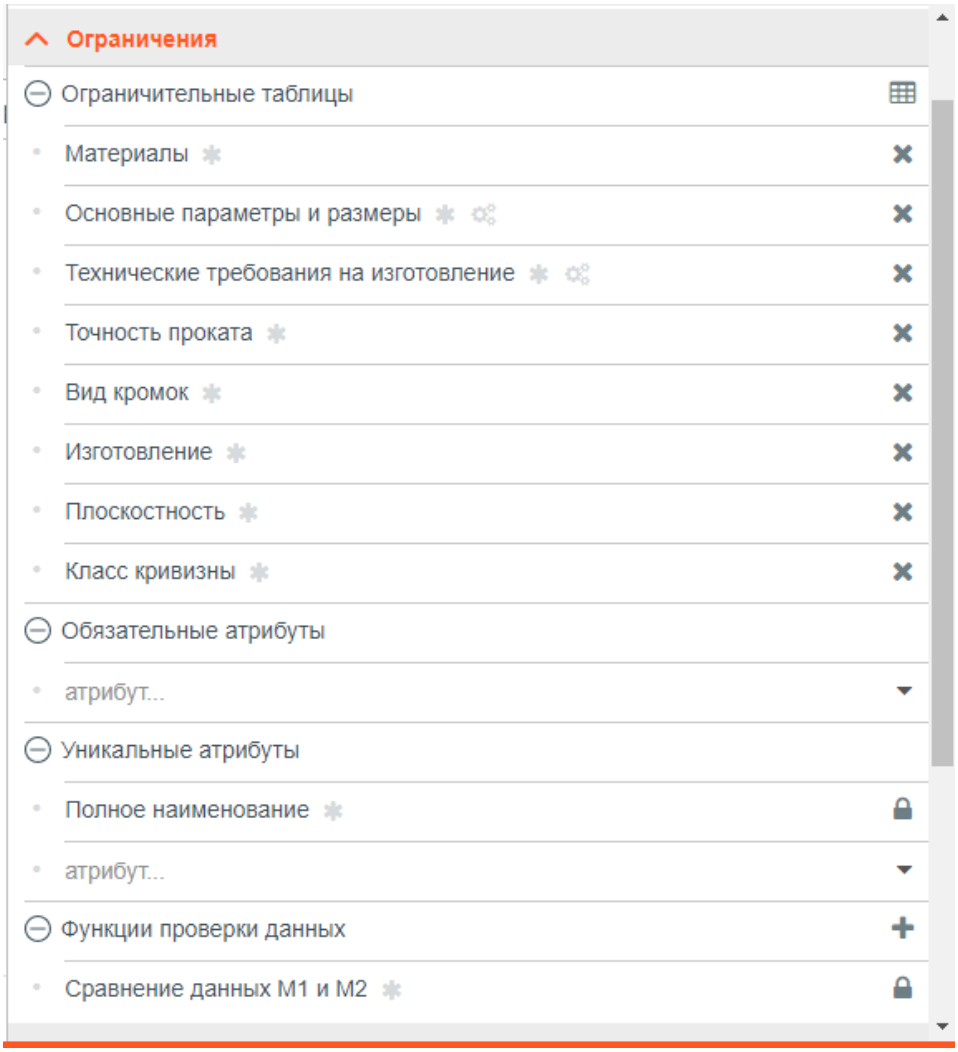


Рисунок 248 - Ограничения, применяемые в справочной группе

*Обязательные атрибуты (*Рисунок 249*)* – если в группе не применяются ограничительные таблицы, рекомендуется указать перечень обязательных атрибутов, без ввода значений которых, запись не может быть опубликована со статусом *Нормализована*. Обязательные атрибуты выбираются из выпадающего списка из числа атрибутов, ранее добавленных в текущую группу.

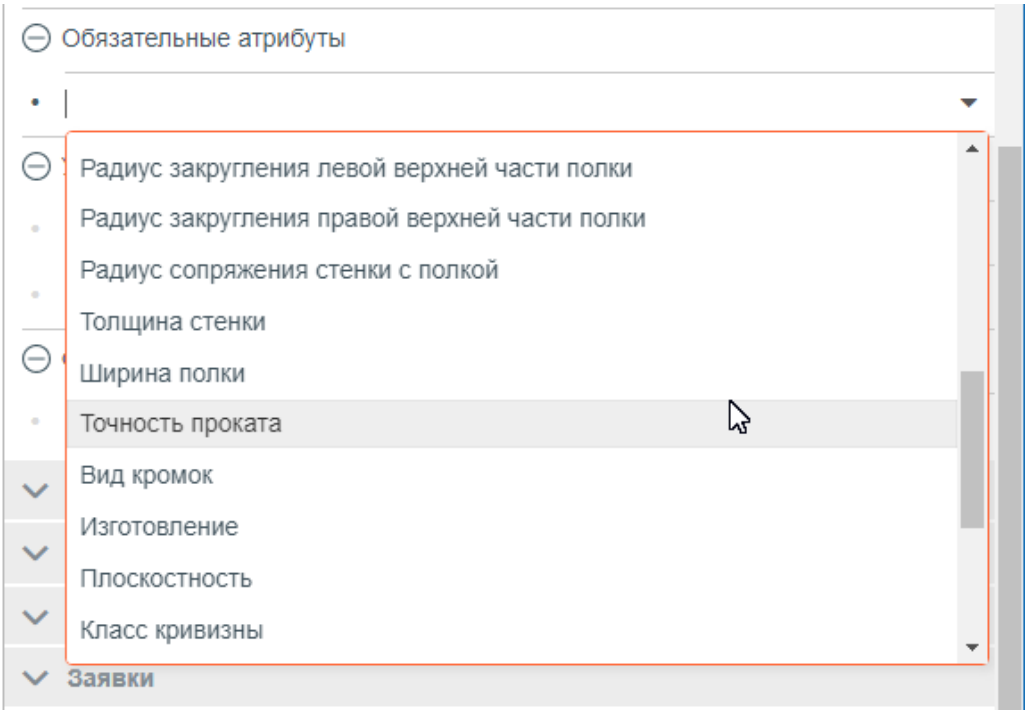


Рисунок 249 - Выбор атрибутов в качестве обязательных

Уникальные атрибуты (Рисунок 250) – это атрибуты, значения которых автоматически проверяются на уникальность в пределах справочной группы. В качестве уникальных атрибутов могут выступать наименования, шифры, коды и т.п. Можно накладывать ограничение уникальности сразу на несколько атрибутов, что позволяет проверять уникальность сочетания значений различных атрибутов. Как и в случае с обязательными атрибутами, они выбираются из выпадающего списка и могут быть унаследованы от родительского к дочернему объекту, то есть атрибут, назначенный уникальным на уровне справочника, будет таковым для всех его веток, если настройкой не определено иное.

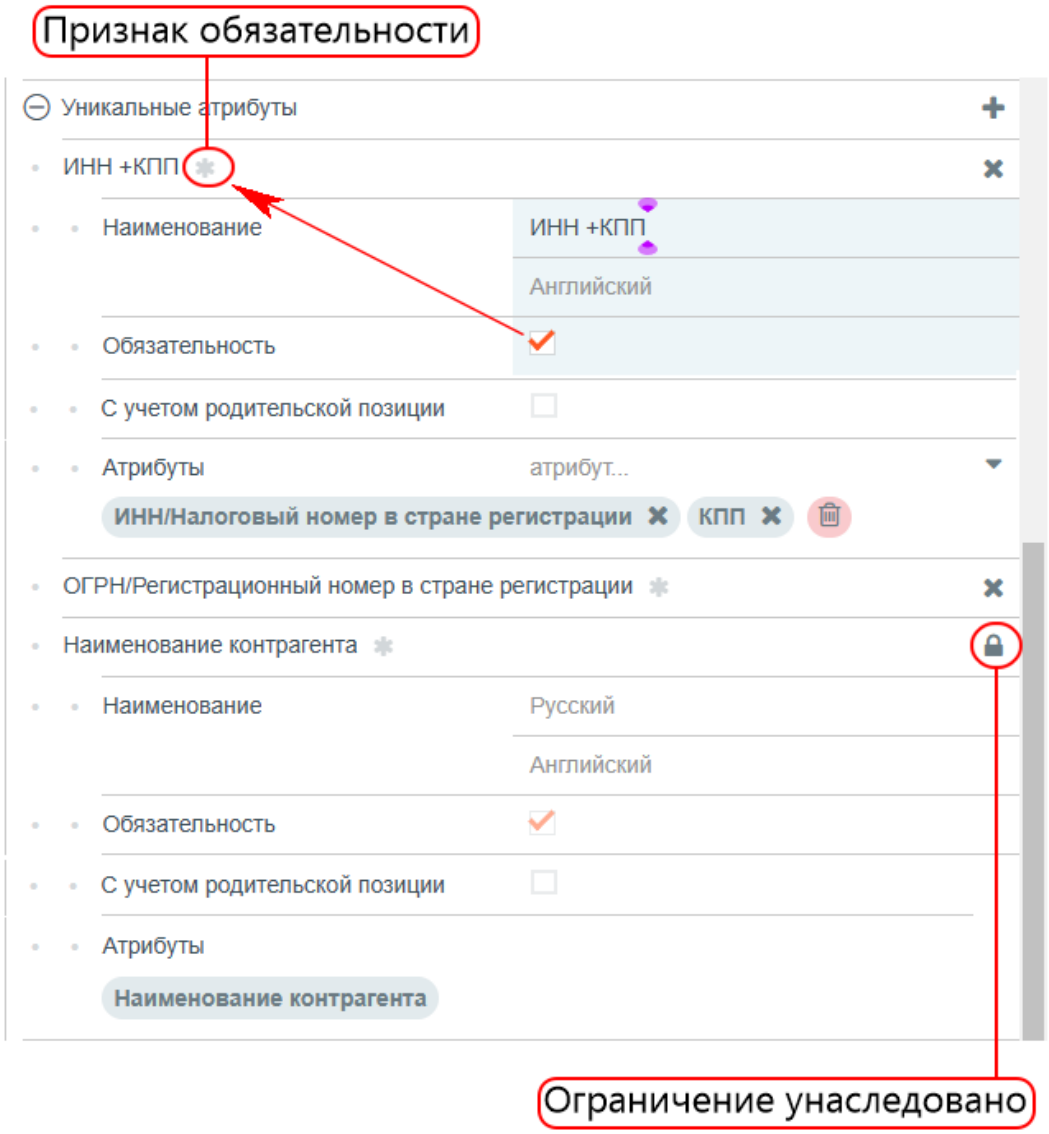


Рисунок 250 - Настройка уникальных атрибутов

Уникальному атрибуту дополнительно может быть присвоена опция *Обязательность*, которая устанавливается маркером в соответствующей ячейке.

Опция *С учетом родительской позиции* обеспечивает проверку уникальности значений атрибутов композируемых позиций в рамках композирующей (родительской) позиции. Например, проверка расчетных счетов в рамках контрагента, которому они принадлежат.

Функции проверки данных (Рисунок 251)– это специальный скрипт Groovy, который обеспечивает проверку вводимых данных по какому-либо

алгоритму. Например, проверку ИНН на контрольное число. Доступ к скрипту осуществляется через строку **Формула**.

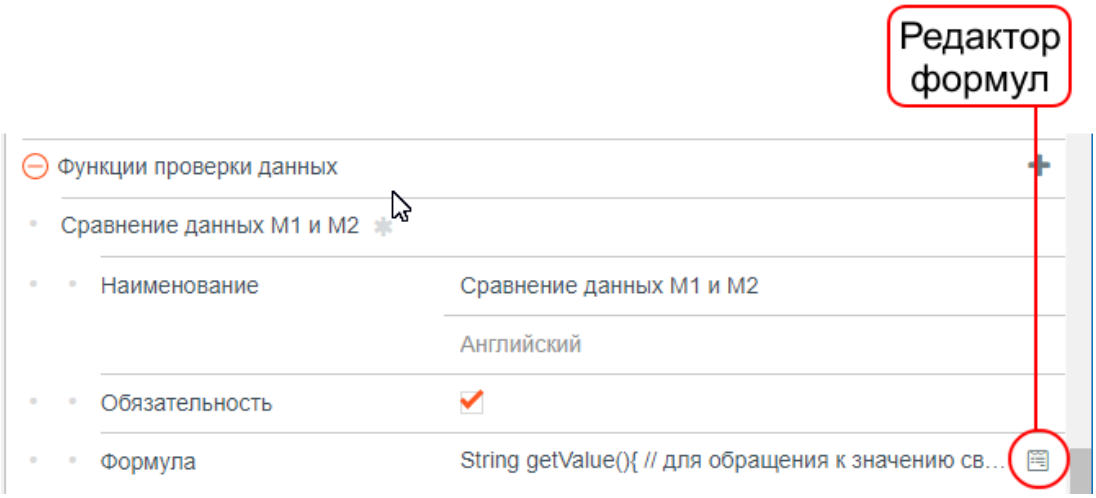


Рисунок 251 - Настройка функции проверки данных

Скрипт создаётся в специальном редакторе формул (Рисунок 252) где, используя ссылки на атрибуты справочника, записывается последовательность и процедуры проверки данных. Ссылка на атрибут формируется путём перетаскивания (Drag&Drop) наименования атрибута из списка в левой части в поле скрипта. При переключении в режим **Описание методов** пользователю становятся доступны все методы, функции, условия и циклы для разработки скрипта. Все элементы добавляются в скрипт путём перетаскивания (Drag&Drop) и содержат необходимый синтаксис и комментарии.

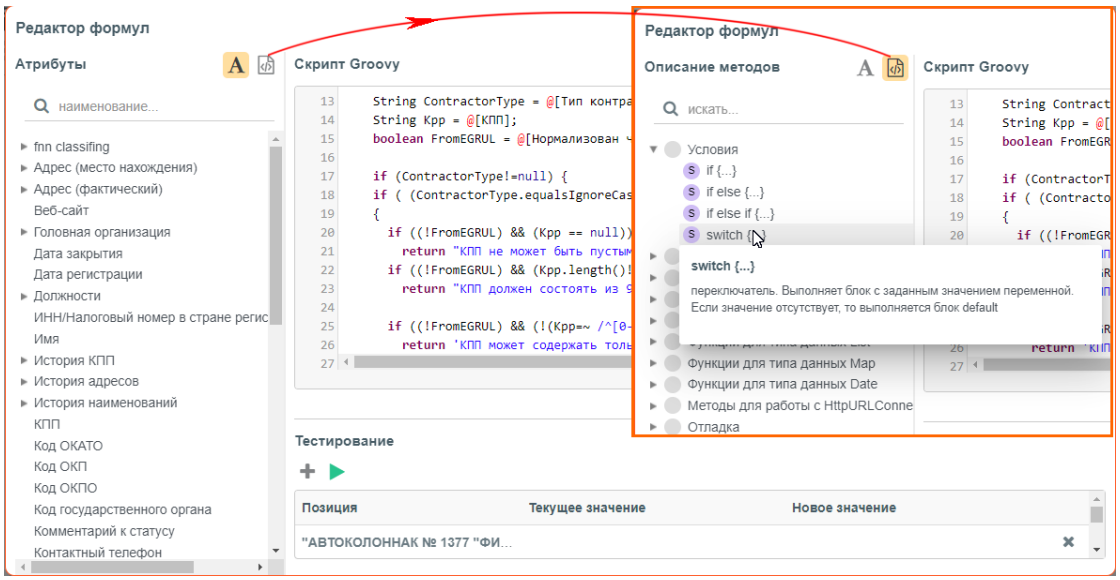


Рисунок 252 - Редактор формул для скрипта Groovy

Для работы с текстом скрипта пользователю доступны следующие функции:

- Просмотр описания метода и его применение в тексте скрипта;
- Выбор функции, переменной или константы из выпадающего списка по нажатию <*Ctrl*>+<*Пробел*>;
- Поиск по коду с возможностью использования регулярных выражений по нажатию <*Ctrl*>+<*F*>;
- Быстрое комментирование / раскоментирование строк или группы строк по нажатию <*Ctrl*>+</>;
- Удаление строки целиком по <*Ctrl*>+<*D*>;
- Автоматическая установка правильного отступа, при переходе на новую строку.

В редакторе может быть выполнено тестирование созданного скрипта. Для этого следует:

- в разделе Тестирование нажать кнопку *Добавить позицию* (Рисунок 253);

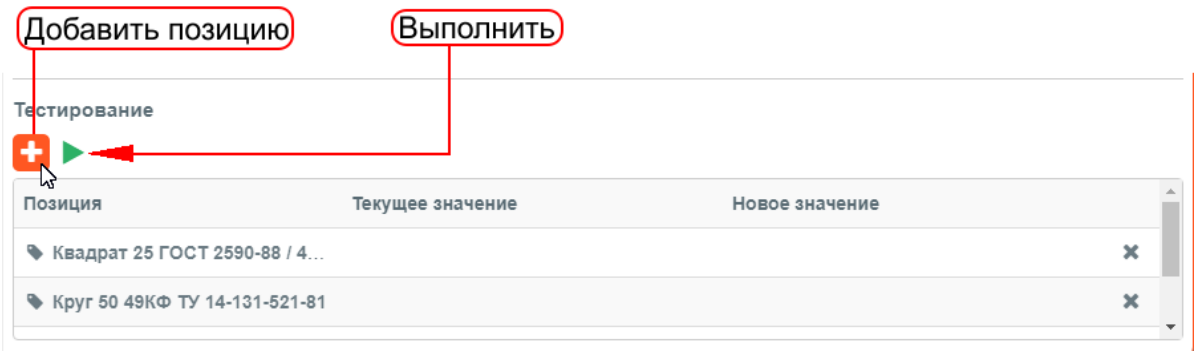


Рисунок 253 - Тестирование скрипта

- в открывшемся справочнике выбрать нужную запись (позиции должны быть заранее созданы в тестируемом справочнике), установить на неё курсор и нажать кнопку ***Выбрать*** (в правом нижнем углу);
- в редакторе нажать кнопку ***Выполнить***.

Результаты тестирования будут представлены в таблице в колонках **Текущее значение** и **Новое значение**.

В процессе тестирования на консоль отладки выводятся отладочные сообщения. Для каждого тестируемого элемента должна быть своя консоль, которая должна открываться по специальной кнопке, расположенной рядом с кнопкой просмотра результата выполнения скрипта над тестируемым объектом.

3.14.11Создание новой позиции в таблице данных

Для того, чтобы добавить позицию в справочную группу (кроме позиций, создаваемых по ограничительным таблицам) необходимо:

- перейти в режим редактирования создав новый пакет изменений или активизировав существующий пакет;
- установить курсор на нужной справочной группе;
- на сетке данных нажать кнопку ***Добавить позицию*** или выбрать аналогичную команду из контекстного меню (Рисунок 254);
- перейти в карточку позиции и заполнить предлагаемые атрибуты нужными значениями;

- повторить процедуру добавления позиций нужное число раз. При необходимости ввода многих одинаковых значений каких-либо атрибутов следует воспользоваться функцией множественного редактирования.

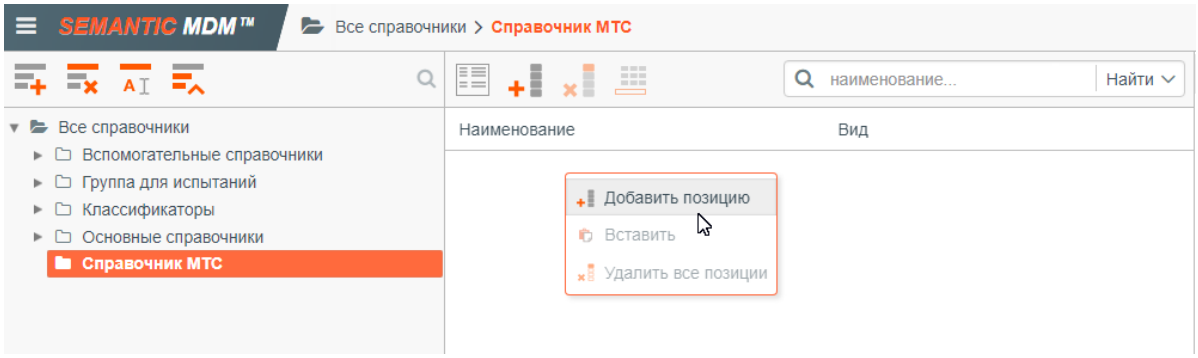


Рисунок 254 - Добавление новой позиции в таблицу данных

3.14.12Ввод значений с плавающей точкой

Для значений, относящихся к типу «число с плавающей точкой», разрешены три способа указания (Рисунок 255):

- *значение* – указывается только номинальное значение с точностью, определённой при настройке атрибута (см. п. 3.16.7);
- *значение (верхнее отклонение; нижнее отклонение)* – указывается номинальное значение, затем без пробела вводится символ открывающей скобки «(», указывается верхнее отклонение (знак «+» не применяется), ставится точка с запятой «;» и затем нижнее отклонение. Закрывающая скобка «)» ставится автоматически;
- *минимальное – максимальное* – вводится минимальное значение затем нажимается клавиша <Пробел> и вводится максимальное значение.

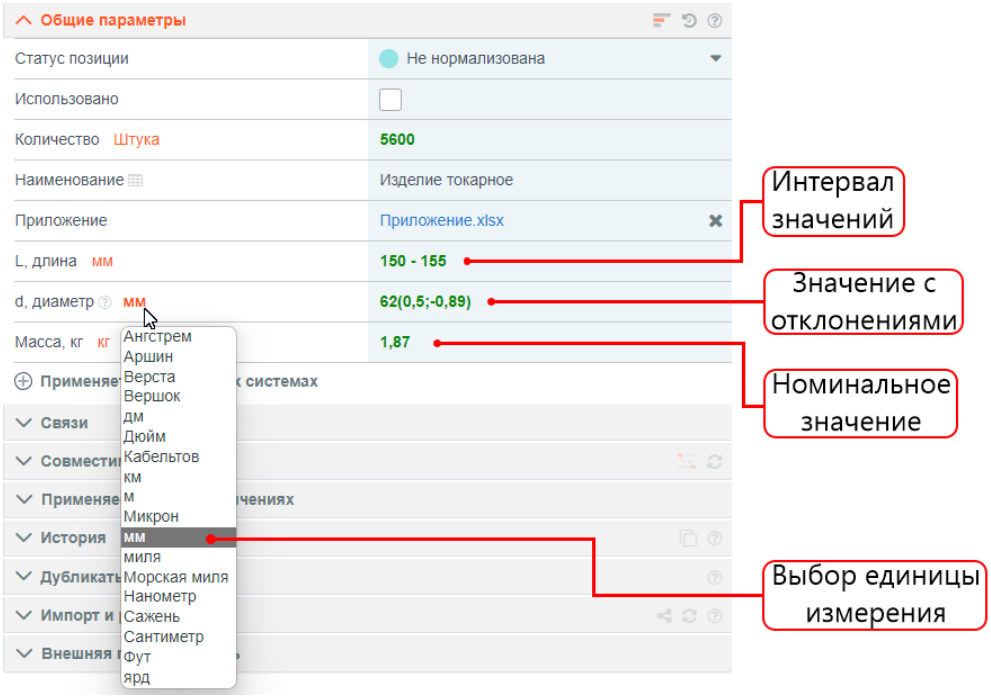


Рисунок 255 - Варианты ввода числа с плавающей точкой

Единица измерения может быть изменена непосредственно в карточке позиции. Для этого нужно щелкнуть левой кнопкой мыши на обозначении ЕИ рядом с наименованием и выбрать требуемое из выпадающего списка.

3.14.13Назначение цвета позиции

Если у позиции в списке характеристик присутствует атрибут типа «цвет», то его выбор производится из выпадающей цветовой палитры путём выбора нужной области цветовой шкалы щелчком левой кнопки мыши с последующим указанием мышью нужного цвета на палитре (Рисунок 256).

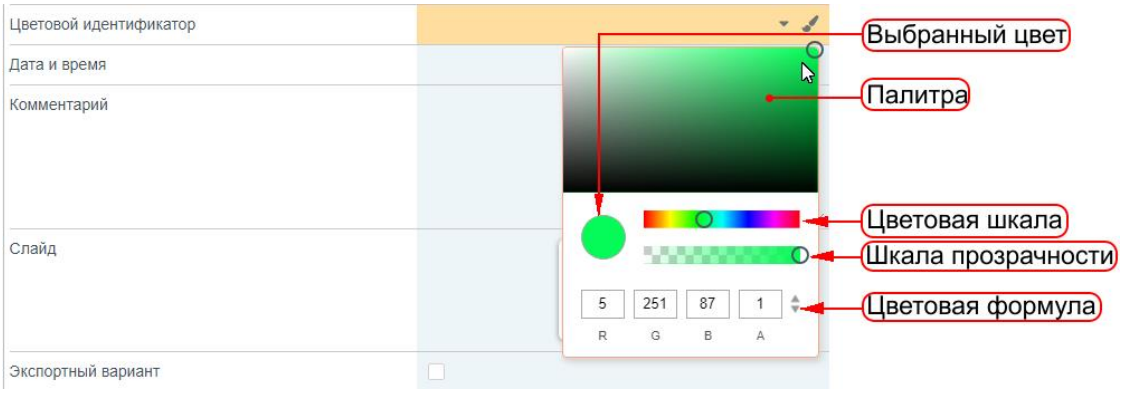


Рисунок 256 - Настройка цвета для позиции

Вторым вариантом является указание цветовой формулы в формате RGBa (где R – красный цвет, G – зелёный цвет, B – синий цвет (0 – нет цвета, 255 – чистый цвет), A – прозрачность (0- прозрачный, 100 – непрозрачный)).

Атрибут цвета может быть выведен в таблицу данных и использоваться в качестве идентификатора.

Редактирование цвета производится аналогичным способом.

3.14.14Добавление и редактирование изображения объекта

Если для объекта (позиции) разрешено добавление изображения (или нескольких изображений, что зависит от настроек атрибута при его создании), то для работы с ним в ячейке присутствуют специальные кнопки рядом с областью рисунка (Рисунок 257):

- ***Загрузить изображение*** – вызывает *Проводник* для выбора файла с изображением. Изображения могут быть в форматах BMP, PNG, JPEG размером не более 5 Мб;



Рисунок 257 - Добавление изображения к описанию позиции

- ***Сделать изображением родительской группы*** – присваивает текущее изображении родительской группе. Кнопка доступна, если изображение уже загружено;
- ***Удалить изображение*** – удаляет изображение из справочника. Кнопка доступна, если изображение уже загружено.

3.14.15 Указание даты и времени

Если в описании позиции используется атрибут типа «Дата и время», то для его заполнения следует нажать на кнопку календаря в правой части ячейки и установить требуемые значения. Выбор месяца и года выполняется в соответствующих ячейках. Установка времени возможна только с использованием кнопок «^» и «V», расположенных над часами и минутами (Рисунок 258).

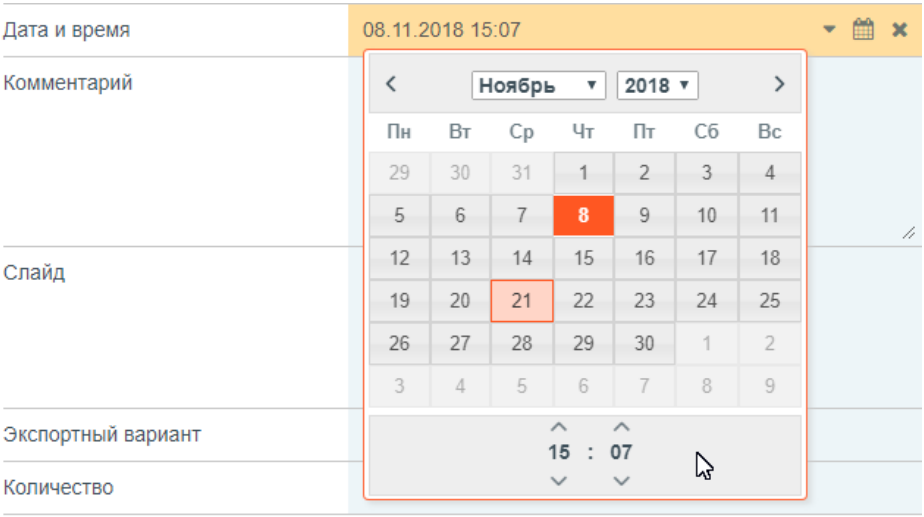


Рисунок 258 - Настройка даты и времени

Если значения даты и времени для других позиций уже были заполнены ранее, то они будут доступны для выбора в выпадающем списке значений атрибута.

3.14.16 Перемещение и удаление позиций

Для того, чтобы удалить позицию(ии) следует (Рисунок 259):

- перейти в режим редактирования, создав новый пакет изменений или активизировав существующий пакет;
- установить курсор на нужной позиции в таблице данных;
- нажать кнопку ***Удалить*** на панели инструментов таблицы или выбрать из контекстного меню команду **Удалить** или команду **Удалить всё**.

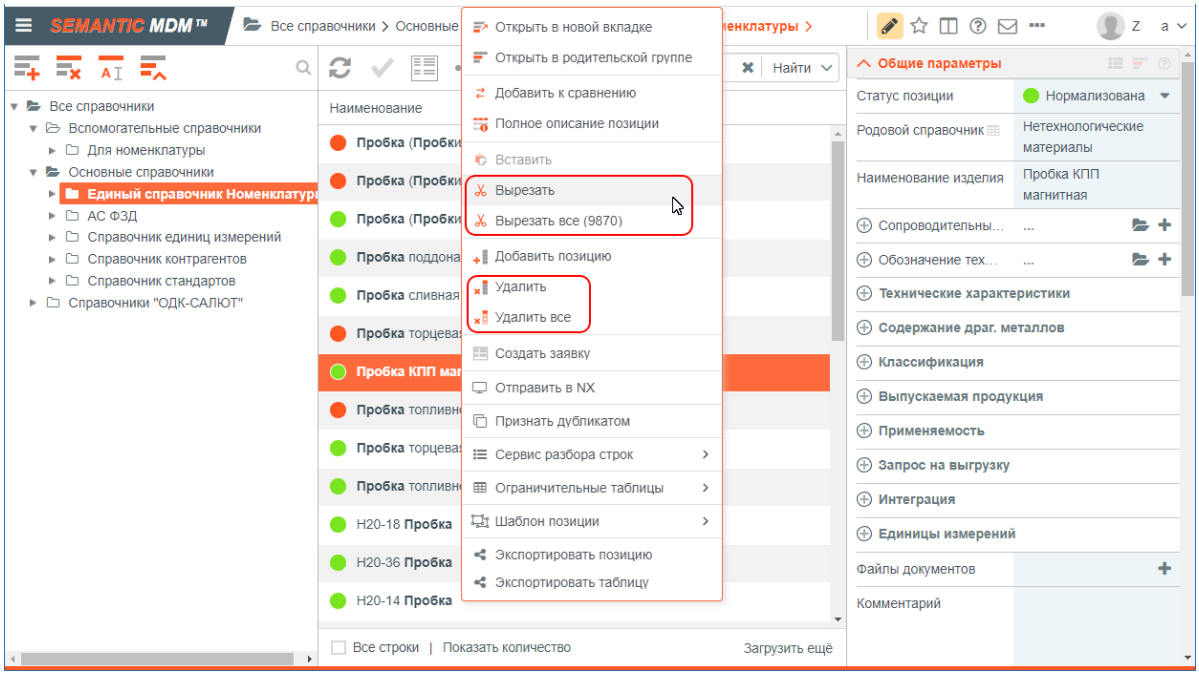


Рисунок 259 - Команды контекстного меню для перемещения и удаления позиций

Для того, чтобы переместить позицию(ии) из одной справочной группы в другую следует:

- перейти в режим редактирования, создав новый пакет изменений или активизировав существующий пакет;
- установить курсор на нужной позиции в таблице данных;
- выбрать из контекстного меню команду **Вырезать** или команду **Вырезать всё**. Рядом с командой **Вырезать все** в скобках указывается количество позиций, к которым будет применена команда;
- установить курсор на справочной группе, в которую перемещается объект / объекты;
- в области таблицы данных вызвать из контекстного меню команду **Вставить** (Рисунок 260).

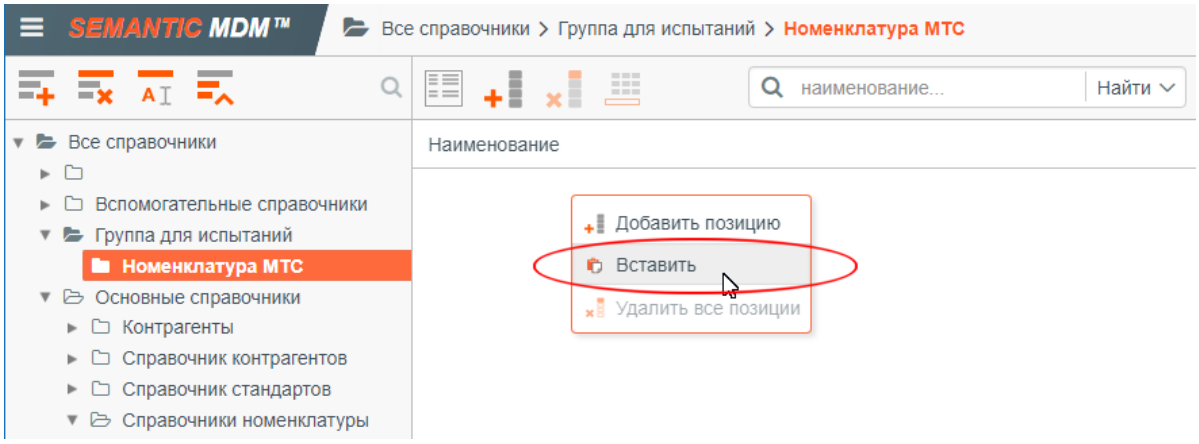


Рисунок 260 - Перемещение справочного объекта

Если в справочной группе, в которую производится перемещение позиции, отсутствуют настроенные атрибуты или они не совпадают (полностью или частично) с атрибутами перемещаемого объекта, пользователю будет предложено выбрать необходимые атрибуты в окне **Незадекларированные свойства** (Рисунок 261) и затем нажать кнопку ***Вставить***. После этого позиция будет перемещена, а её атрибуты (выбранные при настройке) будут присвоены всей группе.

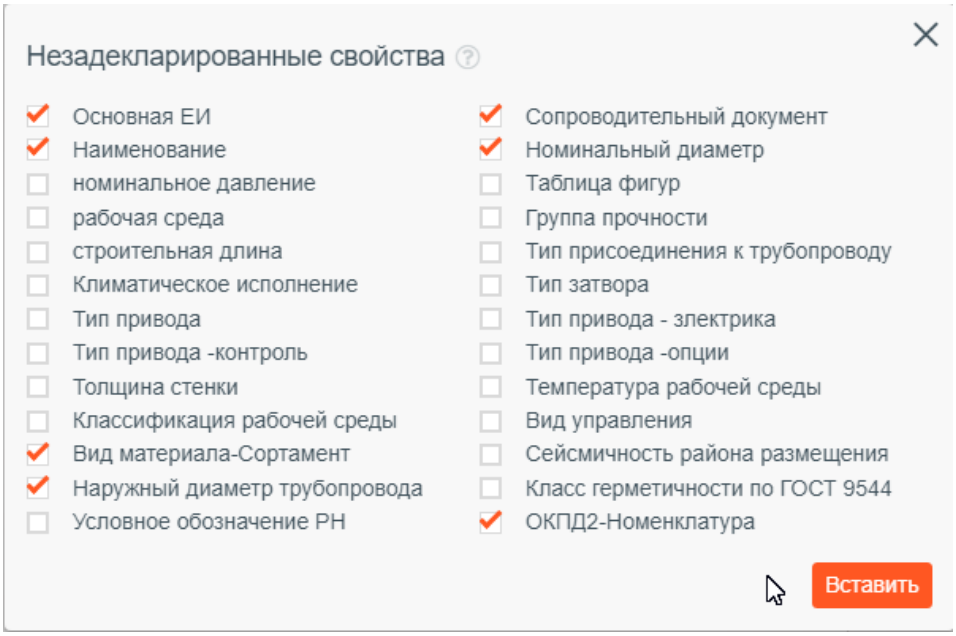


Рисунок 261 - Окно **Незадекларированные свойства** для выбора атрибутов перемещаемой позиции

Перемещение также доступно для объектов, являющихся композицией (Композиция – это связь типа «родитель» – «дочерний объект», причём, если удаляется родительский объект, то автоматически удаляются и все дочерние). Композированная позиция может быть перемещена от одной родительской позиции к другой, либо вставлена в группу как самостоятельная позиция с сохранением идентификатора. Например, вырезать композированный в юридическое лицо филиал и вставить его в другое юридическое лицо, либо сделать его самостоятельным юридическим лицом (если он был по ошибке введён как филиал).

3.14.17 Мастер создания позиции

Для создания позиции в справочнике номенклатуры присутствует мастер создания номенклатурной позиции, который позволяет по ограниченным сведениям из введённой строки:

- автоматически определить группу классификатора, в которой должна быть создана позиция;
- автоматически заполнить идентифицирующие атрибуты;
- выполнить поиск существующей позиции по идентифицирующим атрибутам.

Использование мастера значительно увеличивает скорость работы эксперта НСИ.

Для того, чтобы воспользоваться мастером необходимо:

- перейти в ветку справочников номенклатуры;
- включить режим управления пакетами изменений;

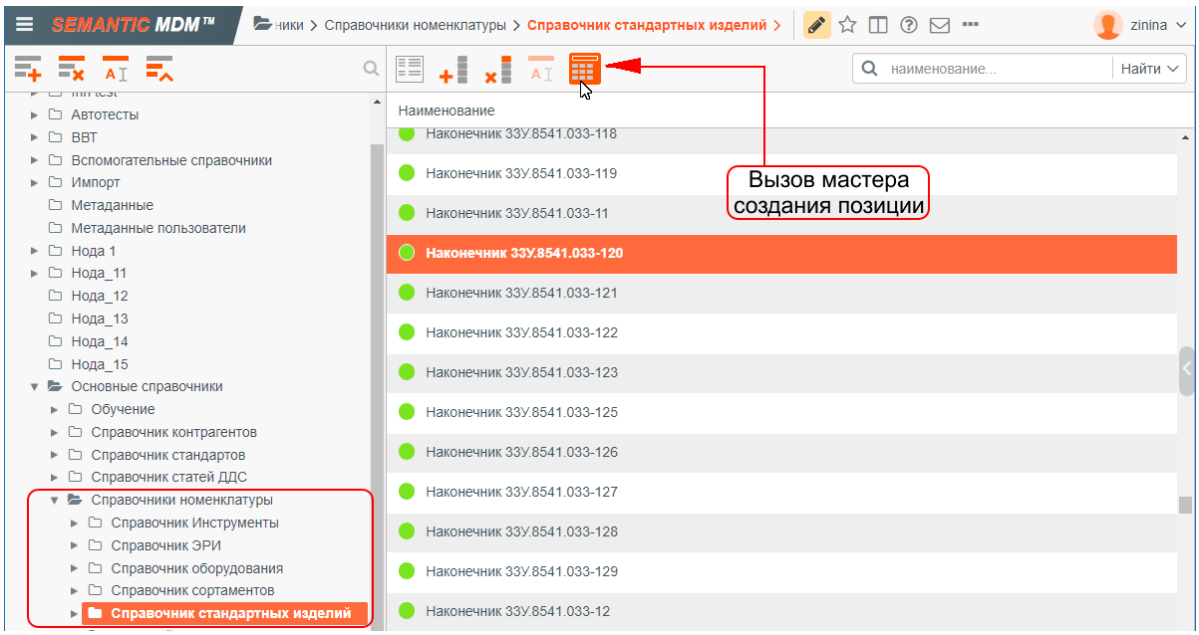


Рисунок 262 - Вызов мастера создания позиции в справочнике номенклатуры

- установить курсор в таблице данных и нажать на кнопку ***Генерация позиций по наименованию***[17] (Рисунок 262);
- в открывшемся окне **Мастер создания позиции номенклатуры** (Рисунок 263) в разделе **Распознавание значений** ввести в ячейку известные атрибуты (наименование, обозначение, типоразмеры) и нажать кнопку ***Распознать***;
- найденные атрибуты аналогичных записей будут расположены ниже ячейки с запросом. Предлагаемые значения атрибутов могут быть при необходимости отредактированы здесь же. Справа в таблице будут расположены похожие позиции, если выбрать одну из них, то будет показана карточка объекта;

[17] *Кнопка* ***Генерация позиций по наименованию*** *доступна только при наличии работающей подсистемы «Машинное обучение», которая обеспечивает данный функционал за счет применения заранее обученного искусственного интеллекта*

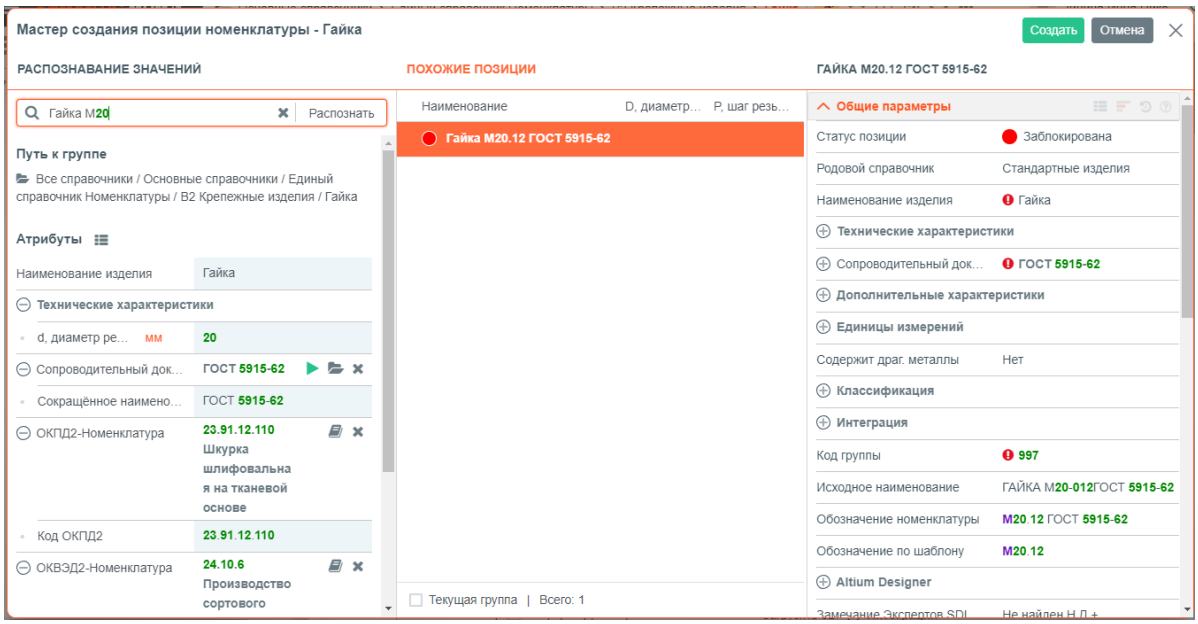


Рисунок 263 - Окно **Мастер создания позиции номенклатуры**

- после редактирования атрибутов следует нажать кнопку ***Создать***, после чего позиция будет сгенерирована и внесена в справочник.

3.14.18 Настройки справочных групп для работы с заявками

Для каждого справочника и справочной группы могут быть выполнены настройки, обеспечивающие работу с заявками пользователей на добавление, изменение или удаление позиций.

Настройка работы с заявками со стороны справочника производится в разделе **Заявки** в карточке группы (Рисунок 264). Последовательность и правила настройки изложены в п. 3.12.18, 3.12.21 и 3.12.25.

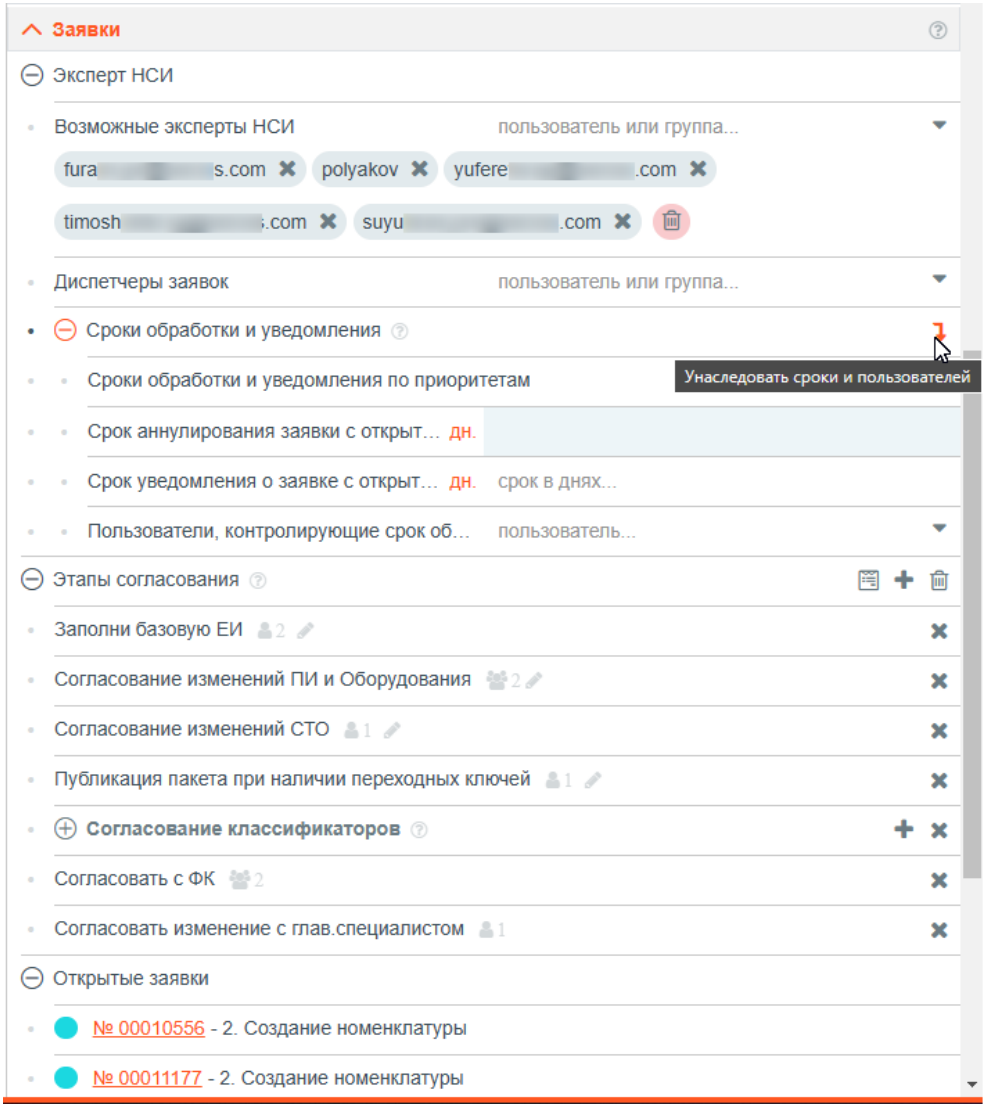


Рисунок 264 - Настройки справочной группы для работы с заявками

Если справочная группа имеет родительскую группу с уже настроенными сроками обработки и уведомления , то их можно унаследовать нажав на кнопку У***наследовать сроки и пользователей***, расположенную в соответствующей строке раздела **Заявки**.

Если для группы существуют унаследованные сроки обработки и уведомления и / или этапы согласования, то они могут быть перекрыты настройками группы. Для этого необходимо в соответствующей строке нажать на кнопку и подтвердить перекрытие (Рисунок 265). После этого можно будет ввести новые данные.

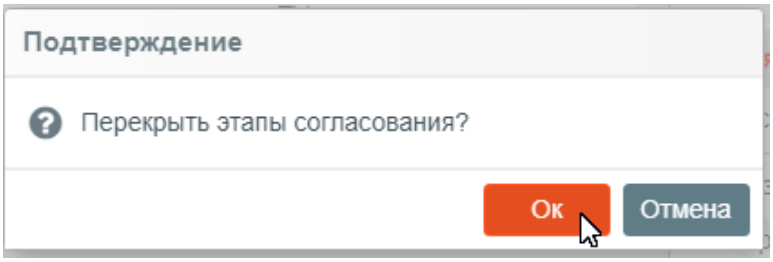


Рисунок 265 - Подтверждение перекрытия этапа согласования

Также в разделе **Заявки** карточки группы присутствует пункт *Открытые заявки,* где расположен список заявок (видны только исполнителям - *Экспертам НСИ*, указанным в шаблонах этих заявок), поданных в текущую и дочерние справочные группы и находящиеся в статусе *Ожидает обработки.* Щелчком левой кнопки мыши на номере в списке можно открыть заявку для ознакомления или обработки.

3.14.19 Создание справочной группы на основе метаданных

Метаданными в Semantic MDM, в том числе являются системные справочники. Таких, доступных для пользователя, справочников два – *Единицы измерения* и *Пользователи*.

В процессе работы часто возникает необходимость использовать атрибуты позиций таких справочников в других справочных группах в виде ссылок. Однако, напрямую обратиться к атрибутам системных справочников не представляется возможным. Для решения этой проблемы на основе системных справочников могут быть созданы обычные справочные группы, сохраняющие свою связь с системными объектами. В частности, подобный функционал может быть востребован для создания справочника сотрудников предприятия.

Создание справочной группы подчиняется общим правилам изложенным в разделе 3.14. При этом имеется три особенности:

- в *Semantic MDM* разрешено наличие только одной справочной группы, связанной с системным справочником;
- подключение системного справочника осуществляется в карточке

группы, в разделе **Связь с метаданными** (Рисунок 266). Системный справочник выбирается из выпадающего списка;

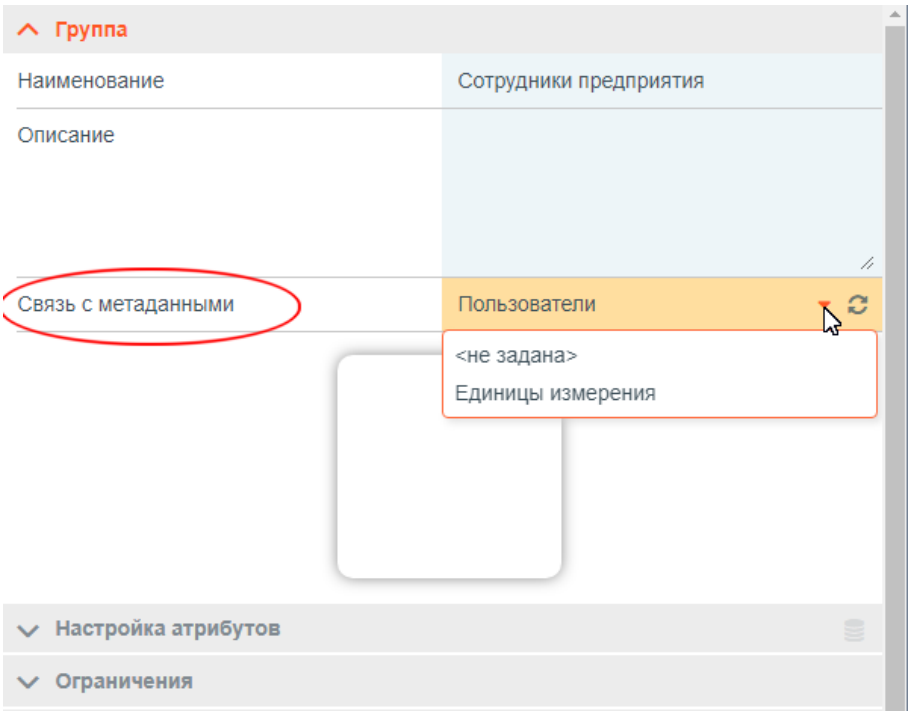


Рисунок 266 - Подключение системного справочника

- при подключении требуется осуществить настройку синхронизации атрибутов метаданных (системных справочников) с атрибутами группы. Для этого в атрибутах группы в ячейке **Атрибут метаданных** (Рисунок 267) из выпадающего списка должен быть выбран нужный атрибут системного справочника или атрибут LDAP (при подключении пользователей через Active Directory, подробнее см. раздел 3.29).

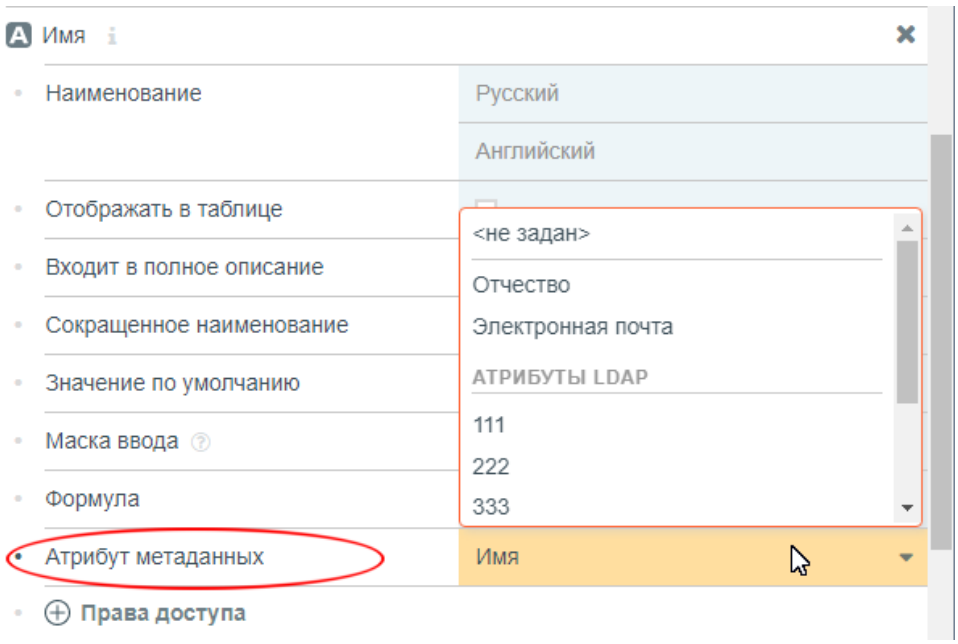


Рисунок 267 - Выбор системного атрибута

Настройка синхронизации выполняется в отдельном окне, которое вызывается кнопкой ***Синхронизировать***, в ячейке **Связь с метаданными**.

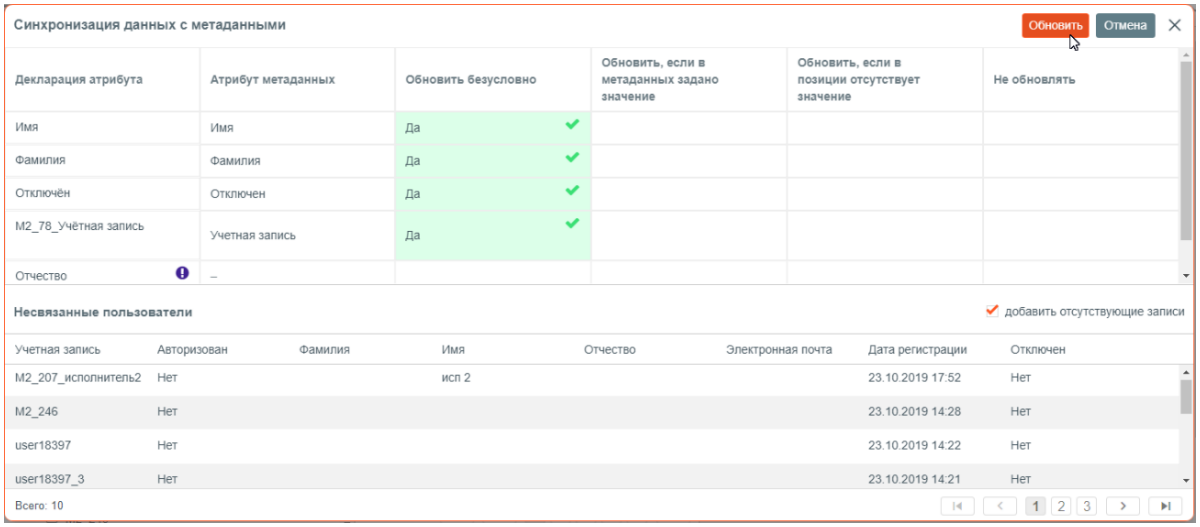


Рисунок 268 - Настройка синхронизации данных с метаданными

В окне **Синхронизация данных с метаданными** (Рисунок 268) выбирается способ синхронизации:

- **Обновить безусловно** – значение атрибута справочной группы всегда обновляется по значению метаданных;

- **Обновить, если в метаданных задано значение** – значение атрибута справочной группы обновляется, если в системном справочнике задано значение;
- **Обновить, если в позиции отсутствует значение** – значение атрибута справочной группы обновляется, если в оно в позиции справочной группы отсутствует (пустое);
- **Не обновлять** – значение атрибута справочной группы не синхронизируется с метаданными.

Выбор способа осуществляется двойным щелчком левой кнопки мыши в соответствующей ячейке в строке с наименованием атрибута. Выбранный вариант подсвечивается светло-зелёным цветом.

Установленный маркер **Добавить отсутствующие записи** позволяет добавлять при синхронизации в справочную группу новые записи (позиции) из системного справочника.

3.14.20 Статусы позиций

При работе с данными их текущее состояние определяется статусом (Таблица 19), который присваивается *Экспертом НСИ* или автоматически в зависимости от действий с объектами. Статус отображается в карточке позиции, в таблице данных, а также виден для агрегированных значений и в разделе **Связи** карточки позиции.

Таблица 19. Статусы позиций

Пиктограмма	Статус	Комментарий
l	**Нормализована**	Статус присваивается *Экспертом НСИ*. При переводе на данный статус Система выполняет проверку на соответствие заданным в настройках ограничениям. Если запись не соответствует ограничениям, то пакет изменений, содержащий данную запись запись с таким статусом не может быть опубликован
l	**Не нормализована**	Статус имеет любая запись, не проверявшаяся *Экспертом НСИ* или не соответствующая заданным ограничениям. Статус присваивается автоматически при создании объекта (позиции)

l	**Заблокирована**	Запись имеет переходные ключи с внешней информационной системой и блокирована от использования *Экспертом НСИ* до удаления переходных ключей с последующим (возможным) удалением записи
l	**Дубликат**	Позиция является дубликатом другой записи. Статус присваивается автоматически при выполнении дедубликации
l	**Удалена**	Позиция (группа) удалена. Статус носит служебное назначение и указывает какие позиции могут и / или должны быть удалены

По умолчанию в *Системе* включен переключатель **Только нормализованные** (Рисунок 269), расположенный под таблицей данных. При установке этой опции в таблице данных будут показаны только записи со статусом *Нормализована*. При отключении переключателя в таблице будут показаны записи с любыми статусами. Состояние переключателя в каждой таблице запоминается для каждого пользователя индивидуально. В режиме работы с пакетом изменений переключатель отключается автоматически. Позиции со статусами отличными от статуса *Нормализована* доступны только для пользователей с привилегиями редактировании данных.

При присвоении записи статуса *Нормализована*, а также при вводе новых позиций *Система* проверяет их на соответствие настроенным ограничениям. В случае если запись не соответствует всем ограничениям, то в строке появляется специальный знак красного цвета. Если *Система* обнаруживает, что объект удовлетворяет ограничениям только частично, то специальный знак в строке записи становится синим. При наведении курсора на пиктограмму можно увидеть список несоответствий.

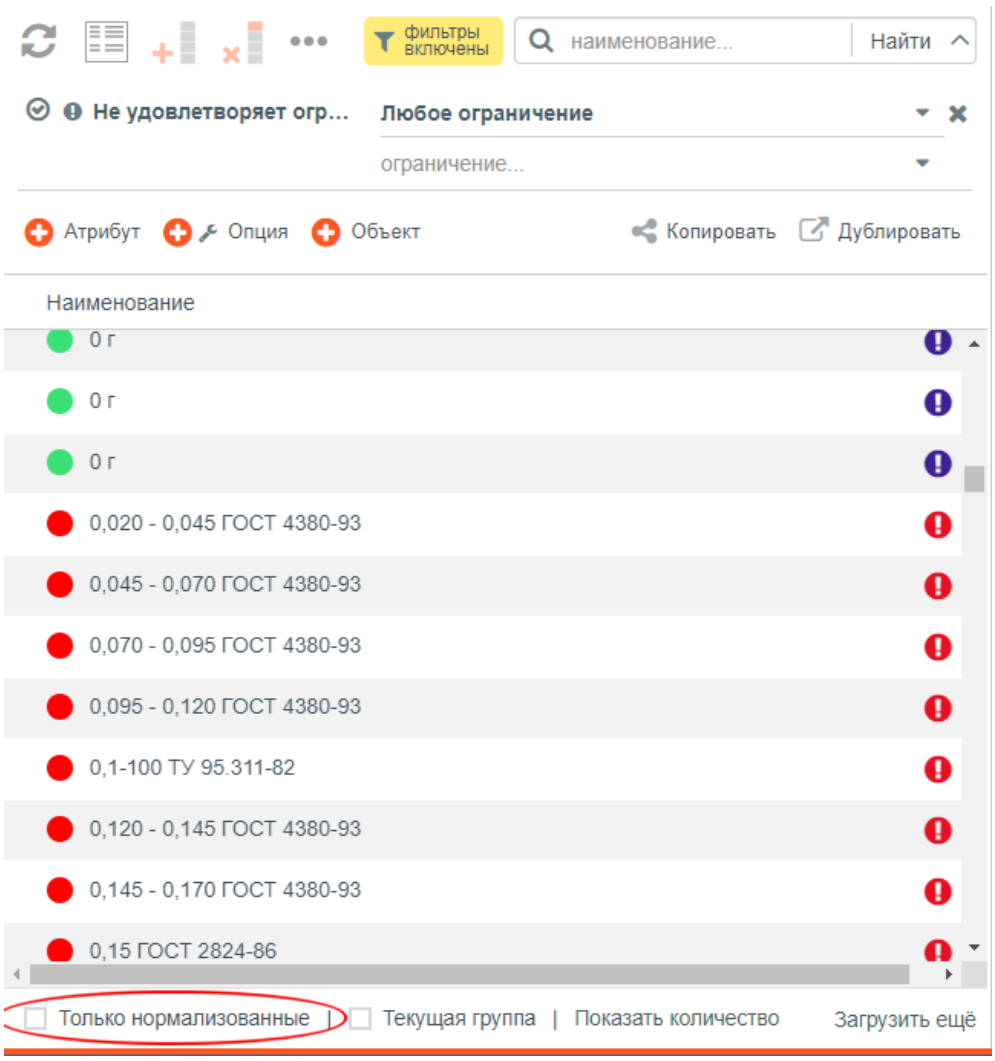


Рисунок 269 - Записи, полностью или частично несоответствующие указанным ограничениям

Записи с полным несоответствием ограничениям не могут быть опубликованы в случае присвоения статуса Нормализована! Записи с частичным несоответствием ограничениям могут быть опубликованы со статусом Нормализована, однако будут отмечены пиктограммой нарушения ограничений.

При установке любого статуса позиции доступно служебное поле **Комментарий к статусу**. Ввод комментария возможен при смене статуса в карточке позиции и массовом изменении статуса. **Комментарий к статусу** виден в карточке позиции и во всплывающей надписи (hint) при наведении курсора на значок с маркером статуса в таблице (Рисунок 270). Если

комментарий к статусу не заполнен, то поле в карточке позиции в режиме чтения не отображается.

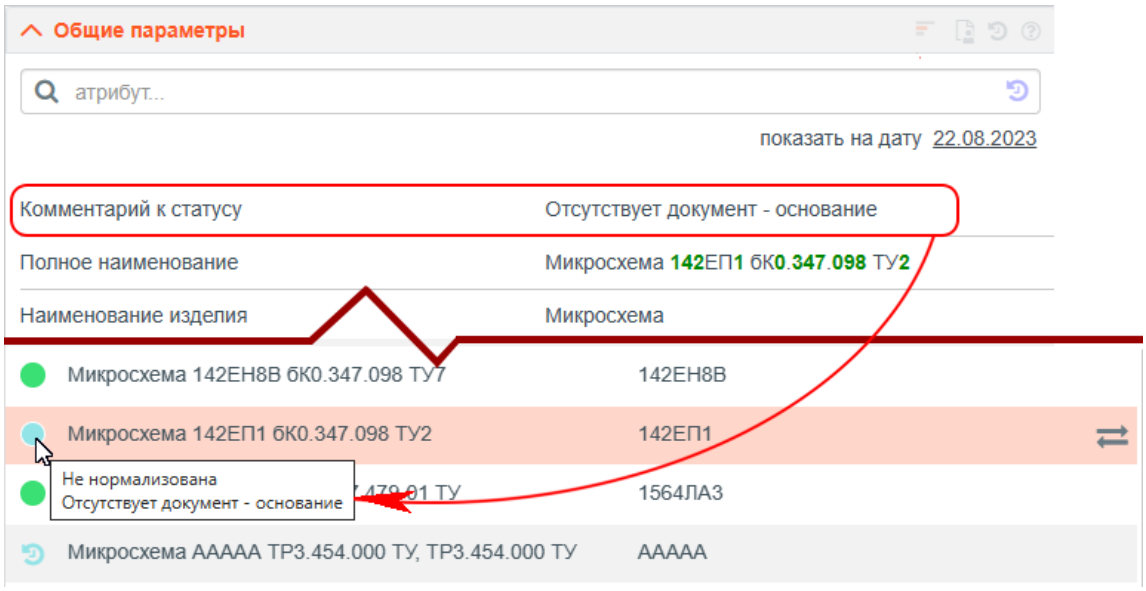


Рисунок 270 - Ввод и просмотр комментария к статусу позиции

3.15 Дедубликация справочных данных

При заполнении *Системы* справочными данными разными пользователями на протяжении длительного периода времени появляется риск повторного внесения справочной информации, когда одно и тоже понятие предметной области описывается несколькими объектами НСИ. Такие объекты называются дубликатами. Дубликаты понижают качество справочной информации и могут запутать пользователя, использующего такие данные.

Перед экспертами системы управления НСИ *Semantic MDM* стоит задача выявления дубликатов, определения оригинального объекта и удаления дублирующих. Стоит учесть, что удаляемые дубликаты могут быть применены ранее во внешних информационных системах, поэтому при их удалении *Система* должна запоминать идентификаторы удаляемых объектов.

Для автоматизации процесса дедубликации данных в *Semantic MDM* присутствуют соответствующие функции.

3.15.1 Создание варианта поиска дубликатов

Поиск дубликатов настраивается в раскрывающейся области (разделе) **Дубликаты** (Рисунок 271) карточки справочной группы, позиции и классификатора. Вариант поиска дубликатов, настроенный на родительском уровне, наследуется дочерними объектами.

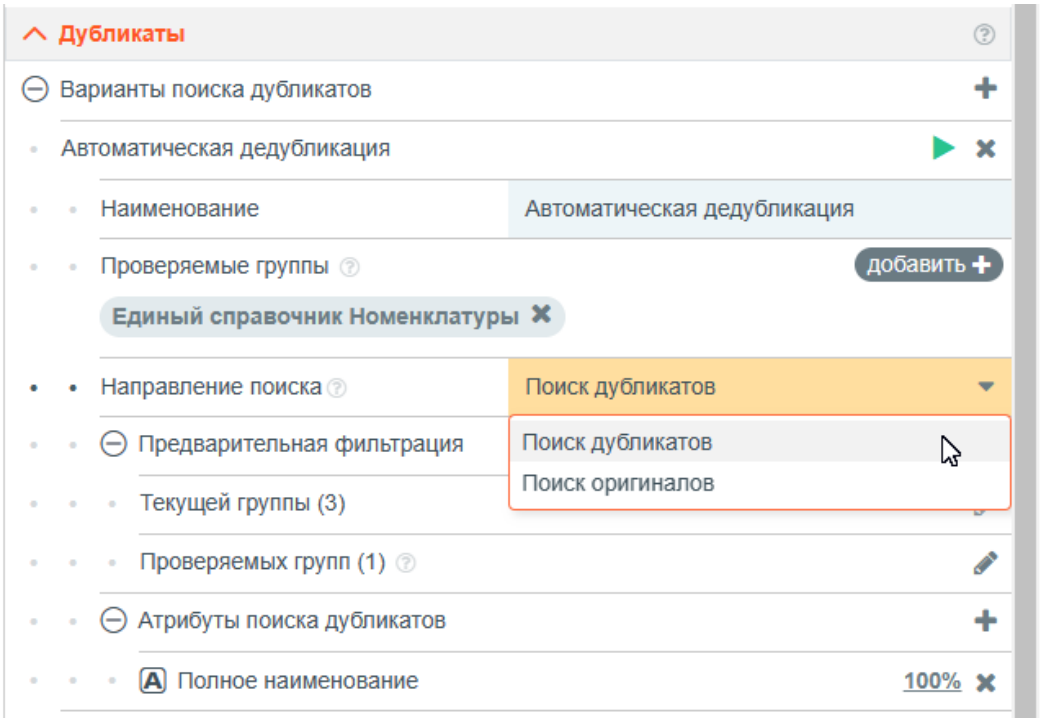


Рисунок 271 - Область **Дубликаты** в карточке объекта

Для создания варианта поиска дубликатов необходимо выполнить следующую последовательность действий[18]:

- раскрыть область **Дубликаты**, нажав кнопку в левой части заголовка;
- нажать кнопку ***Добавить вариант поиска*** в правой части строки **Варианты поиска дубликатов**;
- в открывшемся списке заполнить поля **Наименование** и **Проверяемая группа**. По умолчанию проверяемой группой является текущая (то

[18] *Для поиска дубликатов пользователь должен иметь роль с привилегией «Поиск дубликатов». Привилегии для ролей настраиваются системным администратором Semantic MDM*

есть, та группа, на которой в данный момент установлен курсор в дереве);

- **Направление поиска**:
  - *Поиск дубликатов* – в текущей справочной группе содержаться оригиналы, в проверяемой группе выполняется поиск дубликатов;
  - *Поиск оригиналов* – в текущей справочной группе содержаться дубликаты, в проверяемой группе выполняется поиск оригиналов. Данный вид применяется, в том числе, для поиска позиций при заполнении заявки. Для данного направления поиска доступны опции:
    - **Использовать при сопоставлении при импорте** – установка данного маркера означает, что при выполнении импорта в *Semantic MDM* из внешней ИС по данному варианту будут определяться дубликаты записей. Функция применяется в том числе, при поиске записей, имеющих композиции с другими объектами;
    - **Использовать для автоматического слияния** – дубликаты, найденные при импорте из внешней ИС, будут автоматически объединены с записью в *Semantic MDM*;

Данные опции запрещено включать, если поиск дубликатов содержит в настройках поиска дубликатов атрибут непосредственно сам атрибут агрегации, а не вложенный в него атрибут.

- в разделе **Предварительная фильтрация** можно выполнить отбор позиций по статусу или значениям атрибутов, среди которых осуществляется поиск дублей:
  - **Текущей группы** – группа, в которой установлен курсор и где находятся оригиналы объектов;
  - **Проверяемой группы** – группа, в которой будет осуществляться поиск и где возможно содержаться дубликаты;

- добавить атрибут поиска, нажав кнопку ***Добавить атрибут*** ➕ в правой части строки **Атрибуты поиска**;
- указать атрибуты текущей и проверяемой группы путём выбора из выпадающего списка под кнопкой ▾ или путём выбора в окне **Классификатор атрибутов**, которое открывается нажатием кнопки 🗁 в правой части строки. Для атрибутов связи (агрегации и композиции), если они были выбраны в качестве ключа для поиска дубликатов дополнительно устанавливается **Режим сравнения множественных значений** (Рисунок 272)**:**
  - *Совпадает хотя бы одна (хотя бы для одной позиции «слева» найдена позиция «справа»)* - среди множества значений атрибута у позиции - оригинала («слева») и позиции - дубликата («справа») есть хотя бы одно совпадающее;
  - *Совпадают все слева (для каждой композиции «слева» найдена позиция «справа»)* - для всего множества значений атрибута у позиции - оригинала («слева») найдены соответствующие значения у позиции - дубликата («справа»), при этом значений у атрибуты позиции - дубликата может быть больше;
  - *Полное соответствие (для каждой композиции «слева» найдена позиция «справа»)* - для всего множества значений атрибута у позиции - оригинала («слева») найдены соответствующие значения у позиции - дубликата («справа»), при этом количество значений атрибута у позиций «слева» и «справа» одинаковое;

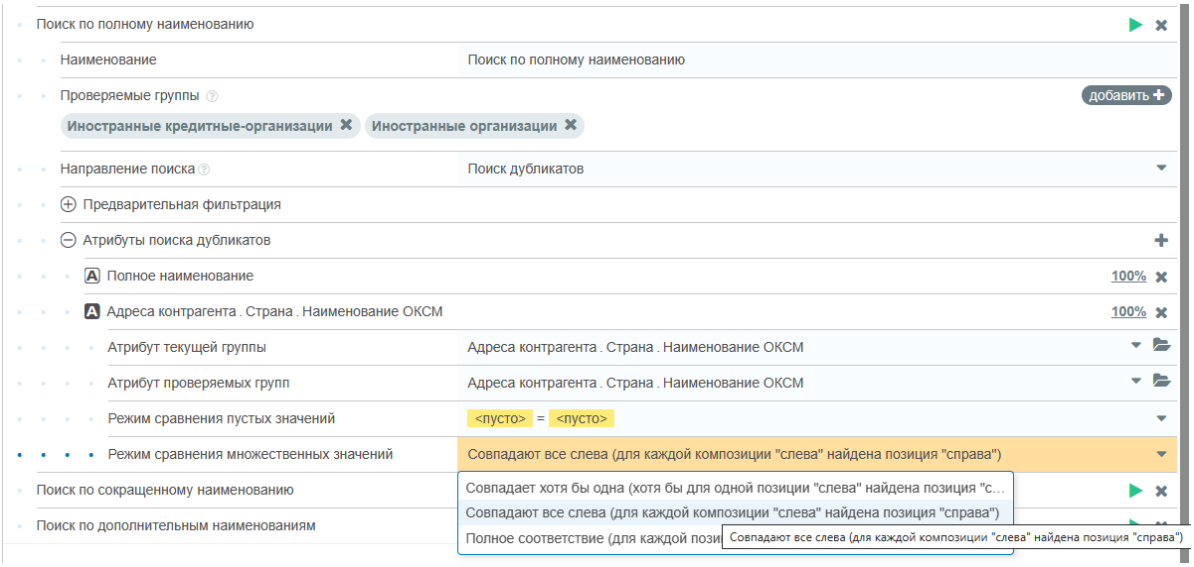


Рисунок 272 - Настройка режима сравнения множественных значений для атрибутов связи

- установить режим сравнения пустых значений:
  - <пусто> = <пусто> – объект признается дубликатом, если значение атрибута у оригинала и дубликата пустые;
  - <пусто> <> <пусто> – объект не признается дубликатом, если значение атрибута у обоих сравниваемых позиций пустое;
- в строке с наименованием атрибута указать процент совпадения (релевантности) значений атрибута у оригинала и дубликата (100% соответствует полному совпадению и устанавливается по умолчанию);
- добавить и настроить необходимое количество атрибутов поиска;
- нажать кнопку ▶ ***Запустить поиск*** рядом с наименованием варианта поиска.

Поиск дубликатов ведётся в фоновом режиме. О чём сообщает специальная пиктограмма в панели инструментов (Рисунок 273). Нажатие на пиктограмму позволяет открыть специальное окно, где отображается дата и время запуска поиска и процент готовности операции (Рисунок 274). Аналогичные сведения можно увидеть в компактном режиме в разделе **Дубликаты** рядом с наименованием варианта поиска.

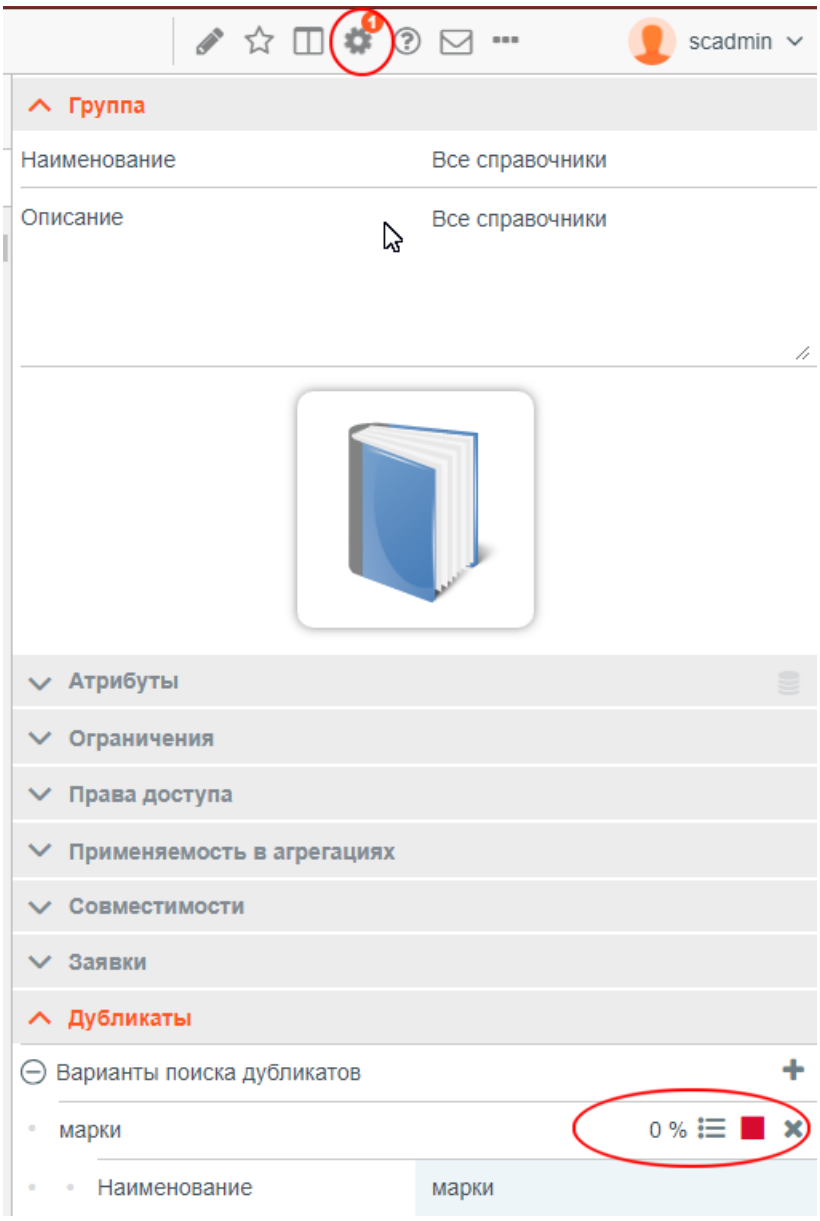


Рисунок 273 - Выполнение поиска дубликатов в фоновом режиме

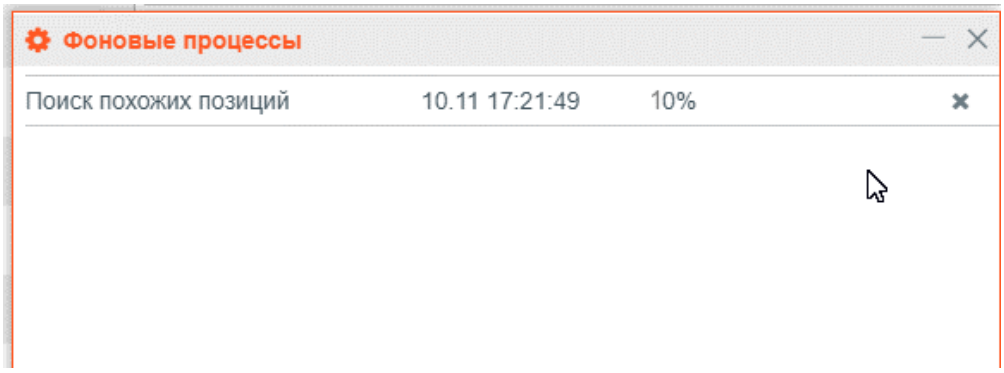


Рисунок 274 - Сведения о процессе поиска дубликатов

Результат поиска дубликатов сохраняется в зависимости от того, как был запущен поиск дубликатов – рамках открытого пакета изменений или без

открытого пакета изменений. Если поиск дубликатов был запущен в рамках открытого пакета изменений, то поиск выполняется только по измененным позициям, а результат поиска дубликатов сохраняется в рамках открытого пакета изменений. После публикации или отмены пакета изменений, результаты такого поиска дубликатов будут удалены. Если поиск дубликатов был запущен в режиме чтения (без активного пакета изменений), то результаты поиска сохраняются в базу данных и доступны всем пользователям, имеющим привилегию на редактирование данных.

Варианты поиска дубликатов наследуются вниз по дереву справочника. Изменить настройки унаследованного варианта поиска дубликатов нельзя. Для унаследованного варианта поиска дубликатов доступны только возможности просмотра настроек и запуска варианта поиска.

Удаление варианта поиска дубликатов осуществляется нажатием кнопки в правой части строки с наименованием варианта поиска.

3.15.2 Анализ результатов поиска дубликатов

По завершению процесса поиска дубликатов в правой части строки с наименованием варианта поиска становится доступна кнопка ***Результаты поиска***, по нажатию на которую открывается окно **Дубликаты по варианту поиска <наименование варианта поиска>** (Рисунок 275), в котором пользователь может оценить результат поиска дубликатов и выполнить объединение дублирующих объектов.

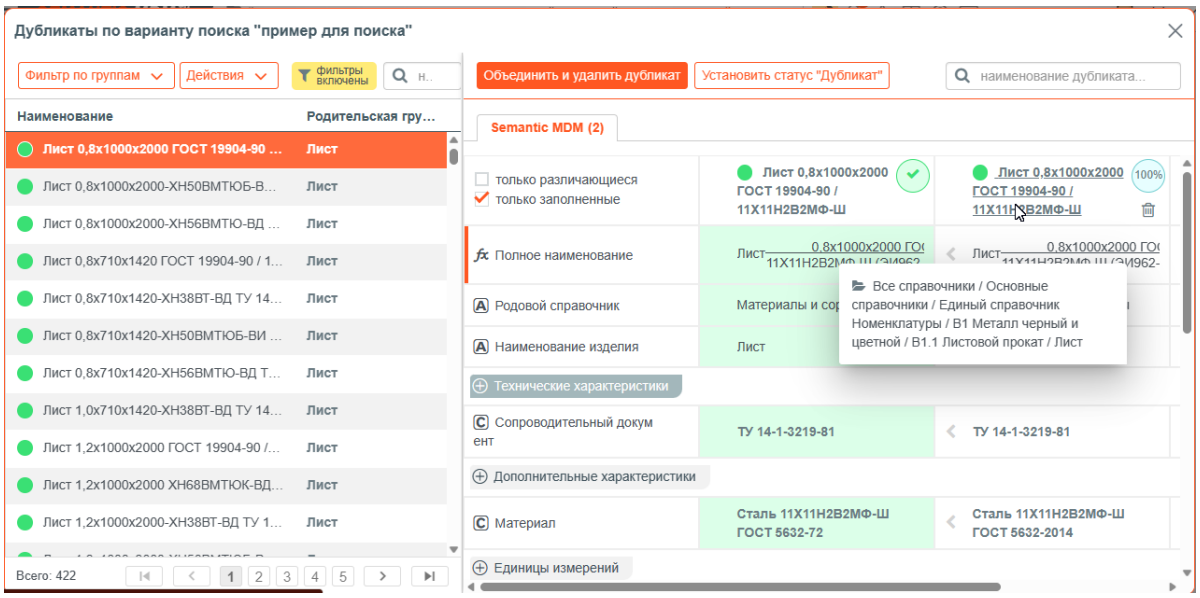


Рисунок 275 - Окно **Дубликаты по варианту поиска**

В левой части окна **Дубликаты по варианту поиска** расположен список всех найденных объектов с из статусами в *Системе*, удовлетворяющих условиям варианта поиска дубликатов. Список имеет свою инструментальную панель, включающую следующие элементы:

- кнопка ***Фильтр по группам*** – позволяет отфильтровать объекты в списке по выбранным группам. Для выбора группы фильтрации необходимо в выпадающем списке кнопки установить маркер слева от нужных групп и нажать кнопку ***Ок***;
- поле быстрого поиска <наименование оригинала> – позволяет найти нужный объект по наименованию.

В правой части окна **Дубликаты по варианту поиска** расположена таблица со значениями атрибутов объекта-оригинала и объектов-дубликатов. Объект-оригинал обозначается значком , а для объектов-дубликатов устанавливается значок с указанием процента совпадения . При наведении курсора на наименование найденного объекта во всплывающем окне (хинте) показывается путь к справочной группе, в которой находится данная позиция. Щелчком левой кнопки мыши по значку можно назначить любую из позиций

оригиналом, ранее назначенная оригиналом позиция станет считаться дубликатом с соответствующей пиктограммой.

Отображение информации для дедублицируемых позиций унифицировано с карточкой позиции и заявками. Для всех атрибутов отображаются категории с настроенной для них подсветкой, в том числе и для множественных атрибутов связи, а при раскрытии композируемых позиций выполняется затенение родительских атрибутов.

Над объектами-дубликатами можно выполнить следующие действия (Рисунок 276):

- сравнить атрибуты оригинала и дубликатов, установив маркеры:
  - **только различающиеся** –будут показаны только те атрибуты, значения которых отличаются у оригинала и дубликатов;
  - **только заполненные** – будут показаны только те атрибуты объектов, которые имеют заполненные значения;
  - комбинация <**только различающиеся + только заполненные**> – будут показаны те атрибуты, которые заполнены и значения которых отличаются у оригинала и дубликатов;
  - **дубликат Semantic MDM** – доступен только на вкладке с результатами дедубликации переходных ключей с внешними ИС, показывает только те атрибуты, которые присутствуют в *Системе*;
- выбрать значения атрибутов для назначения объекту-оригиналу;
- удалить найденную позицию-дубликат из результатов поиска по кнопке «Не является дубликатом»;
- ***Объединить и удалить дубликат*** – кнопка над таблицей с позициями-дубликатами позволяет для просматриваемой позиции объединить настроенные данные и удалить все дубликаты;
- ***Установить статус «Дубликат»*** – кнопка над таблицей с позициями-дубликатами позволяет для просматриваемой позиции объединить настроенные данные, а позициям-дубликатам присвоить статус «Дубликат»;

- при помощи кнопки ***Действия*** выполнить команды:
- **Удалить дубликаты для выбранной позиции** – позволяет для просматриваемой позиции объединить настроенные данные и удалить все дубликаты;
- **Установить статус Дубликат для дубликатов выбранной позиции** – позволяет для просматриваемой позиции объединить настроенные данные, а позициям-дубликатам присвоить статус «Дубликат»;
- **Удалить дубликаты для всех позиций** – позволяет выполнить массовую дедубликацию как для выбранных позиций в списке слева, так и для всех найденных;
- **Установить статус Дубликат для дубликатов всех позиций** – позволяет выполнить массовое присвоение статуса «Дубликат» всем позициям.

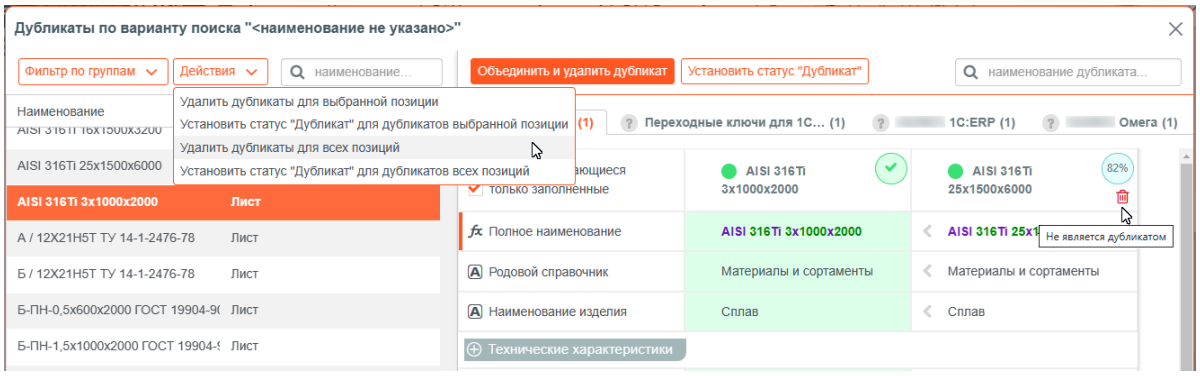


Рисунок 276 - Действия над найденными дубликатами позиций

Чтобы выбрать значение атрибута для присвоения его объекту-оригиналу необходимо выполнить двойной щелчок левой кнопкой мыши в ячейке с нужным значением атрибута. Выбранная ячейка окрашивается в зелёный цвет и в правой части ячейки устанавливается знак . В этом случае при выполнении команды **Объединить** *Система* выполнит слияние атрибутов объекта-оригинала и объектов-дубликатов. Позиции со статусом "Дубликат" получат статус «Не применимо».

При выполнении дедубликации позиций, у которых имеются множественные композиции, выполняется автоматическое сопоставление композируемых позиций дубликатов, композируемым позициям оригиналов. Автоматическое сопоставление выполняется при установленной опции «Использовать для автоматического сопоставления».

Сопоставление множественных значений при дедубликации выполняется при следующих условиях:

- *Множественные значения простых атрибутов* – множественные значения дедублицированных позиций сопоставляются по значению. Система позволяет выбрать отсутствующие значения в эталоне из дублирующих позиций;
- *Множественные агрегации* – множественные агрегации дедублицируемых позиций сопоставляются по идентификаторам агрегируемых позиций. Система позволяет выбрать отсутствующие агрегации в эталоне из дублирующих позиций;
- *Множественные композиции* - множественные композиции дедублицируемых позиций сопоставляются через поиски дубликатов, определенных в композируемых группах. Система позволяет выбрать отсутствующие композиции в эталоне из дублирующих позиций, выполнить ручное сопоставление композируемых позиций, а также выполнить настройку сопоставленных композируемых позиций через дополнительный диалог настройки дедубликации. Вложенность настройки дедублицируемых композиций не ограничена.

В диалоге дедубликации возможно ручное изменение сопоставлений, выполненных автоматически. Пользователь должен с помощью Drag&Drop указать какие композиции дубликата должны быть добавлены, а какие сопоставлены с существующими.

После выполнения настройки дедубликации множественных композиций, требуется выполнить слияние множественных композиций в соответствии настройками, выполненными пользователем.

Если в результате дедубликации позиции был присвоен статус «Дубликат», то сведения об этом сохраняются в карточке позиции - дубликата в виде ссылки на позицию – оригинал (Рисунок 277).

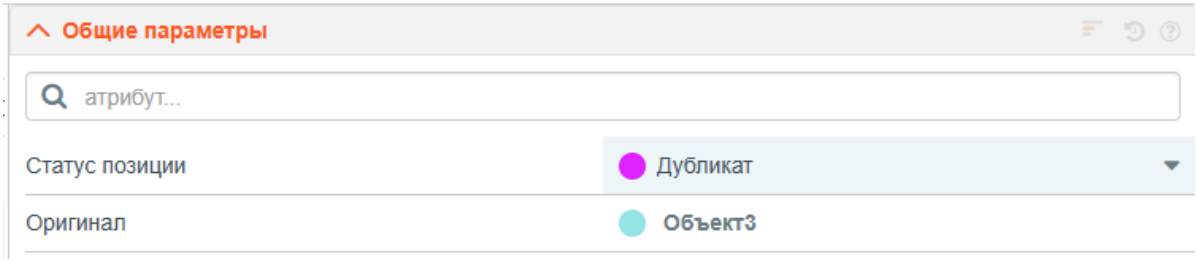


Рисунок 277 - Ссылка на оригинал в карточке позиции со статусом «Дубликат»

В карточке позиции - оригинала также отображается ссылка на позицию - дубликат в поле под статусом позиции. Кроме этого подробные сведения о дубликатах доступны в разделе **Дубликаты** в категории *Позиции, признанные дубликатами*, где отображается GUID позиции -дубликата, а также адрес ее размещения в *Системе* (Рисунок 278).

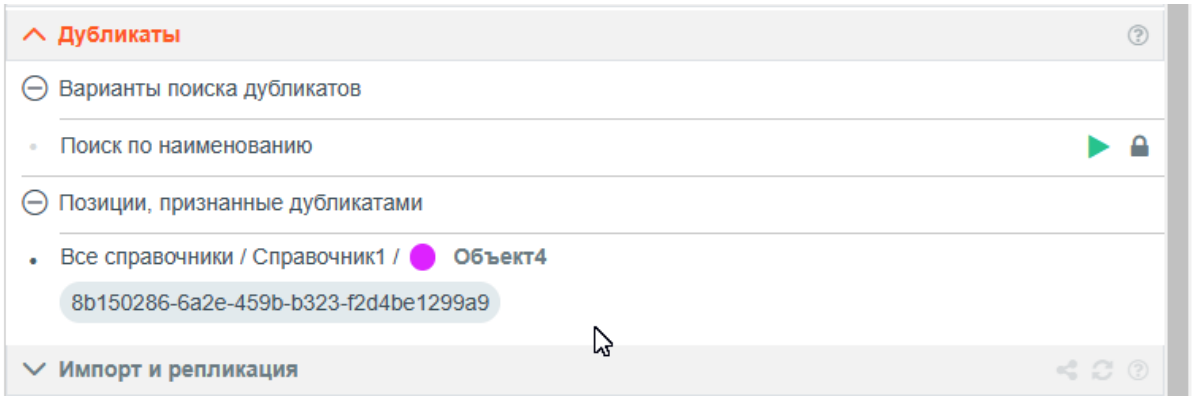


Рисунок 278 - Сведения о GUID дубликата в карточке позиции - оригинала

3.15.3Дедубликация выбранной позиции

В *Системе* возможно выполнение дедубликации выбранной позицию без обращения к сервису дедубликации. Для этого в режиме управления пакетами изменений следует установить курсор на нужной позиции и вызвать из контекстного меню команду **Признать дубликатом** (Рисунок 279). После

этого нужно указать (выбрать) позицию-оригинал. Далее, *Система* отобразит окно для настройки слияния значений атрибутов эталонной позиции и позиции дубликата.

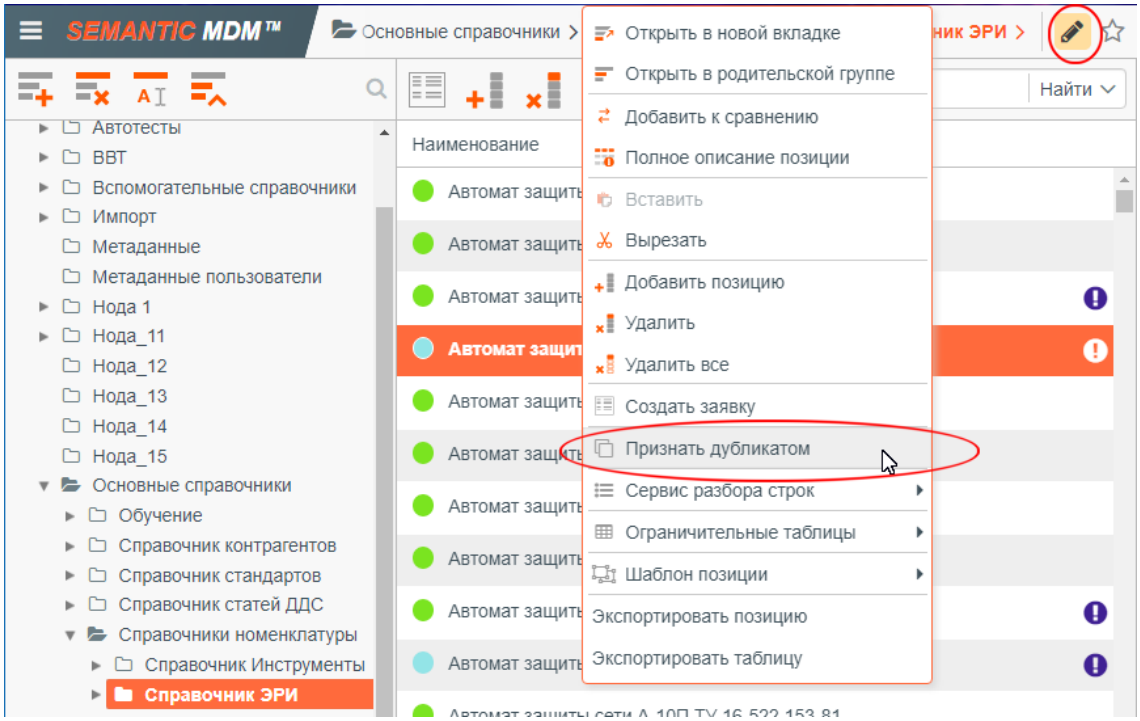


Рисунок 279 - Вызов команды **Признать дубликатом**

3.15.4 Дедубликация позиций с внешней применяемостью

Для любого справочника может быть настроена вкладка **Внешняя применяемость** (см. п. 3.31.1), где отображаются дополнительные сведения о позиции, получаемые из внешней информационной системы, например, складские остатки.

При выполнении дедупликации позиций, имеющих внешнюю применяемость, возникает необходимость указать, какая из найденных записей является оригиналом, а какая дубликатом. Для таких позиций в окне **Слияние дубликатов** (Рисунок 280) появляется дополнительная вкладка (вкладки) с названием внешней ИС, где дополнительно можно указать, данные по внешней применяемости какого из объектов считать оригинальными. Сама процедура полностью аналогична описанной в п. 3.15.2.

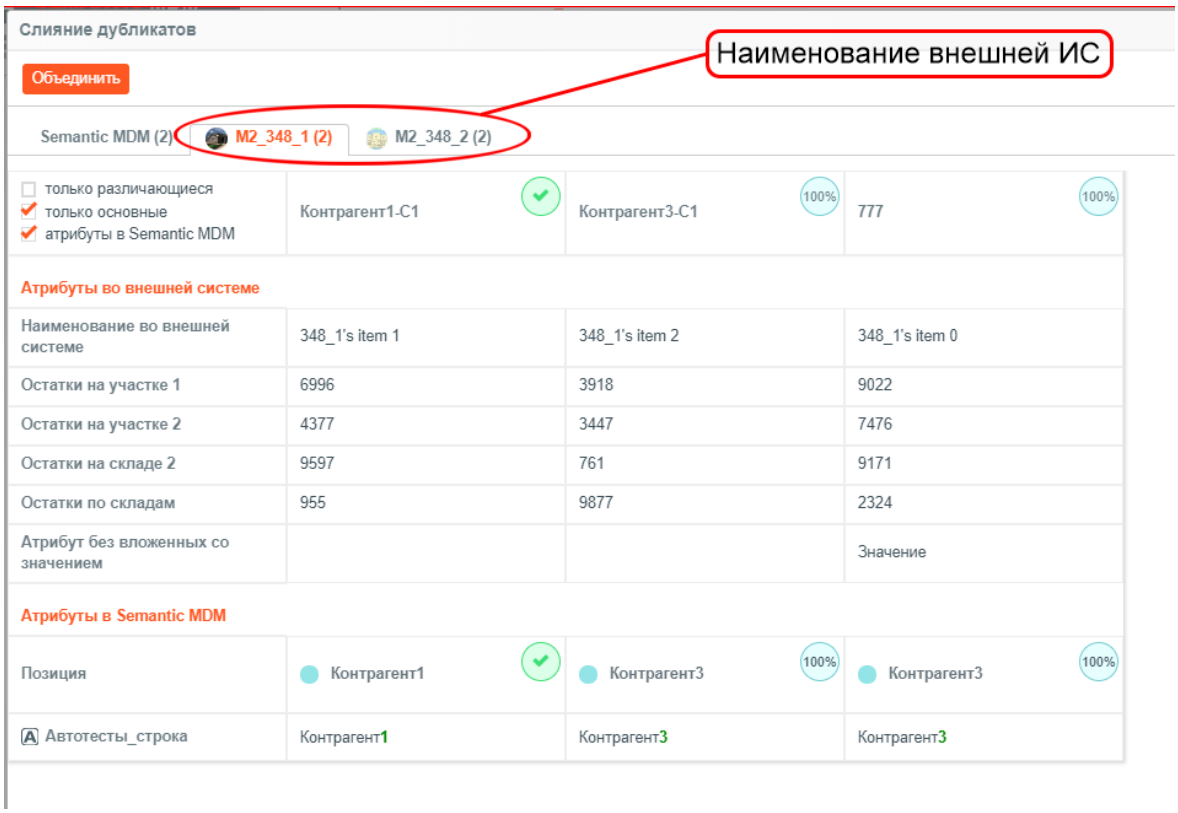


Рисунок 280 - Работа с дедупликацией позиций, имеющих внешнюю применяемость

И позиция-оригинал и позиция-дубликат будут отображены на вкладке **Внешняя применяемость** (Рисунок 281). Пиктограмма ✔ соответствует оригиналу.

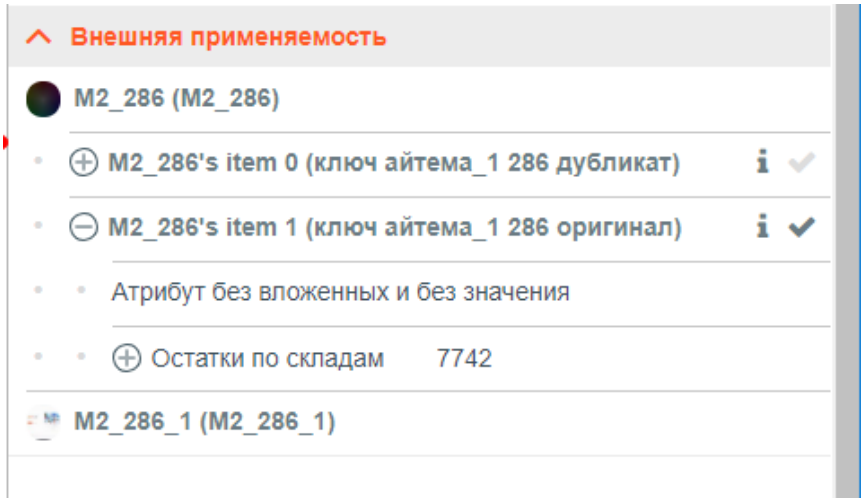


Рисунок 281 - Позиция-оригинал и дубликат в разделе **Внешняя применяемость**

3.15.5 Дедубликация позиций с переходными ключами

При выполнении дедубликации может оказаться, что у позиций присутствуют переходные ключи с несколькими разными внешними информационными системами (см. п. 3.30.14) и в некоторых случаях, дедублицируемые позиции могут иметь переходные ключи с одной и той же внешней ИС. Таким образом, возникает необходимость на этапе дедубликации выбрать позицию во внешней системе, которая должна считаться эталоном (оригиналом).

Если при выполнении дедубликации *Система* обнаруживает, что хотя бы для одной внешней ИС имеется более одного переходного ключа, то в окне **Слияние дубликатов** (Рисунок 282) появляется дополнительная вкладка с наименованием внешней ИС, где расположены идентификаторы переходных ключей из которых требуется выбрать оригинал.

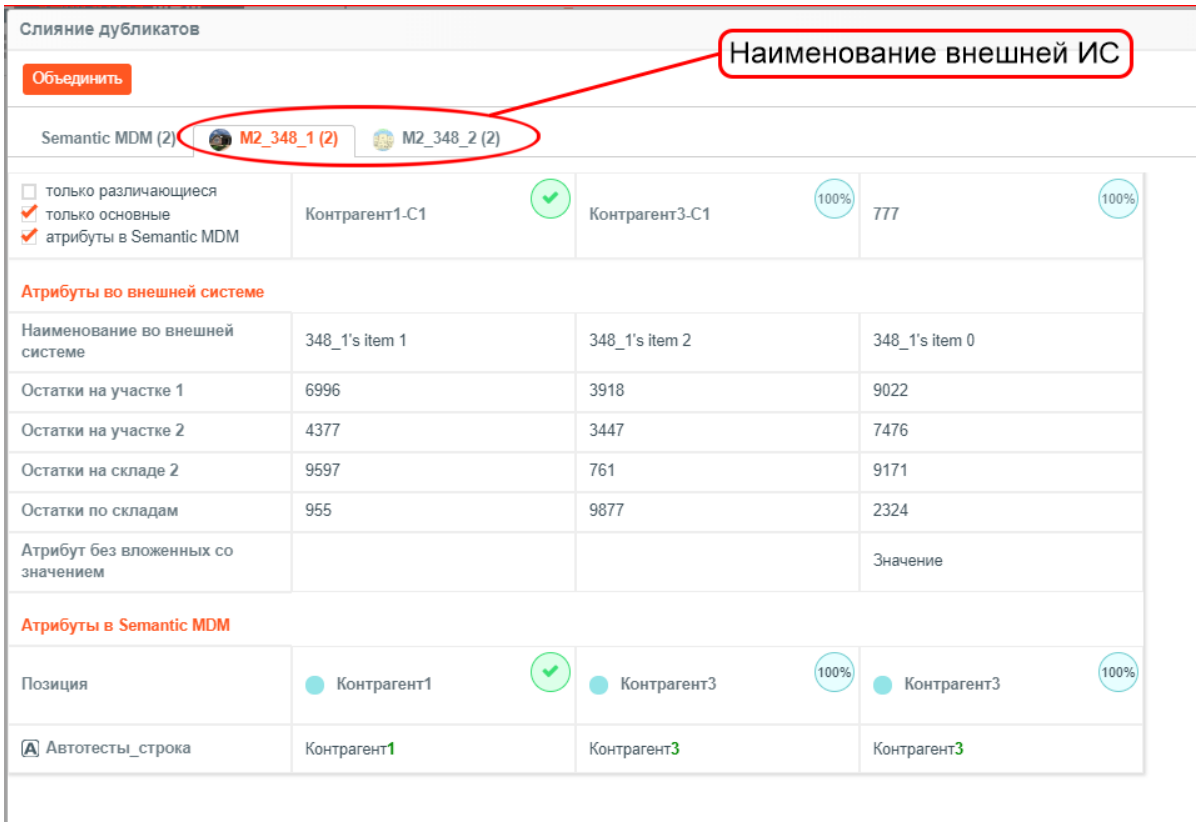


Рисунок 282 - Работа с дедубликацией позиций, имеющих переходные ключи

Процедура дедубликации позиции во внешней системе полностью аналогична описанной в п. 3.15.2, но выполняется на соответствующей вкладке.

3.16 Классификатор атрибутов

Все атрибуты, используемые в различных справочных группах, создаются и хранятся в специальном разделе – **Классификатор атрибутов**. Для удобства атрибуты собраны в группы, соответствующие справочникам (справочным группам), в которых они используются, однако, при настройке новых групп пользователь может использовать любые атрибуты, созданные ранее и применяемые в других справочных группах.

Создание новых атрибутов может происходить в **Классификаторе атрибутов** при обращении к нему в разделе **Конфигурация данных** (выбирается из меню кнопки **Навигация**), а также непосредственно из раздела **Настройка атрибутов** (см. п. 3.14.7) любой справочной группы. Последовательность и правила создания новых атрибутов при любом варианте будут одинаковыми.

Для каждого атрибута в карточке можно узнать в каких справочных группах, формулах, ограничениях и обменных классах он применяется.

3.16.1 Просмотр списка атрибутов

В **Классификаторе атрибутов** все атрибуты разделены по группам. Деление на группы может происходить достаточно произвольно – на основании принадлежности к справочнику, типу данных, описываемым свойствам и т.д. В этом случае актуальной является задача просмотра сквозного списка атрибутов, которые принадлежат текущей или подчинённой классификационной группе.

В зависимости от положения курсора в **Конфигураторе атрибутов** возможны два варианта просмотра списков:

- при установке курсора на классификационной группе в дереве, после

нажатия на кнопку ***Отображать атрибуты подчинённых групп*** (Рисунок 283) в таблице будут показаны все атрибуты, принадлежащие текущей и подчинённым классификационным группам;

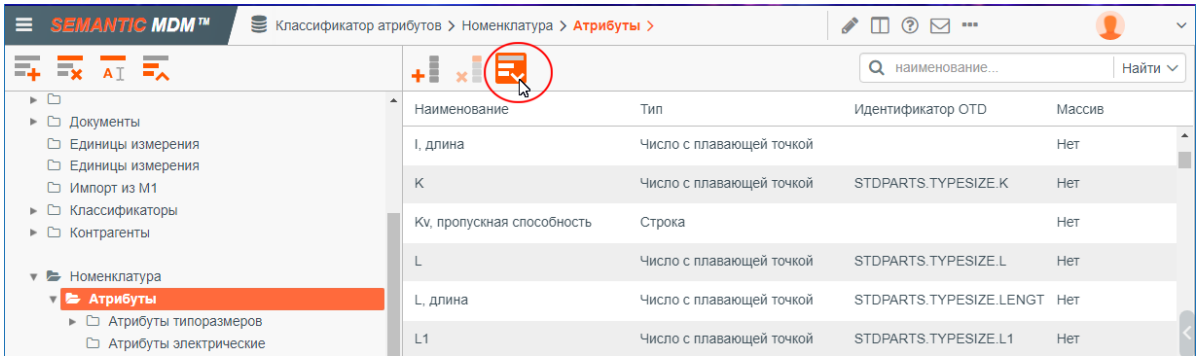


Рисунок 283 - Просмотр списка атрибутов подчинённых групп

- если курсор установлен на одном из атрибутов в таблице, то после нажатия на кнопку ***Отображать атрибуты подчинённых групп***, курсор будет спозиционирован на группе, которой принадлежит этот атрибут.

Система сохраняет выделенный атрибут в фокусе при включении/отключении режима просмотра сквозного списка атрибутов. Если включить режим отображения сквозного списка атрибутов, а затем выделить атрибут и отключить режим, Система выделит группу, которой принадлежит этот атрибут. То же самое работает в обратную сторону, например, если выделить атрибут и включить режим отображения сквозного списка атрибутов, а потом перейти на одну из родительских групп, то выделенный ранее атрибут останется в фокусе.

3.16.2 Поиск атрибутов в классификаторе

Перед добавлением нового атрибута следует убедиться, что подобный атрибут не был зарегистрирован в *Системе* ранее.

Для поиска атрибута следует в окне быстрого поиска ввести наименование атрибута, после чего в дереве будет подсвечена группа атрибутов, содержащая атрибуты подобные искомому (Рисунок 284).

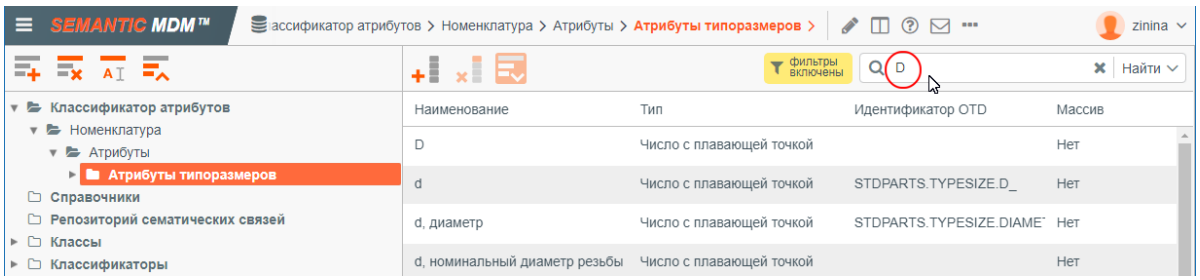


Рисунок 284 - Поиск атрибута в **Классификаторе атрибутов**

При создании атрибута из справочника также можно воспользоваться быстрым поиском (Рисунок 285).

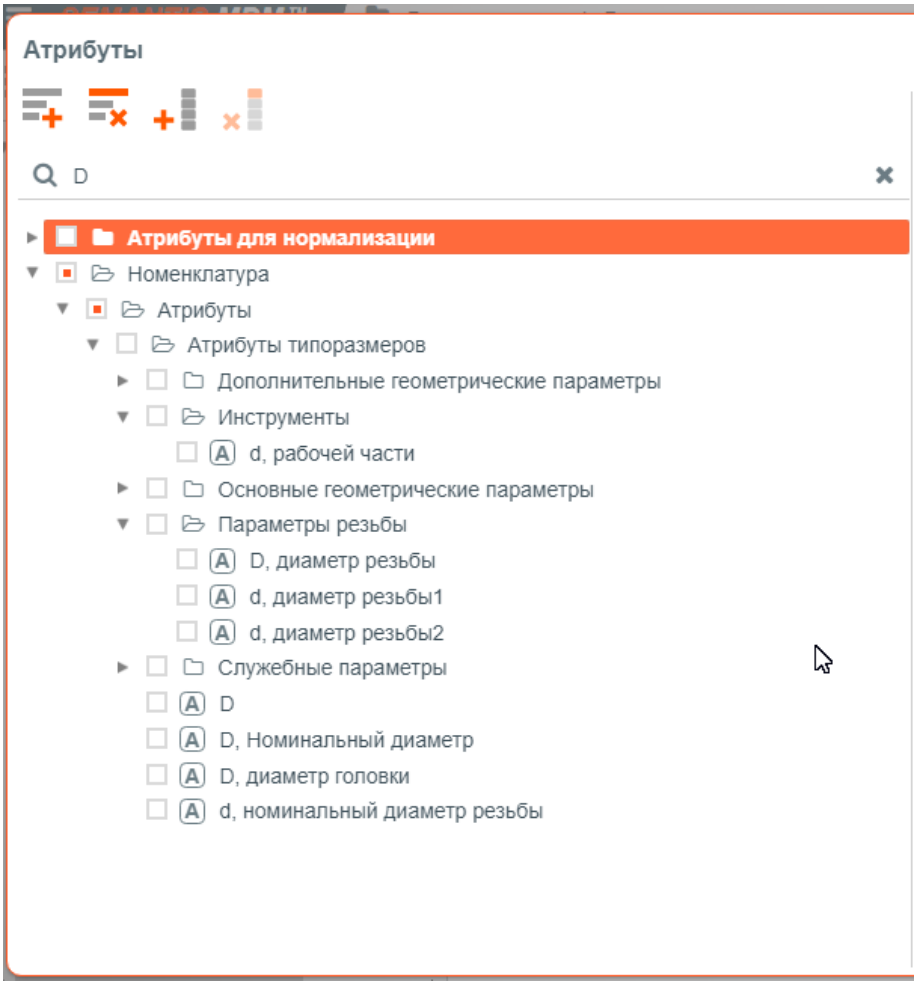


Рисунок 285 - Поиск атрибута в списке при создании его из справочника

Кроме быстрого поиска, в представлении **Конфигурация данных** можно воспользоваться поиском по критериям (расширенным поиском). Для этого необходимо нажать кнопку ***Расширенный поиск***, а затем кнопку ***Критерий*** и выбрать один или несколько критериев из списка (Рисунок 286).

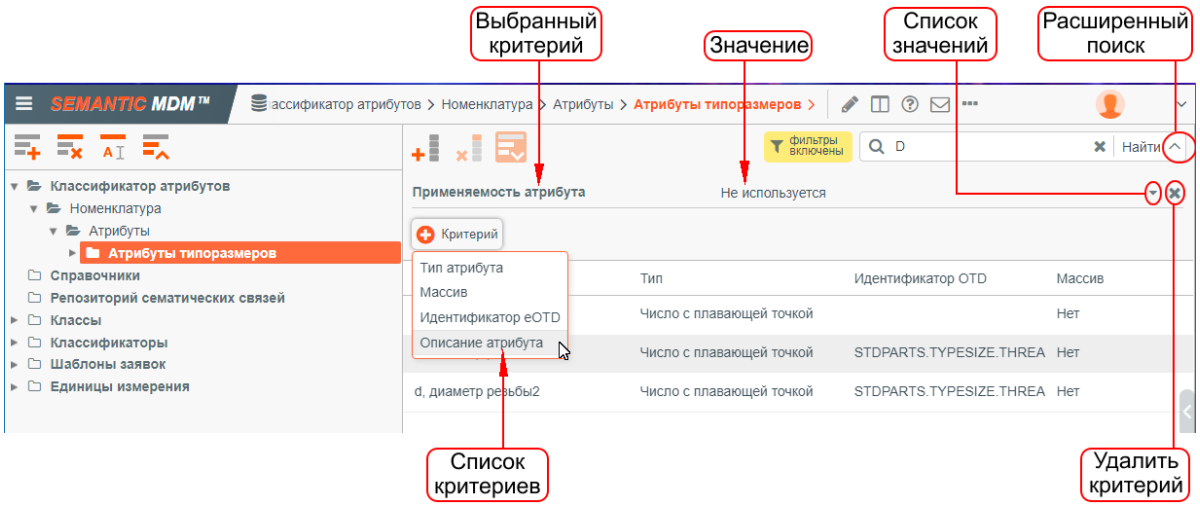


Рисунок 286 - Настройка расширенного поиска по атрибутам

Для расширенного поиска атрибутов доступны несколько критериев, представленных в таблице ниже.

Таблица 20. Критерии поиска по атрибутам

Критерии поиска	Комментарий
Применяемость атрибута	Принимает два значения: не используется – атрибут не применяется ни в одном из справочников; отсутствуют значения – значения атрибута в справочнике не заполнены (пустые)
Тип атрибута	Комплексный критерий состоящий из двух параметров – тип атрибута (простой или атрибут связи) и тип данных атрибута
Массив	Маркер, определяющий является ли искомый атрибут массивом.
Идентификатор eOTD	Позволяет производить поиск по значению идентификатора
Описание атрибута	В качестве значения выступает описание атрибута

Расширенный поиск работает совместно с быстрым, то есть быстрым поиском пользователь может отфильтровать атрибуты по наименованию, а затем с помощью расширенного уточнить возможные значения.

3.16.3 Создание простого атрибута

Простым является любой атрибут, описывающий одно из свойств объекта. В зависимости от описываемых свойств и назначения он может иметь различный тип.

Для создания простого атрибута следует:

- установить курсор на группе, в которую добавляется атрибут, и перейти в таблицу данных. При необходимости можно создать новую группу, используя команду из контекстного меню или кнопку на панели инструментов дерева **Добавить группу** (Рисунок 287);

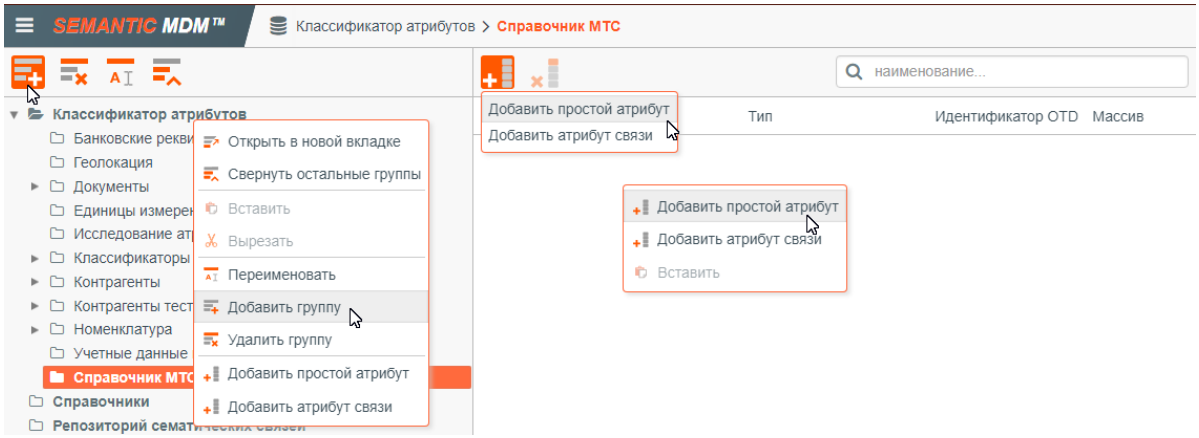


Рисунок 287 - Добавление нового атрибута

- в таблице данных нажать кнопку ***Добавить*** или вызвать контекстное меню и выбрать команду **Добавить простой атрибут** (Рисунок 287);
- перейти в раздел **Общие параметры** (Рисунок 288) и задать наименование атрибута в соответствующей ячейке. Наименование на английском языке присваивается по необходимости;
- в ячейке **Описание** можно дать комментарий о назначении атрибута и его использовании (рекомендуется давать описание, с целью различения атрибутов, аналогичных по наименованию);
- **Источник (документ)** – в ячейке рекомендуется указывать документ (справочник, стандарт и т.п.), согласно которому введён данный атрибут;
- **Изображение** – позволяет загрузить изображение, характеризующее данный атрибут, например, рисунки стандартных изделий с размерами, каждый из которых описывается своим атрибутом;

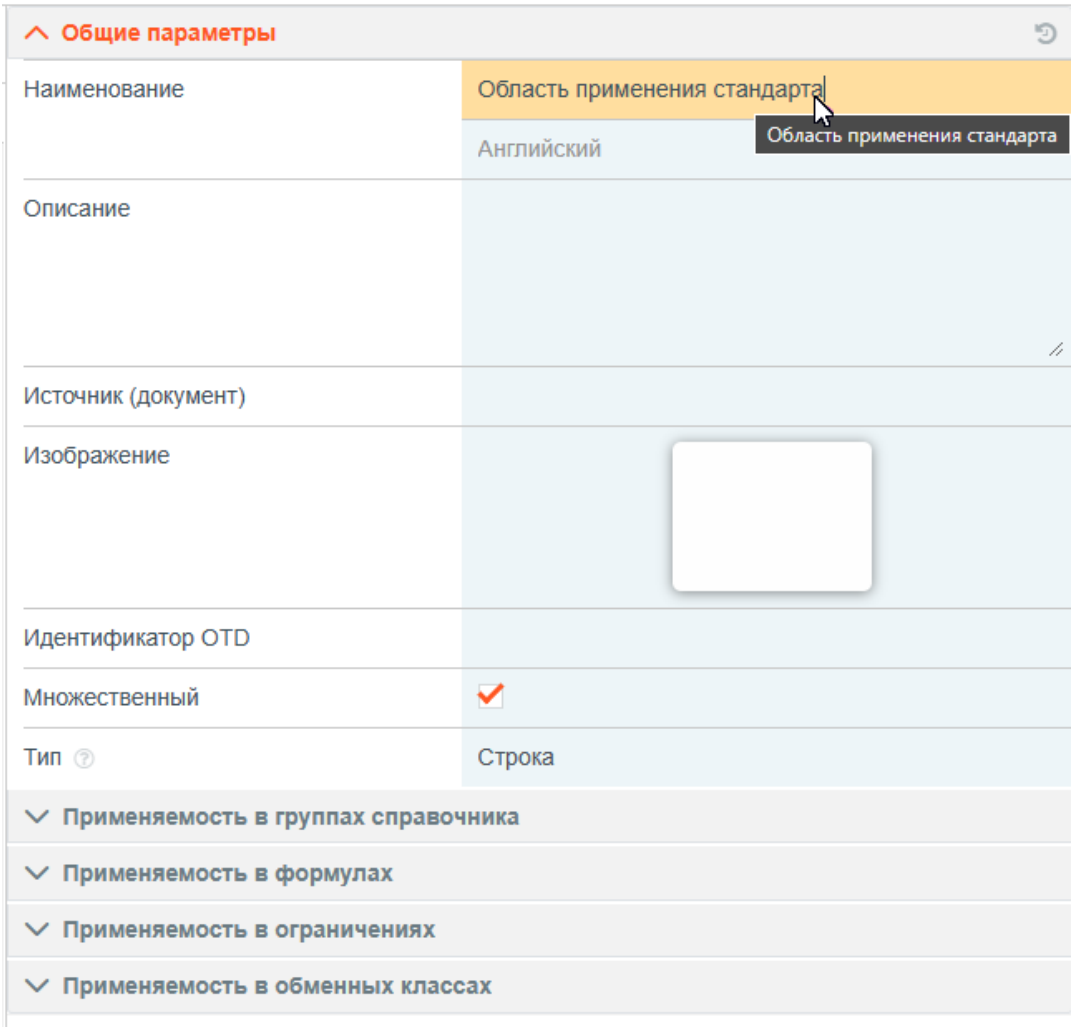


Рисунок 288 - Присвоение наименования атрибуту

- ячейка **Идентификатор ОТД** используется для задания идентификатора по словарю eOTD по ГОСТ Р ИСО/ТС 22745;
- установленный маркер **Множественный** будет означать, что данный атрибут для объекта может иметь несколько значений (Рисунок 289), последовательно указанных в виде списка . Кроме атрибутов типа – **Логический тип**, для которых массив не разрешён;

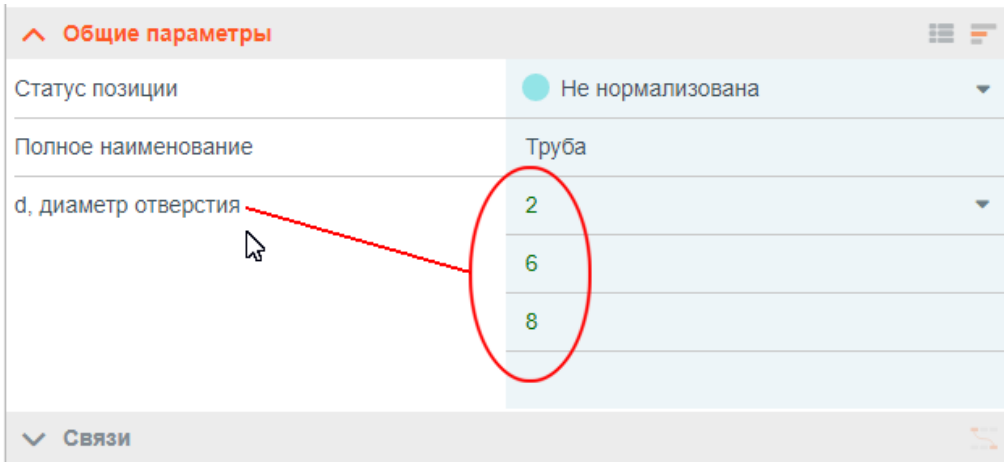


Рисунок 289 - Множественное значение для одного атрибута

- для атрибутов типа **Число с плавающей точкой**, **Целое число**, **Дата** и **Цвет** при установке маркера **Множественный**, становится доступной опция **Контекстное значение** (Рисунок 290). Опция предусматривает, что множество значений атрибута будет зависеть от контекста. Настройка подобных значений описана в п. 3.16.8;

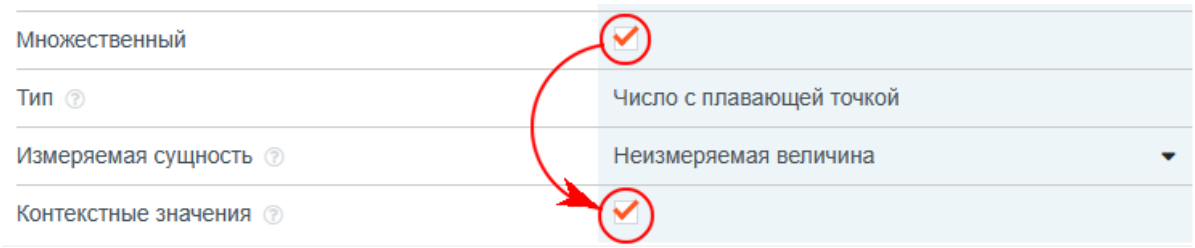


Рисунок 290 - Опция **Контекстные значения** для массива

3.16.4 Типы данных атрибутов

Каждый атрибут должен иметь определённый тип, выбор которого зависит от свойства, описываемого данным атрибутом. Для простых атрибутов *Системы* разрешены следующие типы (Таблица 21).

Таблица 21 – Типы атрибутов

Тип данных атрибута	Описание назначения
Строка	Строка длиной от 1 до 5000 символов. Количество символов может быть ограничено соответствующей настройкой в декларации атрибута

Число с плавающей точкой	Дробное число, для хранения которого выделяется 64 бита памяти (примерный диапазон значений [–1,844 ′ 10^{19}; 1,844 ′ 10^{19}]). Для чисел с плавающей точкой имеется возможность указывать единицу измерения и настраивать точность дробной части.
Целое число	Целое число, для хранения которого выделяется 32 бита памяти (диапазон значений [-2147483648; 2147483647])
Логический тип	Тип данных (boolean), которые могут принимать два возможных значения, называемых правдой и ложью (true / false). Предоставляет ячейку для установки маркера. Не может являться множественным!
Изображение	В атрибутах данного типа хранятся изображения объектов
Многострочный текст	Используется для хранения текста произвольной длины, обычно применяется для текстовых комментариев, содержащих от 1 до 10000 символов
Дата и время	Позволяет указать в качестве значения атрибута дату в формате <дд.мм.гггг>, а также время в формате <часы:минуты>
Цвет	Данный тип необходим для хранения информации о выбранном пользователем цвете. Используется для условной цветовой идентификации объектов на сетке данных.
Файл	Позволяет присоединять к объекту файлы, как текстовых, так и графических документов, любых форматов
Формула	Позволяет создать шаблон, по которому будет производиться вычисление значения атрибута. Шаблон для позиций должен создаваться на уровне родительской группы.

Настройка вывода значений по всем типам атрибутов производится в той справочной группе, где они используются.

3.16.5 Создание и настройка атрибута связи. Агрегация и композиция

Атрибут связи представляет собой особый вид атрибута, который обеспечивает возможность использования значений атрибута одного справочника в виде значений атрибута другого справочника, то есть ссылочное значение. Использование связей подобного рода (называемых агрегацией) обеспечивает сокращение числа вводимых однотипных данных и единство применяемых значений. Например, в справочной группе с крепежом, материал для крепёжных изделий можно вводить вручную, а можно обратиться к справочнику материалов. В первом случае неизбежно возникновение ошибок в записи марок и стандартов, а во втором все записи выверяются экспертом НСИ, отвечающим за справочник материалов.

Для создания атрибута связи следует:

- установить курсор на группе, в которую добавляется атрибут, и перейти в таблицу данных;
- в таблице данных нажать кнопку ***Добавить*** или вызвать контекстное меню и выбрать команду **Добавить атрибут связи**;
- перейти в раздел **Общие параметры** (Рисунок 291) и задать наименование атрибута в соответствующей ячейке. Наименование на английском языке присваивается по необходимости;
- **Наименование обратной связи** - наименование атрибута, которое будет отображаться в связанном справочнике. Может быть не заполненным;

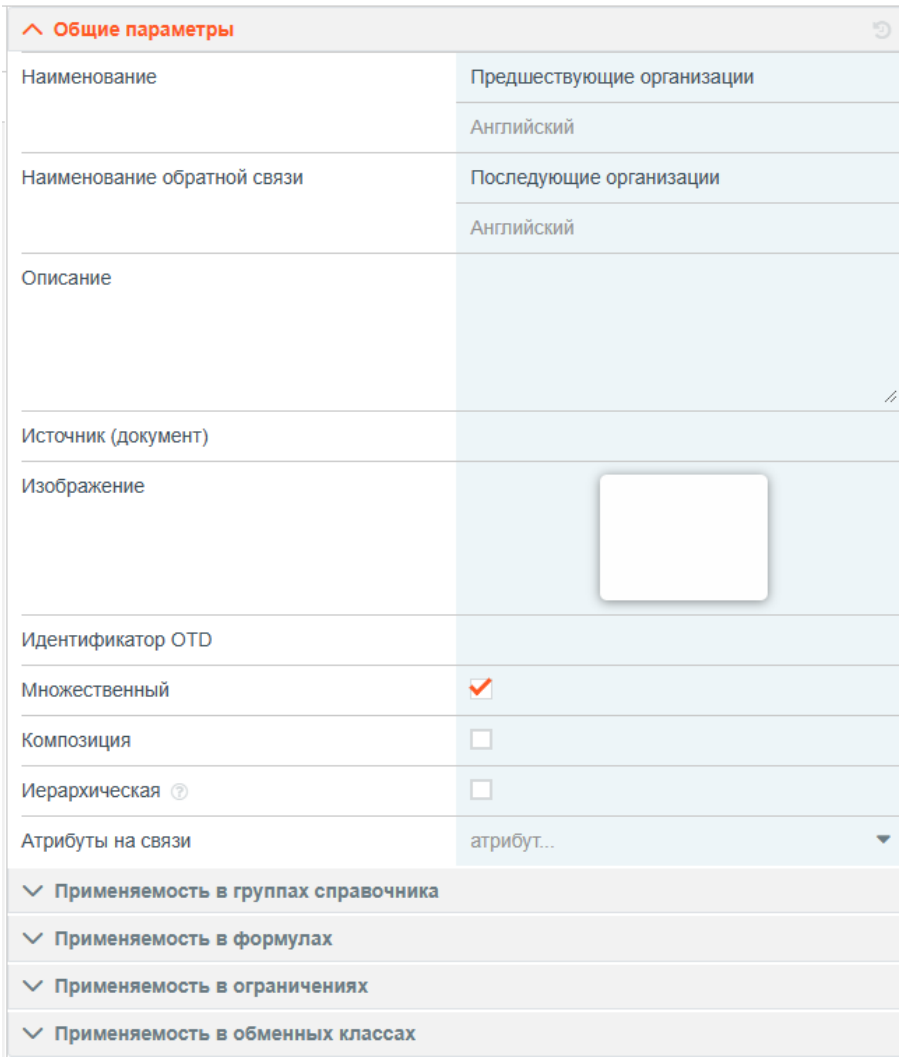


Рисунок 291 - Создание атрибута связи

в ячейке **Описание** можно дать комментарий о назначении атрибута и его использовании (на усмотрение пользователя). Данное описание будет находится во всплывающем сообщении при наведении курсора на пиктограмму рядом с атрибутом в режиме редактирования в карточке справочной группы (Рисунок 292);

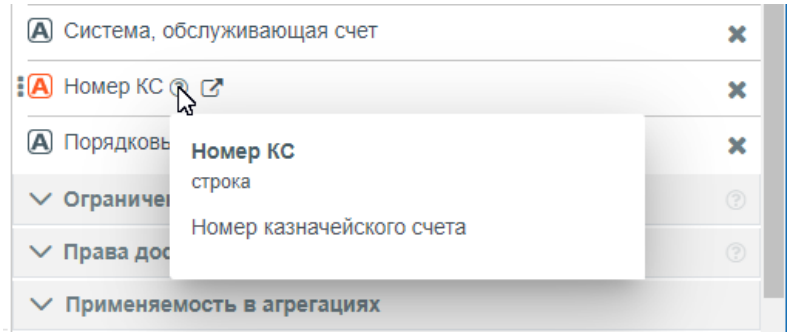


Рисунок 292 - Всплывающее окно с описанием атрибута и его типа данных

- **Источник (документ)** – в ячейке рекомендуется указывать документ (справочник, стандарт и т.п.), согласно которому введён данный атрибут;
- **Изображение** – позволяет загрузить изображение, характеризующее данный атрибут, например, рисунки стандартных изделий с размерами, каждый из которых описывается своим атрибутом;
- ячейка **Идентификатор ОТД** используется для задания идентификатора по словарю eOTD по ГОСТ Р ИСО/ТС 22745;
- установленный маркер **Множественный** будет означать, что данный атрибут для объекта может иметь не одно значение, а несколько, последовательно указанных в виде списка;
- маркер **Композиция** обеспечивает возникновение связи «композирующая» позиция – «композируемая» позиция (типа «родитель» – «дочерний объект»), причём, если удаляется родительский объект, то автоматически удаляются и все дочерние. Таким образом, если будет удалён объект, на значение атрибута которого ссылается текущий объект, то он также будет удалён;

- маркер **Иерархическая** применяется для моделирования организационной структуры (см. п. 3.16.6) подразделений предприятий, структуры холдинга или состава изделий;
- в ячейке **Атрибуты на связи** необходимо выбрать из выпадающего списка атрибут, который описывает агрегацию. Например, можно ввести атрибут связи со справочником *Логистические ЕИ* и добавлять в данный атрибут коэффициент пересчета из базовой ЕИ в логистическую, причем для каждой позиции коэффициент может быть свой. Допускается выбор нескольких атрибутов или значения могут быть не выбраны.

На этом создание атрибута заканчивается, и его дальнейшая настройка выполняется на уровне справочной группы.

Настройка атрибута связи в справочной группе в основном повторяет настройки простых атрибутов, кроме элемента **Область**, где, используя кнопку ***Добавить***, следует указать справочные группы, ссылки на объекты которых могут использоваться в качестве значения атрибута (Рисунок 293).

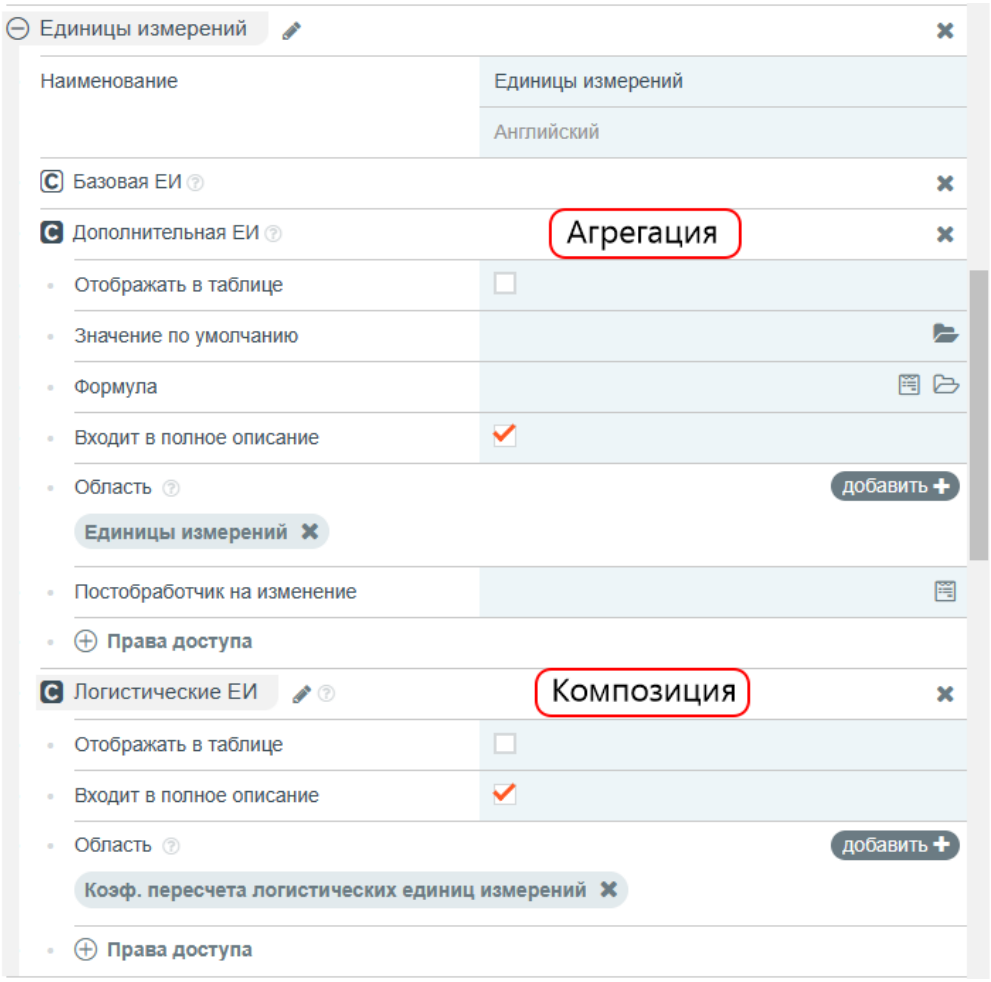


Рисунок 293 - Настройка атрибута связи в справочной группе

Для атрибута связи типа «множественный» в карточке группы подсвечивается ее наименование. Подсветка настраивается системным администратором, но может быть изменена в справочной группе.

При редактировании позиции с атрибутом связи типа *Композиция*, пользователю доступно табличное представление, вызываемое кнопкой ***Показать таблицу*** (Рисунок 294).

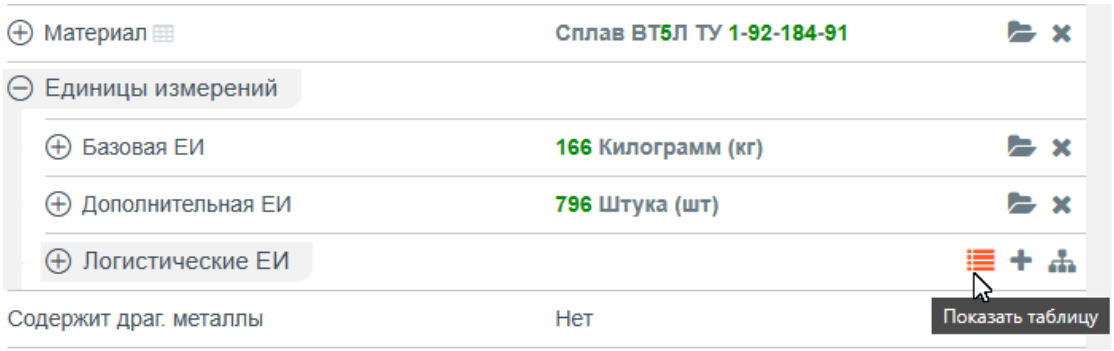


Рисунок 294 - Доступ к табличному представлению композируемого справочника

В табличном представлении (Рисунок 295) пользователю доступен для работы весь функционал, связанный с созданием и редактированием позиций, а также поиском и фильтрацией.

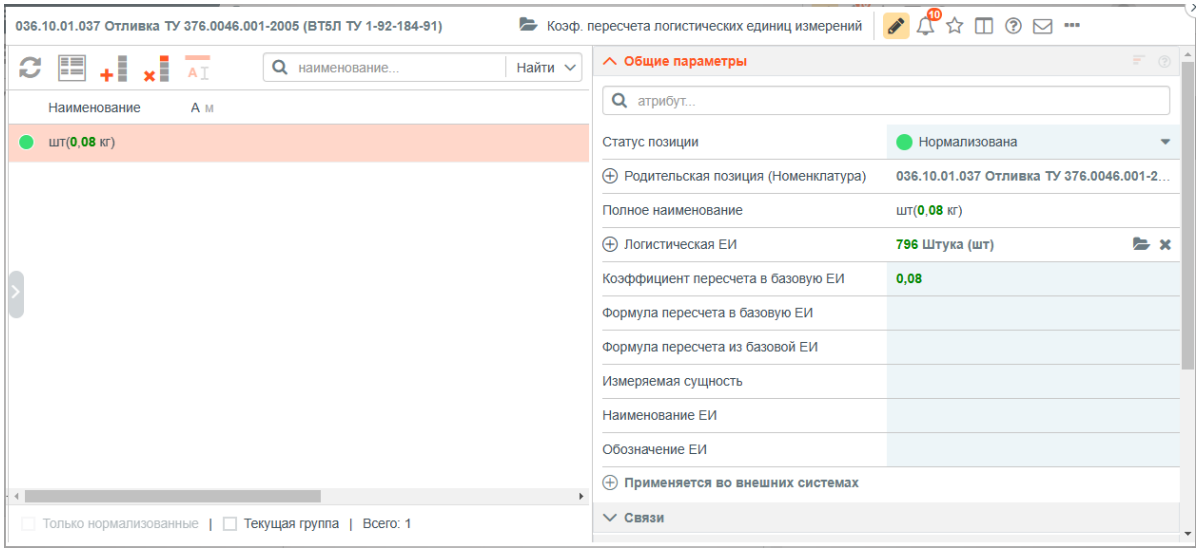


Рисунок 295 - Табличное представление композируемого справочника

3.16.6 Иерархические связи и создание организационной структуры

Понятие иерархической связи используется для атрибутов связи, которые могут использоваться для моделирования организационной структуры подразделений предприятий, структуры холдинга или состава изделий. Для определения иерархической связи в карточке атрибута связи присутствует маркер **Иерархическая** (Рисунок 296).

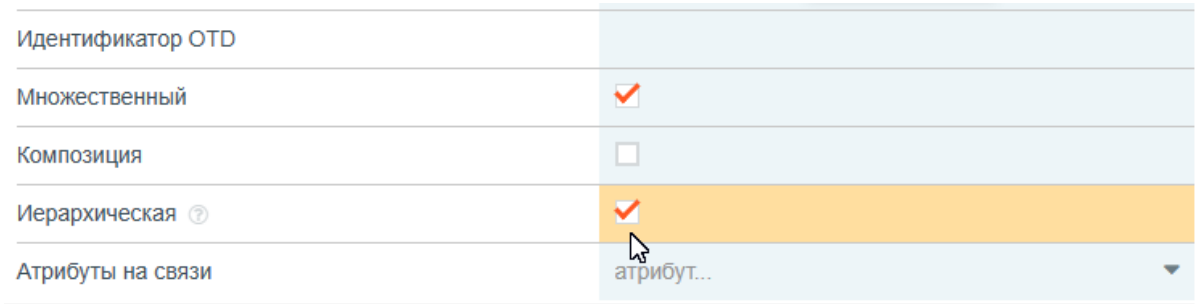


Рисунок 296 - Маркер **Иерархическая** для атрибута связи

Установка данного маркера возможна как для единичной агрегации, так и для множественной (массива). Отсутствие маркера **Множественный** означает, что иерархия позиций формируется через указание родительской позиции. Указание родительской позиции производится выбором из выпадающего списка в ячейке атрибута.

Установка маркера для множественной агрегации будет означать, что иерархия позиций формируется через указание списка подчинённых позиций. Список подчиненных позиций может быть сформирован их последовательным выбором из выпадающего списка или при помощи вызова **Редактора связей**.

Открыть **Редактор связей** для атрибута агрегации можно как из карточки позиции, так и по кнопке над табличным представлением позиций. При открытии редактора связей из карточки позиции, иерархия будет построена от текущей позиции, отображённой в карточке. При открытии редактора иерархии связей по кнопке над таблицей, система найдёт все корневые позиции, после чего отобразит редактор иерархии связей, начиная от корневых позиций. **Редактор связей** (Рисунок 297) позволяет как просмотреть имеющуюся иерархию связей, так и дополнить её новыми позициями.

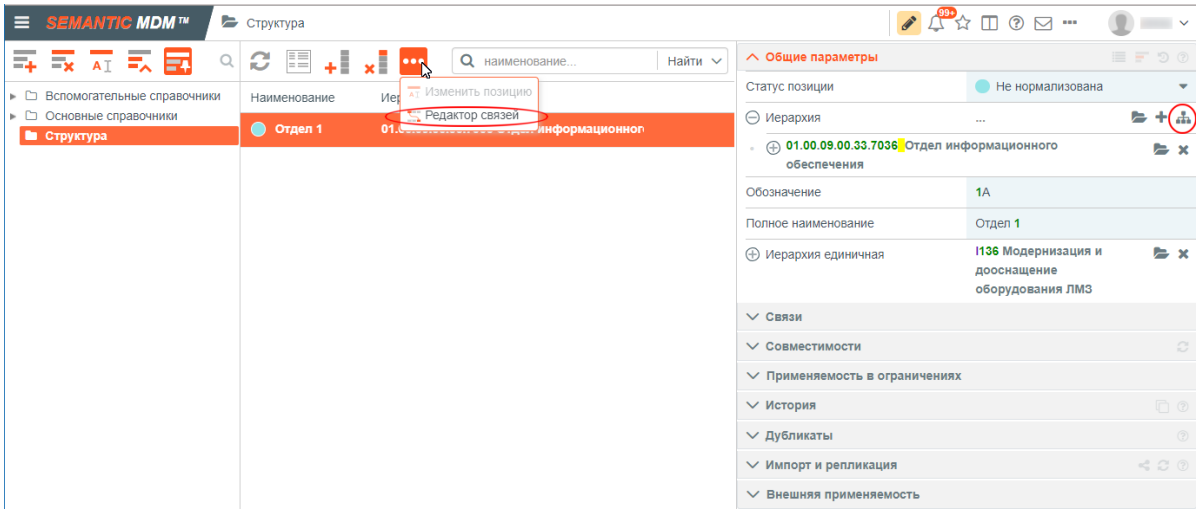


Рисунок 297 - Доступ к редактору связей

Добавление новой подчиненной позиции выполняется при помощи кнопки ***Добавить связь с позицией*** (Рисунок 298) над таблицей подчиненных позиций в **Редакторе связи**. Позиции выбираются из открывшегося связанного справочника.

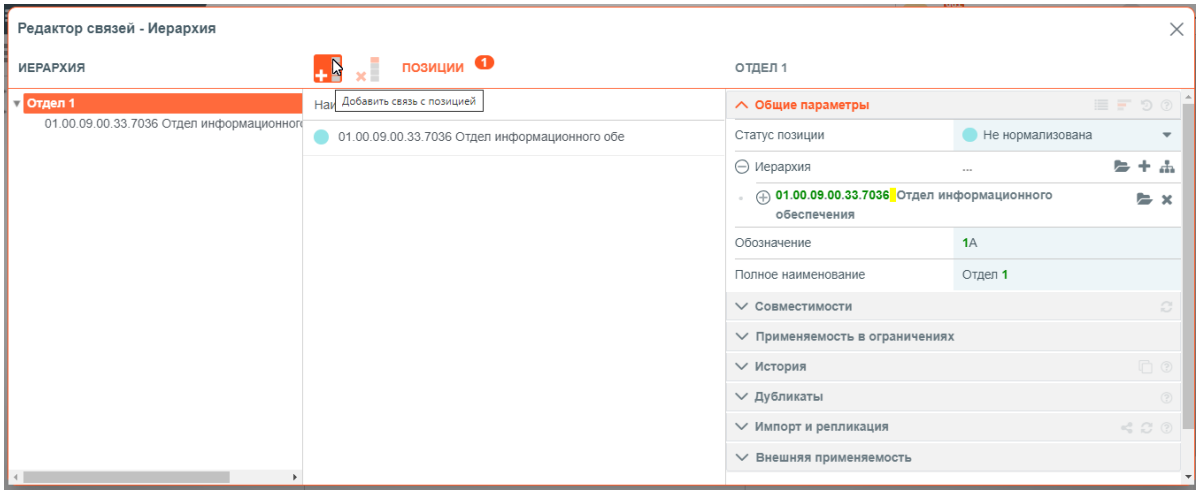


Рисунок 298 - Настройка организационной структуры через редактор связей

3.16.7 Настройка атрибута типа «Число с плавающей точкой»

Настройка атрибута типа «Число с плавающей точкой» включает в себя указание точности и выбор единицы измерения из выпадающего списка (Рисунок 299).

Точность указывается в виде количества цифр, показываемых после запятой. Значение вводится вручную.

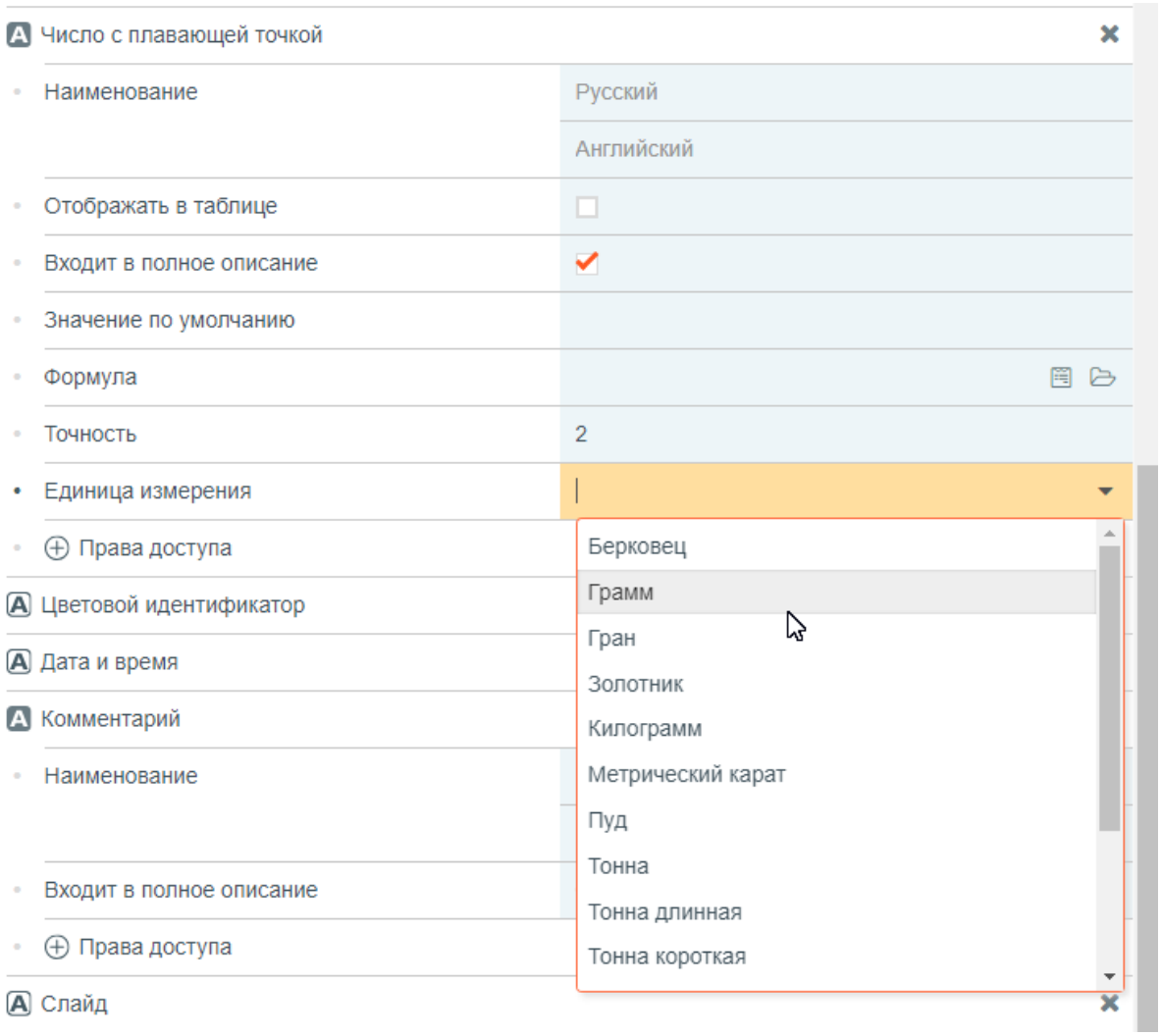


Рисунок 299 - Настройка точности и единицы измерения

Выбор единиц измерения зависит от измеряемой сущности, определённой на этапе создания атрибута. Измеряемая сущность – масса, сила, длина и т.д. выбирается из выпадающего списка (Рисунок 300). Содержание этого списка определяется системным справочником *Единицы измерения* (подробнее см. п. 3.21).

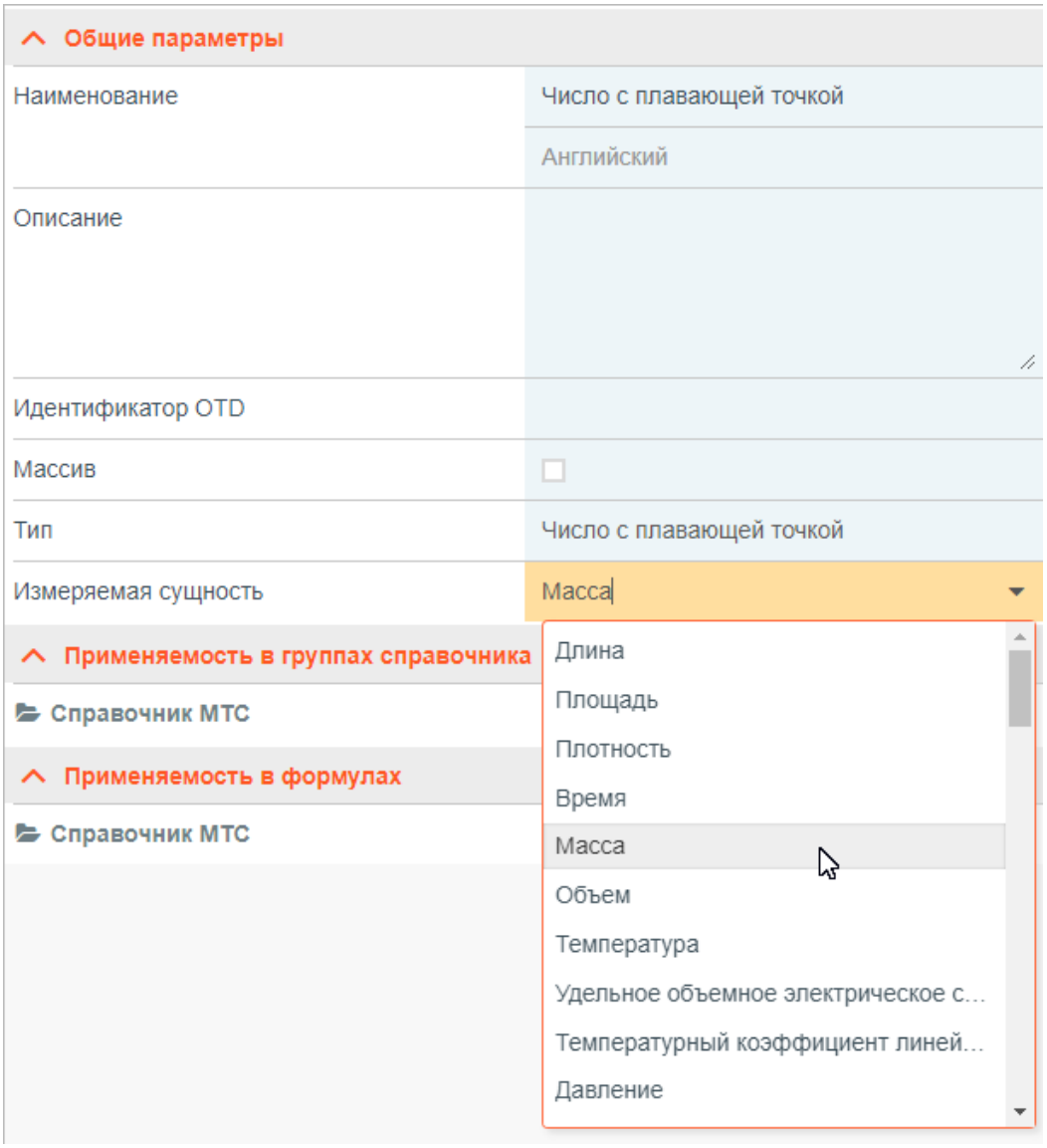


Рисунок 300 - Настройка измеряемой сущности

3.16.8 Настройка контекстных значений атрибута

Атрибуты типа **Число с плавающей точкой**, **Целое число**, **Дата** и **Цвет** могут иметь массив значений, которые в свою очередь будут зависеть от какого-либо контекста. Например, значение прочности сталей и сплавов зависит от температуры окружающей среды.

Указание контекста и заполнение массива значений атрибута выполняется в справочнике.

Значение атрибута представляется в виде таблицы, состоящей из двух колонок: первая – собственно контекст и вторая колонка со значениями.

Колонка с контекстом всегда принимает строковое значение (Рисунок 301).

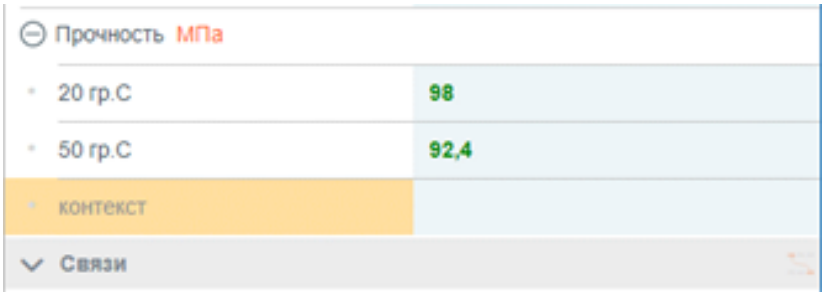


Рисунок 301 - Настройка массива значений с контекстом

3.16.9 Маска для значения атрибута

Маски атрибутов предназначены для ограничения количества и вида вводимых символов при заполнении данных. Они могут применяться для ввода шифров, кодов, номеров телефонов, условных обозначений и т.д.

Маска составляется из набора специализированных символов (регулярные выражения), значение которых представлено в таблице ниже, а также во всплывающем окне при наведении курсора на пиктограмму со знаком вопроса в ячейке.

Таблица 22. Маска для значения атрибутов

Символ	Значение
9	любая цифра
a (лат.)	любая буква
A (лат.)	любая буква с автоматическим приведением к верхнему регистру
*****	любая цифра или буква
&	любая цифра или буква с автоматическим приведением к верхнему регистру
#	любой символ числа в шестнадцатеричной записи
(символы)	круглые скобки позволяют группировать символы
[символы]	в квадратных скобках заключается массив символов, из которых пользователь может ввести только один символ
<символы>	в треугольных скобках заключается массив символов, ввод которых необязателен
|	размещается между другими символами, если из них выбирается один по условию ИЛИ. Допускается указывать 3 альтернативы сразу, например, (b|c|d), а также группы символов, например, (999)|(aa9)
{n}	повторить предыдущий символ (группу символов) n раз

{n,}	повторить предыдущий символ (группу символов) n раз или более раз
{n,m}	повторить предыдущий символ (группу символов) как минимум n раз, но не более m раз
{*}	повторить предыдущий символ (группу символов) произвольное количество раз, включая нулевое
{+}	повторить предыдущий символ (группу символов) 1 или более раз
\(символ)	любой из специальных символов как константа
остальные символы	константа

Ннже приведены примеры масок ввода.

Таблица 23. Примеры масок ввода

Пример маски ввода	Комментарий
Морзе №[1-6]	Ввод строки с обозначением конуса Морзе, например, *Морзе №2*
((л|Л)евый)|((п|П)равый)	Возможно два значения: *Левый* или *Правый*, причём начинать ввод значении можно как с прописной, так и со строчной буквы
9{4}-9{4}	Обозначение оснастки, например, *2624-0001*

Если у атрибута заполнено *Значение по умолчанию*, и оно не соответствует маске атрибута, то в в ячейке со значением будет показан специальная пиктограмма - восклицательный знак в красном круге.

3.16.10 Создание атрибутов из справочной группы

Порядок создания атрибутов непосредственно из справочной группы подчиняется тем же правилам, что и при создании их непосредственно в конфигураторе.

Если при выборе атрибута для группы не был найден подходящий, то для создания нового следует воспользоваться кнопками, расположенными над списком атрибутов (Рисунок 302).

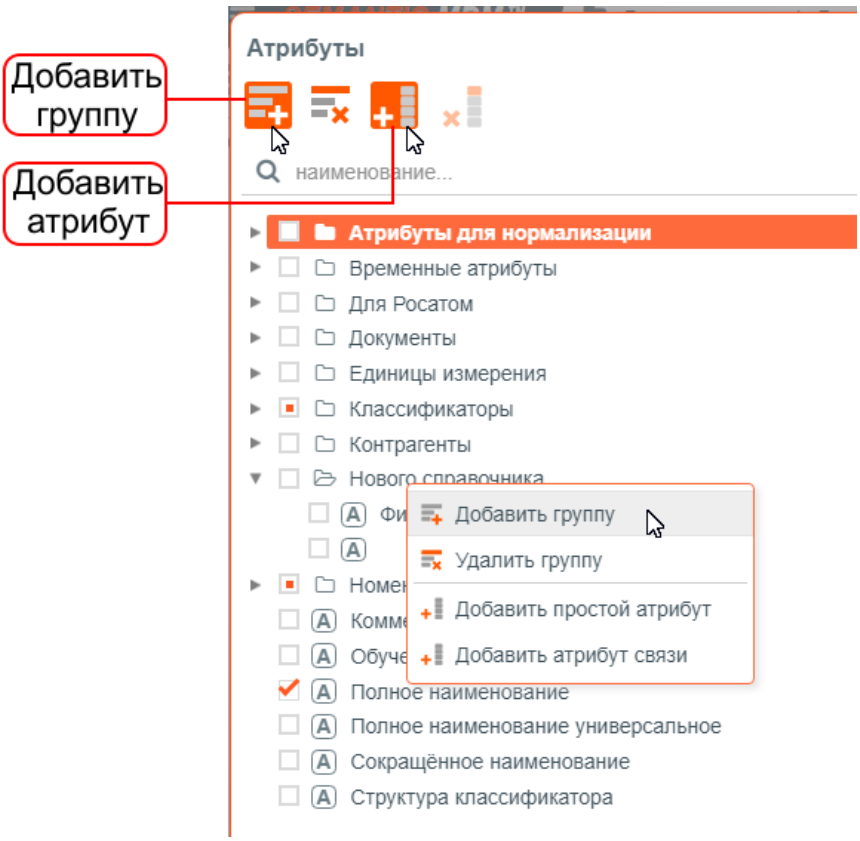


Рисунок 302 - Создание атрибута из справочника

Команды **Добавить группу**, **Добавить простой атрибут** и **Добавить атрибут связи** могут быть вызваны из контекстного меню окна **Атрибуты**. Настройка атрибутов производится в карточке атрибута в правой части окна. Все процедуры настройки и применения описаны в пунктах этого раздела.

3.16.11Загрузка атрибутов в Классификатор

Для увеличения скорости работы с атрибутами, особенно в случаях, когда они заимствуются из других информационных систем, можно воспользоваться процедурами экспорта / импорта Excel. Таблица с атрибутами может быть выгружена из **Классификатора атрибутов** и доработана в табличном редакторе *Excel*, а затем загружена обратно (Рисунок 303).

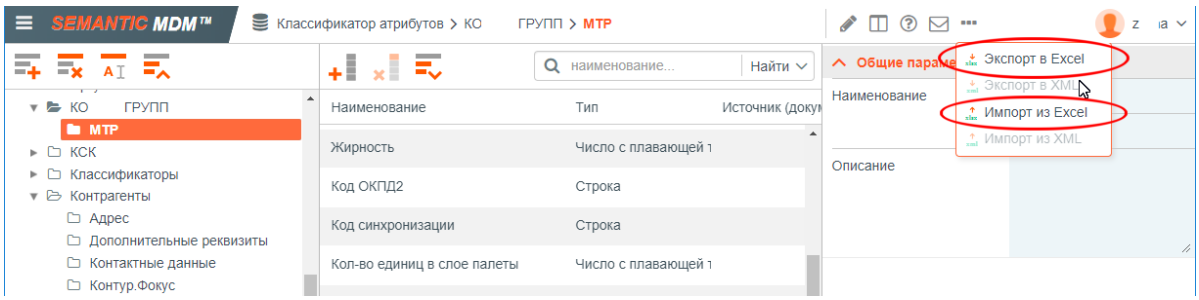


Рисунок 303 - Вызов команд экспорта и импорта в формат Excel

При выборе команды **Экспорт в Excel** автоматически формируется файл формата *.xlsx, имя которого соответствует названию ветки **Классификатора атрибутов**, из которой они выгружаются, и сохраняется в указанное пользователем место. При открытии файла в нем доступны для редактирования и заполнения столбцы, соответствующие ячейкам карточки атрибута (Рисунок 304).

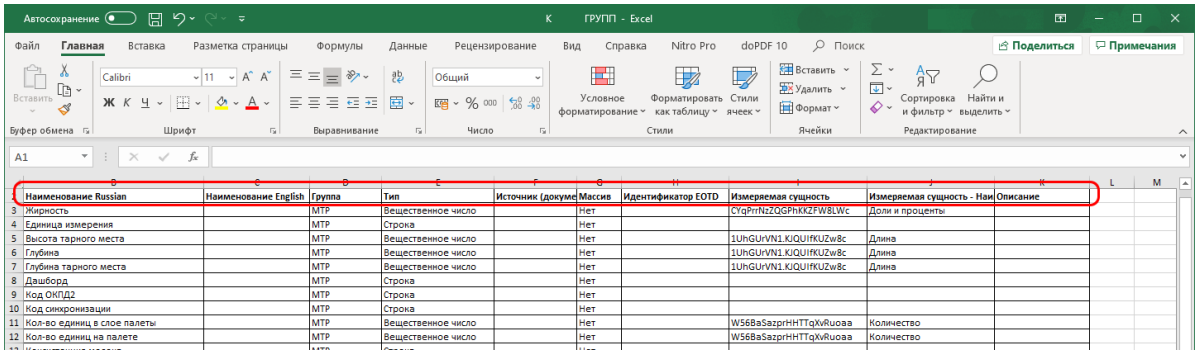


Рисунок 304 - Ветка **Классификатора атрибутов**, выгруженная в Excel

При загрузке атрибутов из файла Excel они передаются в **Классификатор** автоматически, если наименование столбцов соответствует наименованию ячеек карточки атрибута. В противном случае выводится сообщение об ошибке.

3.16.12 Дедубликация атрибутов

При вводе новых атрибутов регулярно возникает ситуация, что в **Классификаторе атрибутов** уже существует аналогичный атрибут. Это вызывает определённые сложности при настройке справочников и приводит к

тому, что одна и та же характеристика объекта НСИ в разных ветках описывается разными атрибутами.

Для универсализации списка используемых атрибутов и устранения дублирующих используется функционал дедубликации, который позволяет выполнить дедубликацию атрибутов с разрешением возможных конфликтов. Например, когда оба атрибута были использованы в одной позиции или в ограничительной таблице или одной формуле.

Дедубликация атрибутов выполняется в рамках отдельного пакета изменений (создаётся автоматически при наличии заполненных значениях, если хотя бы один из атрибутов имеет значение, был продекларирован в группе, использован в ограничениях или формуле), что позволяет увидеть, как дедубликация атрибутов отразилась на данных. При публикации пакета изменений, выполняется фиксация изменений, выполненных в данных, в результате дедубликации атрибутов. При дедубликации атрибутов не имеющих заполненных значений и нигде не используемых, слияние происходит без создания пакета изменений.

Чтобы выполнить дедубликацию атрибутов в **Классификаторе атрибутов** следует:

- найти требуемый объект и установить на нем курсор в таблице;
- вызвать из контекстного меню команду **Признать дубликатом** (Рисунок 305);

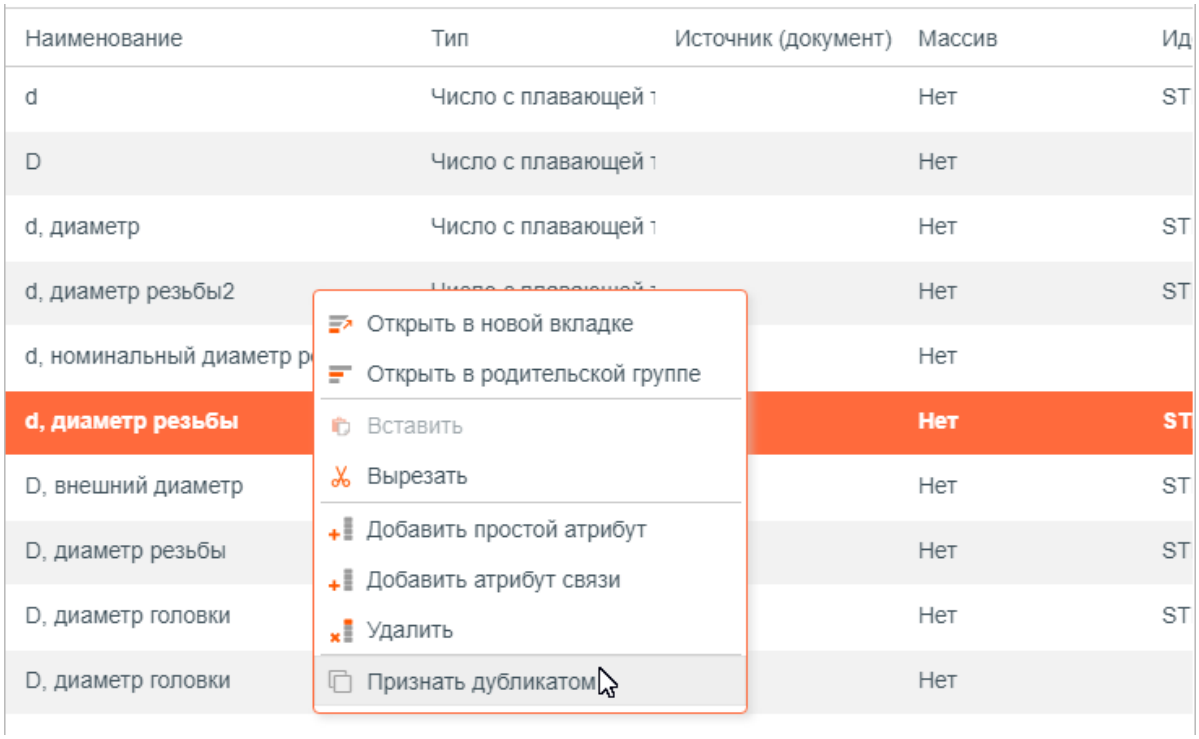


Рисунок 305 - Вызов команды **Признать дубликатом** в таблице

- в открывшемся окне **Атрибуты** (Рисунок 306) выбрать из списка атрибут-оригинал и нажать кнопку ***Ок***;

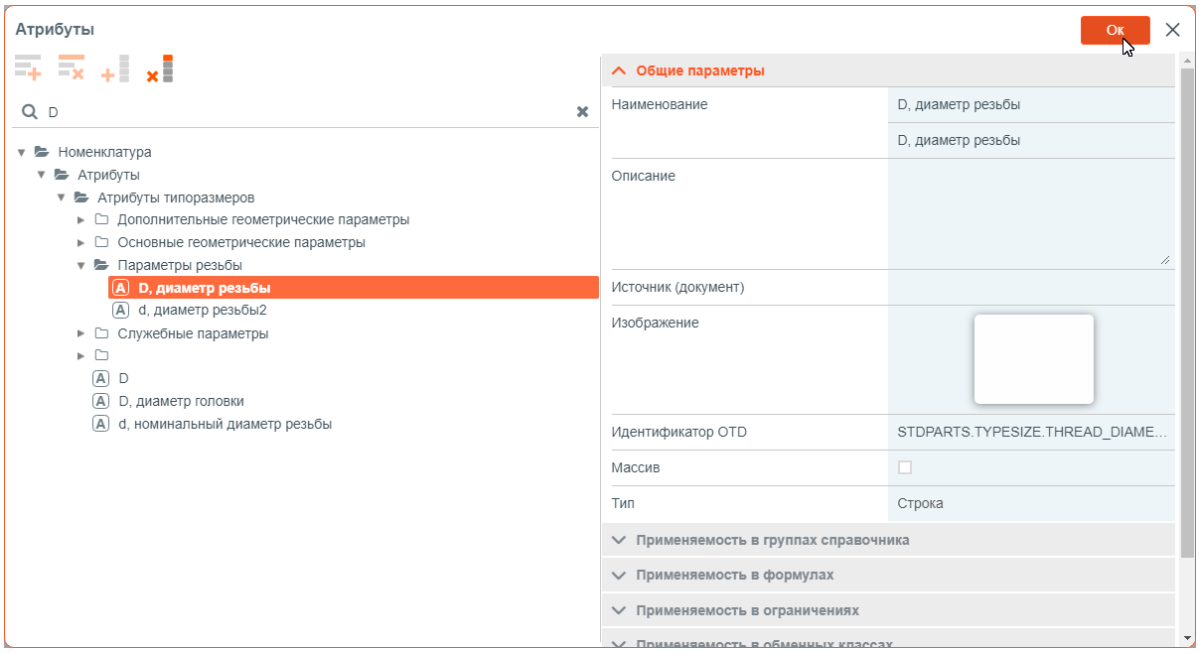


Рисунок 306 - Окно Атрибуты для выбора атрибута-оригинала

- в окне **Слияние атрибутов** (Рисунок 307) на вкладке **Атрибуты** показаны карточки, которые позволяют провести сравнение перед слиянием. Установленный маркер только различающиеся позволит

видеть только те характеристики атрибутов, которые имеют отличия. В разделе **Предупреждения** появляются записи, если атрибут был использован в обменном классе, или ограничительной функции, а также в формулах вычисления. В этом случае, конфликт не может быть разрешён автоматически и требуется его устранить до выполнения дедубликации;

- атрибут-оригинал отмечается обозначается значком , а дубликаты значком . Чтобы поменять местами оригинал и дубликат достаточно щёлкнуть левой кнопкой мыши по значку;

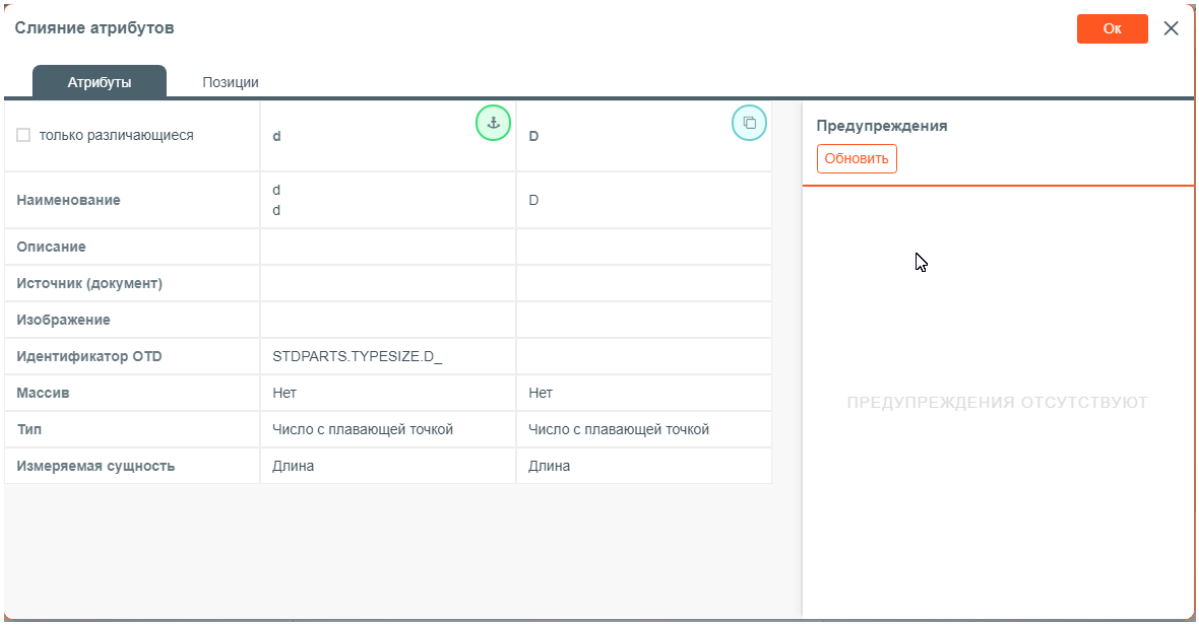


Рисунок 307 - Сравнение атрибутов перед их слиянием

- на вкладке **Позиции** (Рисунок 308) окна **Слияние атрибутов** можно посмотреть на наличие конфликтующих значений в позициях (маркер только конфликтующие), а также выбрать двойным щелчком левой кнопки мыши нужные значения (они будут подсвечены светло-зелёным цветом, как на рисунке ниже);

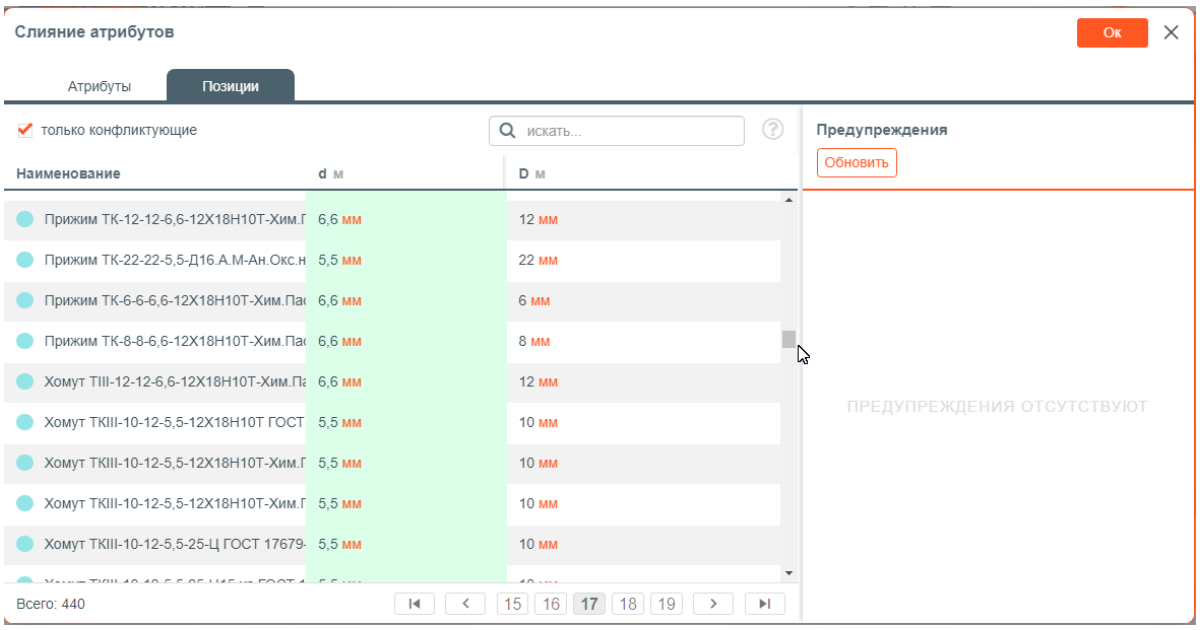


Рисунок 308 - Сравнение атрибутов перед их слиянием

- после всех необходимых процедур в окне Слияние атрибутов следует нажат кнопку ***Ок***. После чего *Система* выполнит дедубликацию атрибутов, а сведения о проведённых изменениях будут отражены в отдельном пакете изменений, что позволит опубликовать данные или отказаться от выполненных действий.

3.17 Формулы для вычисления значений и обогащения данных

Для вычисления значения любого атрибута [19] либо обогащения позиции при её создании или редактировании, а также везде, где система предполагает применение различных дополнительных алгоритмов для работы с данными можно использовать формулы, которые представляют собой либо шаблон, либо скрипт Groovy, либо являются композицией того и другого сразу.

В том случае, если пользователь предполагает использовать одну общую формулу для позиций, входящих в разные справочные группы, её следует настраивать для родительской группы, в которую входят все эти справочные группы. Для этого используются атрибуты типа «Формула». Например, при

[19] *Атрибуты типов «Файл», «Формула», «Многострочный текст» и «Слайд» не имеют настройки собственных формул.*

вычислении полного наименования стандартного изделия на основе данных нормативного документа, шаблон с формулой вычисления наименования следует расположить в этом НД в справочнике стандартов.

Процедура настройки шаблона одинакова для всех типов атрибутов и не зависит от места расположения шаблона в иерархии справочных групп. Атрибуты, значения которых будут использоваться в формуле, должны быть созданы, продекларированы в группе и настроены заранее.

Доступ к формулам осуществляется через одноименную ячейку, находящуюся в описании атрибута (Рисунок 309).

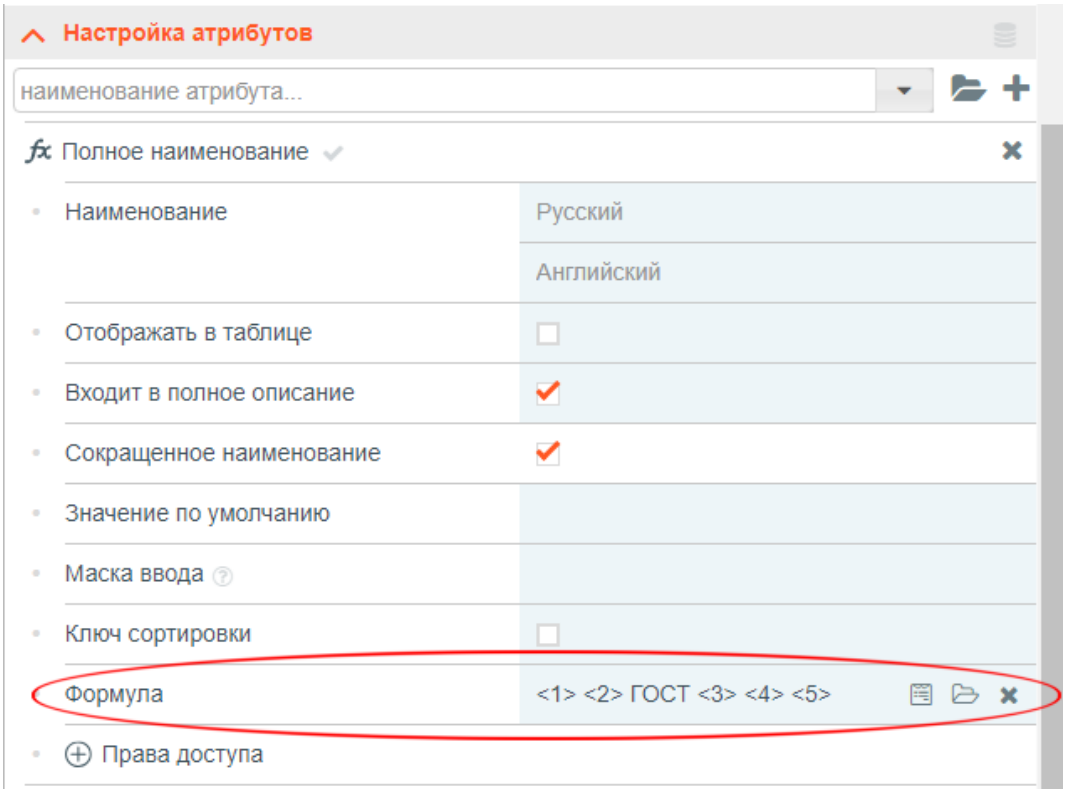


Рисунок 309 - Расположение ячейки с формулой

Для работы с формулами в ячейке присутствуют специальные кнопки: **Редактор формул**, **Ссылка на формулу** и **Удалить формулу** (доступна, если формула уже создана).

Кнопка **Ссылка на формулу** (Рисунок 310) открывает одноименное окно и позволяет выбрать атрибут типа «формула», располагающийся в родительской группе или агрегированном (связанном) справочнике.

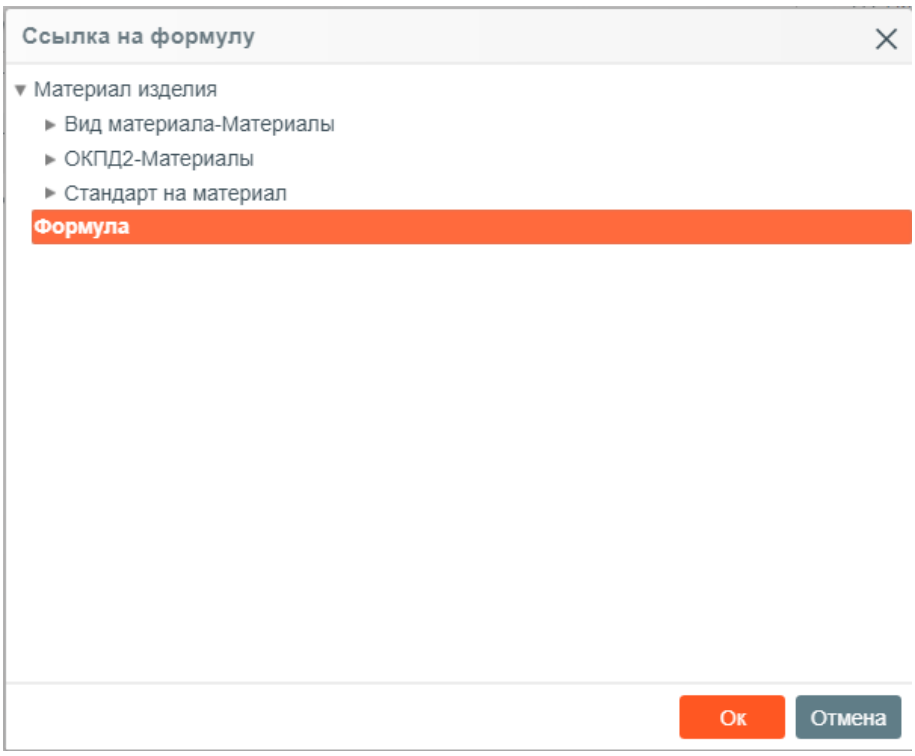


Рисунок 310 - Выбор формулы по ссылке

Кнопка ***Удалить формулу*** позволяет удалить формулу у текущего атрибута. Если удаляется формула, являющаяся шаблоном для других атрибутов (добавлена в виде ссылки), то значения, вычисленные по этой формуле, исчезнут (станут пустыми).

3.17.1 Описание редактора формул

Разработка формулы (шаблона) ведётся в **Редакторе формул** (Рисунок 311), который вызывается нажатием на одноименную кнопку в ячейке. Редактор представляет собой всплывающее окно, в котором в левой части располагается список атрибутов  и описание методов  (для скрипта GROOVY), которые могут быть использованы в формуле, а в правой части представлен шаблон строки, ниже которого расположены вкладки:

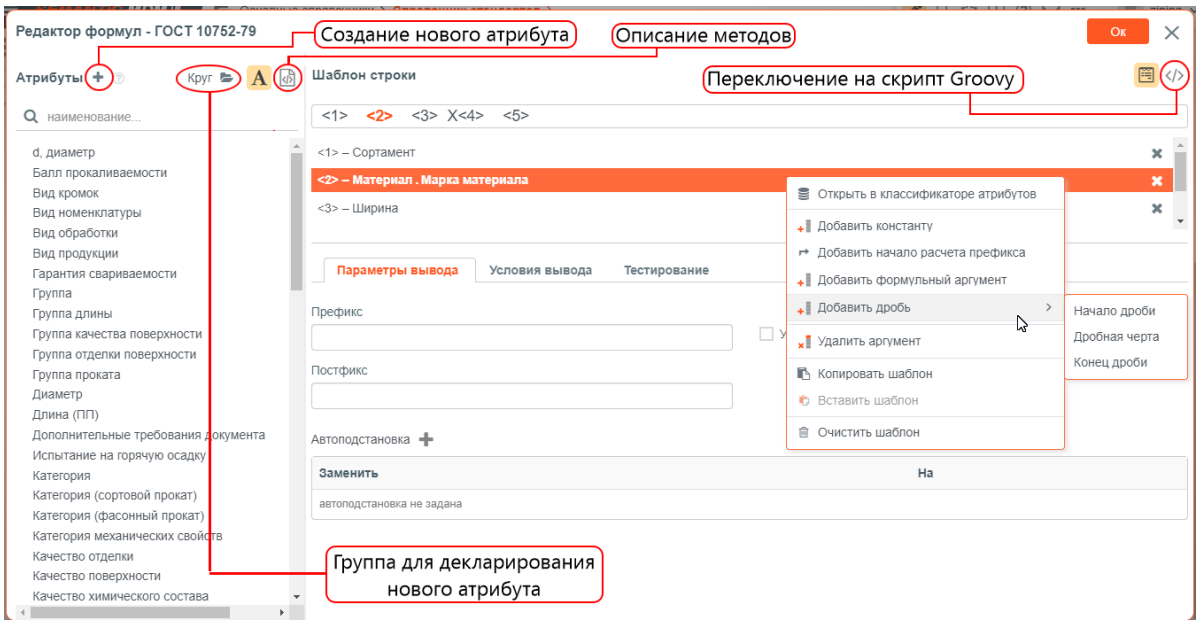


Рисунок 311 - Редактор формул в виде шаблона строки

- **Параметры вывода** – содержит ячейки для настройки префикса и постфикса, используемых при выводе атрибута, а также поле автоподстановки, которое обеспечивает замену значения атрибута на другое. Например, если у крепёжной детали есть признак Левая резьба, заменить его в шаблоне на LH;
- **Условия вывода** – содержит таблицу с условиями использования значения атрибута в шаблоне. Например, не указывать шаг резьбы в обозначении крепежа, если это значение соответствует мелкому шагу.
- **Тестирование** – позволяет выбрать любую позицию из справочной группы для проверки работы шаблона.

Атрибут, необходимый для разработки формулы, может быть создан непосредственно из **Редактора формул**. Для этого используется кнопка ***Добавить атрибуты***, расположенная над списком атрибутов. Если формула создается непосредственно в декларации атрибута, то новый атрибут будет продекларирован в той же группе. Если формула создается в шаблоне вычисления (атрибут типа «формула»), то для создания нового атрибута в начале следует выбрать справочную группу, в которой он будет продекларирован.

При разработке формулы в виде шаблона строки в качестве элемента шаблона могут использоваться дополнительные элементы, создание которых вызывается из контекстного меню в строке шаблона:

- **Добавить константу** - создает поле, в котором могут быть указаны любые символы, которые будут выводится в строке в неизменном виде;
- **Добавить начало расчета префикса** - создает поле, в котором могут быть указаны любые символы, которые будут использоваться в качестве префикса последующего атрибута;
- **Добавить формульный аргумент** - создает поле, в котором может быть разработан скрипт Groovy, использующийся в качестве одного из элементов шаблона строки.

Редактор формул имеет два представления – в виде шаблона строки и в виде скрипта Groovy (Рисунок 312), причём, скрипт формируется автоматически при создании шаблона строки и может быть отредактирован при необходимости. Кроме того, формула может сразу создаваться в виде скрипта. Для работы с текстом скрипта пользователю доступны следующие функции:

- Описание скриптовых методов и функций с возможностью их вставки в текст скрипта с синтаксисом и комментариями;
- Выбор функции, переменной или константы из выпадающего списка по нажатию <*Ctrl*>+<*Пробел*>;
- Поиск по коду с возможностью использования регулярных выражений по нажатию <*Ctrl*>+<*F*>;
- Быстрое комментирование / раскоментирование строк или группы строк по нажатию <*Ctrl*>+</>;
- Удаление строки целиком по <*Ctrl*>+<*D*>;
- Автоматическая установка правильного отступа, при переходе на новую строку;
- Подсвечивание парных скобок.

Примеры различных скриптов, применяемых в формулах для вычисления значений атрибутов приведены в п. 3.33.

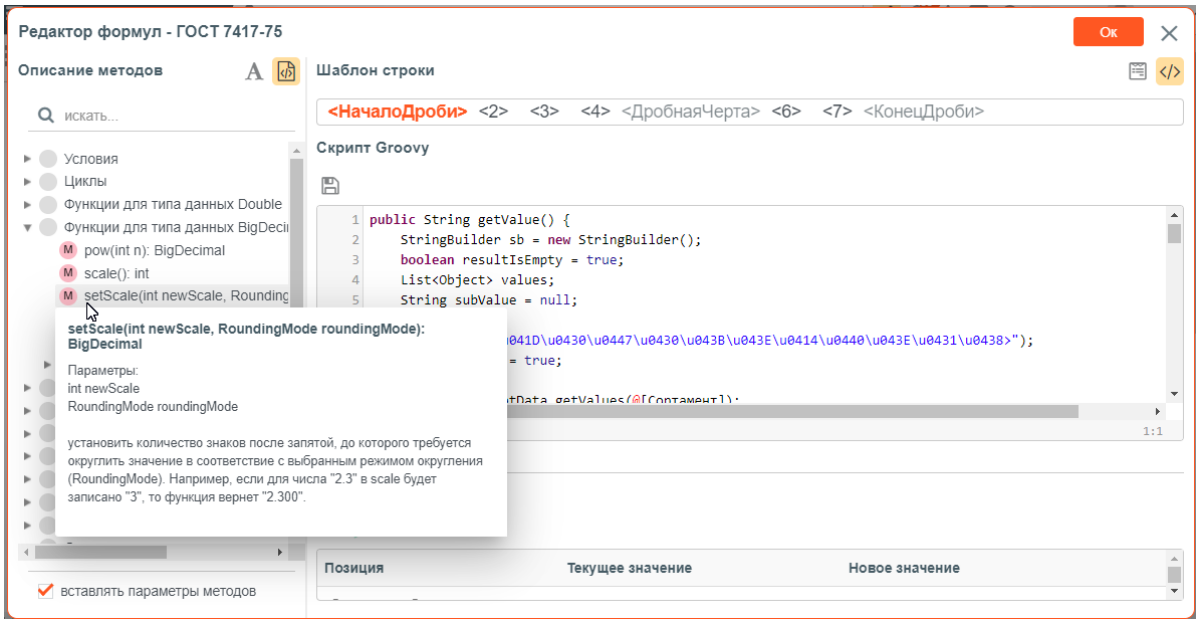


Рисунок 312 - **Редактор формул** с формулой в виде скрипта Groovy

Созданный шаблон может быть скопирован в буфер обмена для переноса в другой атрибут. Для этого используются команды контекстного меню **Копировать шаблон / Вставить шаблон**.

Для быстрого удаления всех аргументов, констант и атрибутов в шаблоне можно использовать команду контекстного меню **Очистить шаблон**.

3.17.2 Создание шаблона строки

Добавление атрибута в формулу осуществляется путём перетаскивания атрибута из списка слева в шаблон строки, где он получает порядковый номер в треугольных скобках, например, <1>. Если позицию атрибута необходимо изменить в записи, то её следует перетащить, используя функцию Drag&Drop непосредственно в шаблоне строки.

Ниже шаблона строки даётся легенда, поясняющая какому атрибуту, какой номер соответствует. Наименование атрибута также доступно во всплывающей подсказке (hint) при наведении курсора на номер в строке с шаблоном.

Кроме атрибутов в шаблоне строки могут быть использованы константы – значения, которые присутствуют в записи независимо от остального содержимого. Это могут быть служебные символы типа «-», «/» либо слова и аббревиатуры. Добавление констант осуществляется через контекстное меню под правой кнопкой мыши.

3.17.3 Параметры вывода

Настройки параметров вывода (Рисунок 313) одинаковы для всех типов атрибутов, кроме атрибутов с типом «число с плавающей точкой», для которых есть дополнительные настройки.

В ячейке **Префикс** указывается символ, слово, число или пробел, которые будут вписываться перед значением атрибута в строке. Тоже самое может быть указано в ячейке **Постфикс**, но в этом случае указанное будет записано после значения атрибута.

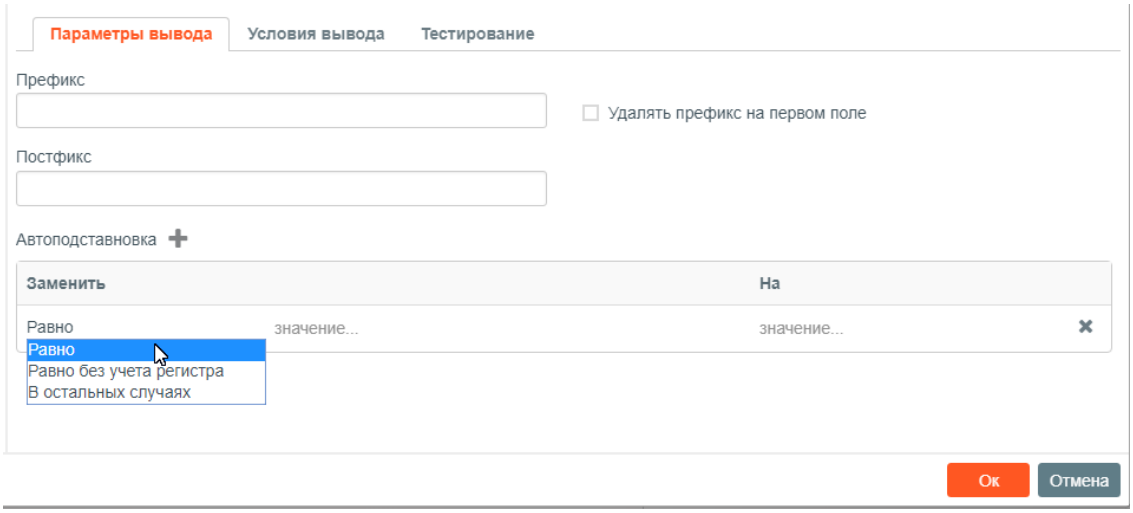


Рисунок 313 - Параметры вывода

Маркер **Удалять префикс на первом поле** позволяет убрать символы префикса, если значение атрибута в записи окажется на первом месте в случае отсутствия значений других атрибутов.

Для заполнения автоподстановки необходимо нажать кнопку ✚, и, в появившейся строке, вписать условия её выполнения. В качестве условий может быть выбрано:

- **Равно** – полное соответствие указанному значению;
- **Равно без учёта регистра** – соответствие указанному значению вне зависимости от регистра в записи значения. Например, *левая* резьба и *Левая* резьба будут учтены одинаково;
- **В остальных случаях** – указывается значение, принимаемое в записи, если не соблюдаются остальные условия автоподстановки.

Количество условий автоподстановки не ограничено.

Для атрибутов типа «число с плавающей точкой» имеются дополнительные настройки параметров вывода (Рисунок 314).

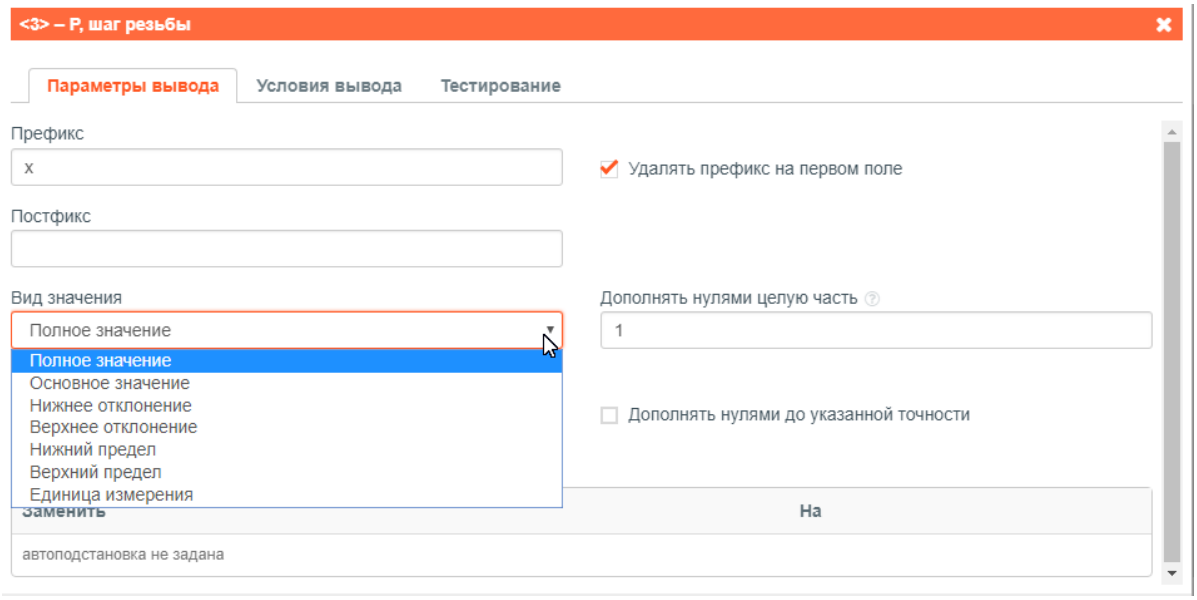


Рисунок 314 - Параметры вывода для числа с плавающей точкой

В ячейке **Вид значения** можно выбрать какую именно величину требуется указывать в шаблоне.

Дополнять нулями целую часть – позволяет указать количество нулей, дописываемых перед значением числа (для кодов и обозначений).

Маркер **Дополнять нулями до указанной точности** – обеспечивает запись нулей после запятой до указанной при настройке атрибута точности.

Маркер **Выводить обозначение ЕИ** - позволяет вывести в строке после значения условное обозначение ЕИ, заданное в справочнике *Единицы измерения*.

3.17.4 Условия вывода

На вкладке настраиваются условия вывода в шаблон текущего атрибута. Если условий несколько, то они могут использоваться совместно (значение И) либо одно из них (значение ИЛИ) (Рисунок 315).

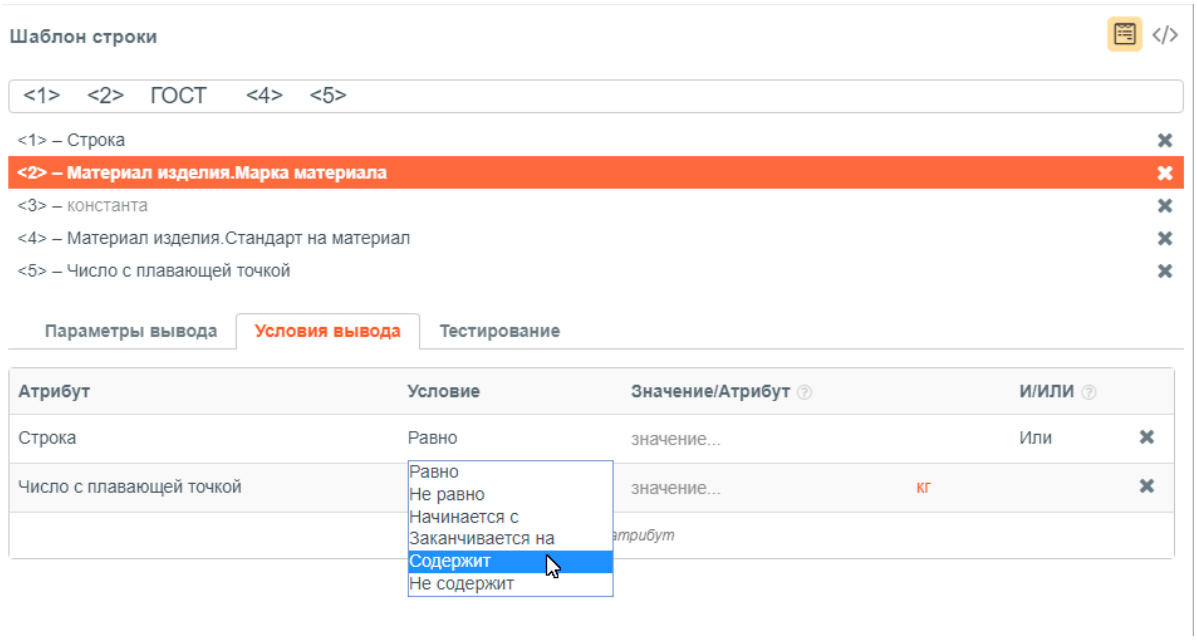


Рисунок 315 - Условия вывода значения атрибута

Атрибут-условие добавляется из списка перетаскиванием на строку таблицы. Дополнительно выбирается условие, определяющее значение атрибута. Оно может принимать вид – *Равно*, *Не равно*, *Начинается с*, *Заканчивается на*, *Содержит*, *Не содержит*.

Для атрибутов типа «число с плавающей точкой» дополнительно можно указать единицы измерения, в которых выводится значение.

3.17.5 Тестирование формулы

Для тестирования разработанной формулы (шаблона) нужно перейти на вкладку **Тестирование** (Рисунок 316).

Нажать на кнопку + ***Добавить позицию***, установить курсор в открывшейся справочной группе на нужную позицию и нажать кнопку ***Выбрать***.

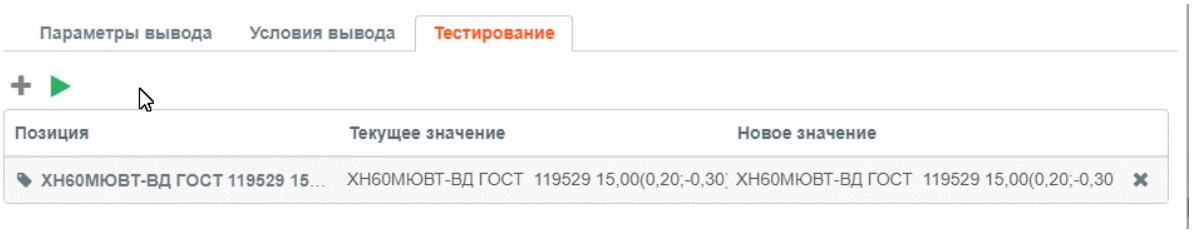


Рисунок 316 - Тестирование формулы

После добавления позиции нажать кнопку ▸ ***Выполнить***. Результаты работы формулы, будут показаны в таблице в столбце **Новое значение**.

Если в скрипт добавлена функция отладки add (String message): void, то в процессе тестирования на консоль отладки выводятся отладочные сообщения. Отладочная консоль открывается по специальной кнопке 🐞, расположенной рядом с кнопкой ***Удалить*** в строке с тестируемой позицией (Рисунок 317).

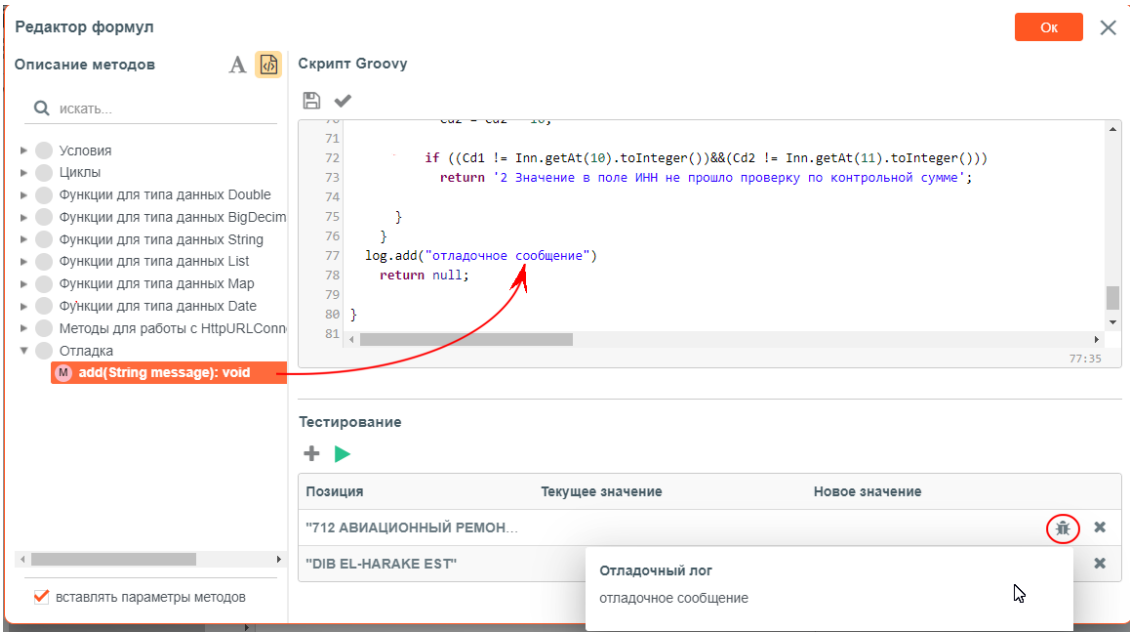


Рисунок 317 - Отладочные сообщения при тестировании формулы

3.18 Ограничительные таблицы

Ограничительные таблицы – это особый вид информационных объектов, которые хранят и позволяют редактировать данные о возможных значениях атрибутов, принимаемых позициями справочника, в зависимости от

нормативных документов, условий использования или иных ограничений. Создание и редактирование ограничительных таблиц разрешается пользователям с привилегией «Редактирование структуры».

Ограничительные таблицы делятся на глобальные и локальные.

Локальные ограничительные таблицы - создаются и ведутся в рамках одной справочной группы. Данный вид таблиц востребован, когда необходимо ввести ограничительный перечень данных по каталогу или документу, действия которого распространяются только на позиции текущей справочной группы. Локальные ограничительные таблицы могут быть скопированы из одной справочной группы в другую, с возможностью её дополнения и расширения. Количество ограничительных таблиц, назначаемых на одну справочную группу не ограничено. Количество колонок в ограничительной таблице ограничено количеством атрибутов, зарегистрированных для справочной группы.

Глобальные ограничительные таблицы создаются в представлении **Конфигурация данных** в отдельной собственной группе **Ограничительные таблицы**. Данный вид таблиц востребован, когда необходимо создать единый словарь данных и использовать его при вводе данных в различных справочных группах. Например, атрибуты *Фамилия, Имя* и *Отчество* могут использовать как в группе, в которой содержится информация по сотрудникам, так и в группе, в которой хранится информация об индивидуальных предпринимателях, с которыми предприятие заключает договора. В этом случае удобным было бы создать глобальные словари *Фамилии, Имена* и *Отчества*, а также возможности их сочетаний, на основе которых происходило формирование значений атрибутов, а также оценка качества данных этих атрибутов. Например, если фамилия, имя и отчество есть в глобальной ограничительной таблице, то качество данных этих атрибутов является высоким.

Подключение глобальной ограничительной таблицы к группе разрешается пользователям с привилегией «Редактирование структуры».

Редактирование структуры и данных глобальных ограничительных таблиц разрешено пользователям с привилегией «Редактирование метаданных».

3.18.1 Создание новой ограничительной таблицы

Последовательность создания ограничительной таблицы, как локальной, так и глобальной одинаково.

Чтобы создать новую таблицу необходимо:

- перейти в режим редактирования;
- установить курсор на справочной группе, для которой будет создаваться ограничительная таблица, и перейти в карточку группы.
- в разделе **Ограничения**, в строке **Ограничительные таблицы** нажать кнопку ***Редактор ограничительных таблиц;***
- в открывшемся редакторе (Рисунок 318) нажать кнопку ***Добавить таблицу*** и выбрать команду **Локальная таблица**, если предусматривается, что с таблицей будет работать только текущая справочная группа или **Глобальная таблица**, если её ограничения будут использоваться во всех группах;

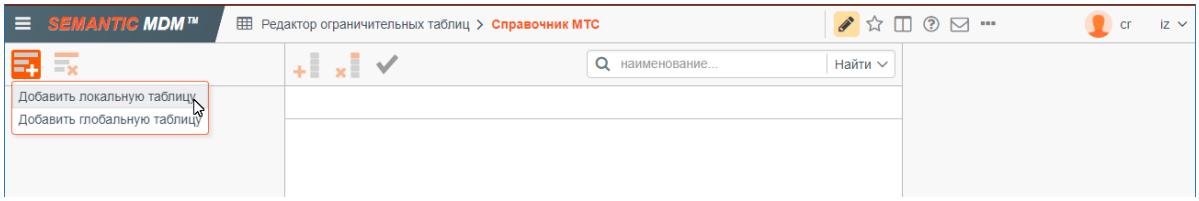


Рисунок 318 - Создание новой ограничительной таблицы

- в карточке таблицы (Рисунок 319) следует указать наименование таблицы (на английском языке на усмотрение пользователя);
- маркер **Обязательность** отвечает за то, что все позиции справочной группы будут проверяться по данной таблице, а если маркер не установлен, то ограничительная таблица будет носить вспомогательный характер и будет предназначена для выбора значений атрибутов из списка;

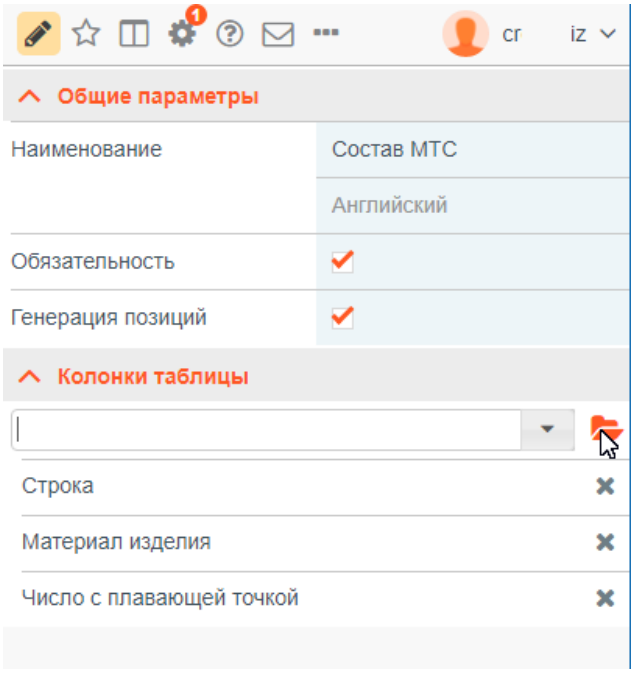


Рисунок 319 - Карточка ограничительной таблицы

- установленный маркер **Генерация позиций** обеспечивает автоматическую генерацию позиций справочной группы в соответствии с ограничениями, установленными текущей таблицей. Если маркер не установлен, то позиции создаются обычным порядком (см. п. 3.14.11), а ограничительная таблица используется для проверки данных;
- в разделе карточки **Колонки таблицы** необходимо выбрать атрибуты, значения которых будут проверяться по ограничительной таблице. В выпадающем списке можно выбрать атрибуты, зарегистрированные для текущей справочной группы, а при нажатии кнопки ***Добавить атрибуты*** из **Классификатора атрибутов** можно выбрать любые необходимые атрибуты.

В результате будет создана ограничительная таблица, в которой число колонок равно числу выбранных атрибутов. Далее следует перейти к заполнению таблицы.

3.18.2 Настройка ограничительной таблицы

Заполнение ограничительной таблицы происходит на сетке данных. Для создания новых строк в таблице следует нажать кнопку ***Добавить строку*** (Рисунок 320) на панели инструментов над областью с таблицей.

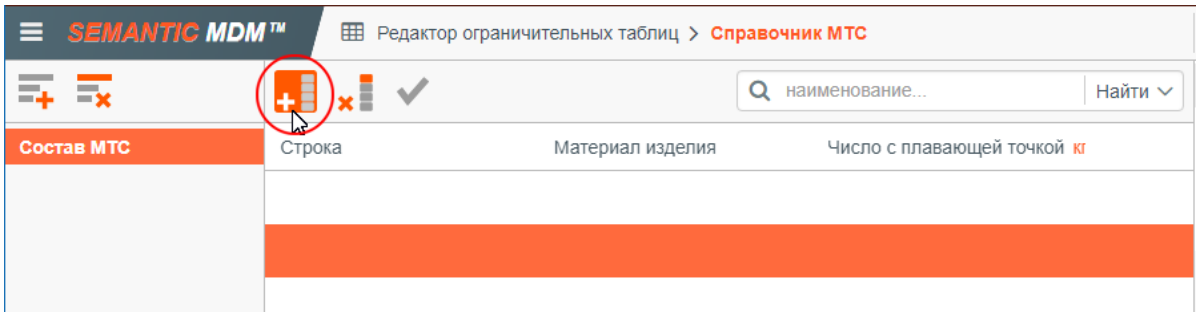


Рисунок 320 - Добавление строк в ограничительную таблицу

Значения в ячейках таблицы указываются в карточке вручную или при помощи выпадающего списка, если значения были указаны в ранее созданных позициях.

Если значение атрибута является ссылкой на значение атрибута другого справочника (например, изделие имеет ссылку на марку материала), то в ограничительной таблице можно выбрать значения из списка значений агрегированного атрибута.

Кроме абсолютных (напрямую указанных) значений ограничительные таблицы поддерживают условные значения, которые вводятся с клавиатуры или с помощью пиктограмм (Таблица 24):

Таблица 24. Условные значения в ограничительных таблицах

Условное значение	Описание назначения
	С клавиатуры вводится символ «*». Значение атрибута может быть любым
	С клавиатуры вводится символ «|». Атрибут должен быть обязательно заполнен.

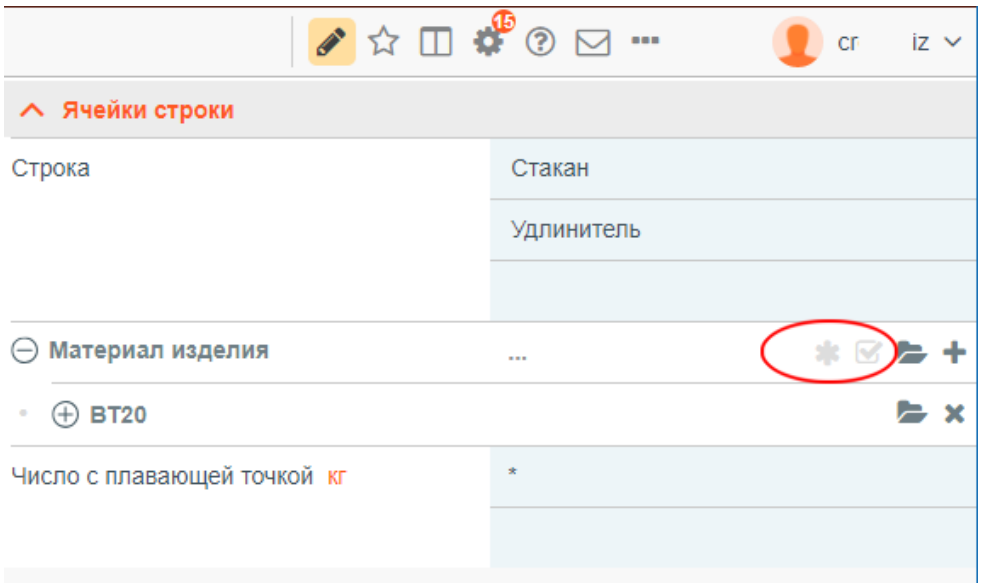


Рисунок 321 - Заполнение полей ограничительной таблицы

Условные значения могут быть скомбинированы. Запись «*|» будет означать, что значение атрибута может быть любым, но обязательно заполненным, т.е. ячейка не может остаться пустой (Рисунок 321).

Если на группу назначены несколько ограничительных таблиц, то они могу быть как независимыми друг от друга, так и связанными. Связанными окажутся те таблицы, в которых используются одинаковые атрибуты. Это следует учитывать при использовании таблиц для генерации позиций. Значения атрибута, применяемого в нескольких ограничительных таблицах, должны быть везде одинаковыми, в противном случае, позиции, создаваемые по ним, никогда не будут удовлетворять ограничениям.

Над таблицей справа расположена ячейка быстрого поиска с возможностью вызова расширенного поиска. Быстрый поиск выполняется по всем значениям ограничительной таблицы. Расширенный поиск ведется по значениям атрибутов, которые можно выбрать из выпадающего списка или задать вручную (Рисунок 322).

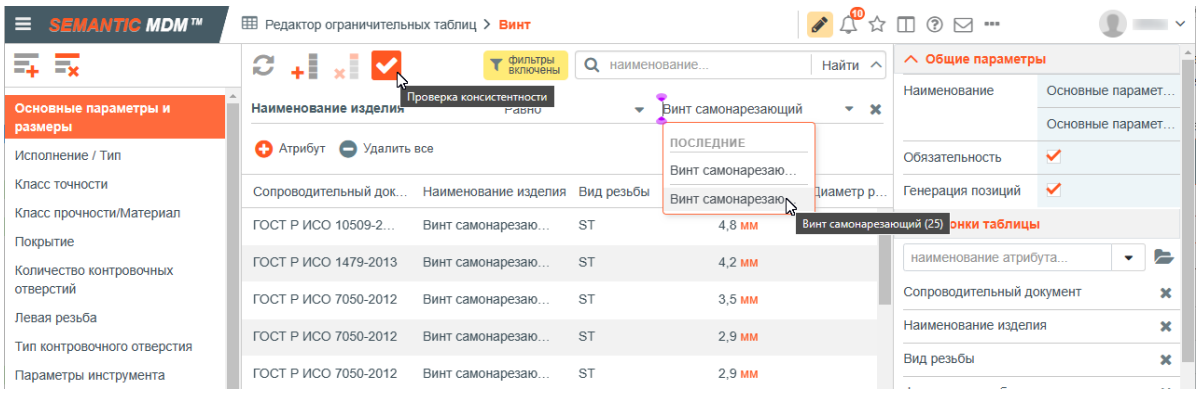


Рисунок 322 - Доступ к быстрому и расширенному поиску в ограничительных таблицах

Кнопка ***Проверка консистентности*** при активизации проверяет, что данные в связанных ограничительных таблицах не противоречат друг другу. При наличии противоречий *Система* выдает предупреждение.

3.18.3 Генерация ограничительных таблиц по существующим позициям

Очень часто вопрос использования ограничительных таблиц встаёт перед пользователями после того, как справочник был частично или полностью наполнен. В таком случае, можно воспользоваться обратным способом заполнения ограничительных таблиц.

Способ заключается в следующем:

- включить режим управления изменениями;
- создать и настроить в карточке группы ограничительную таблицу, так как указано в предыдущих пунктах, но оставить её пустой – без значений;
- перейти в нужную ветку справочника и установить курсор на записи в таблице данных;
- вызвать контекстное меню и выбрать команду **Ограничительные таблицы -> Сгенерировать строки для позиции** либо **Сгенерировать строки для всех позиций** (Рисунок 323).

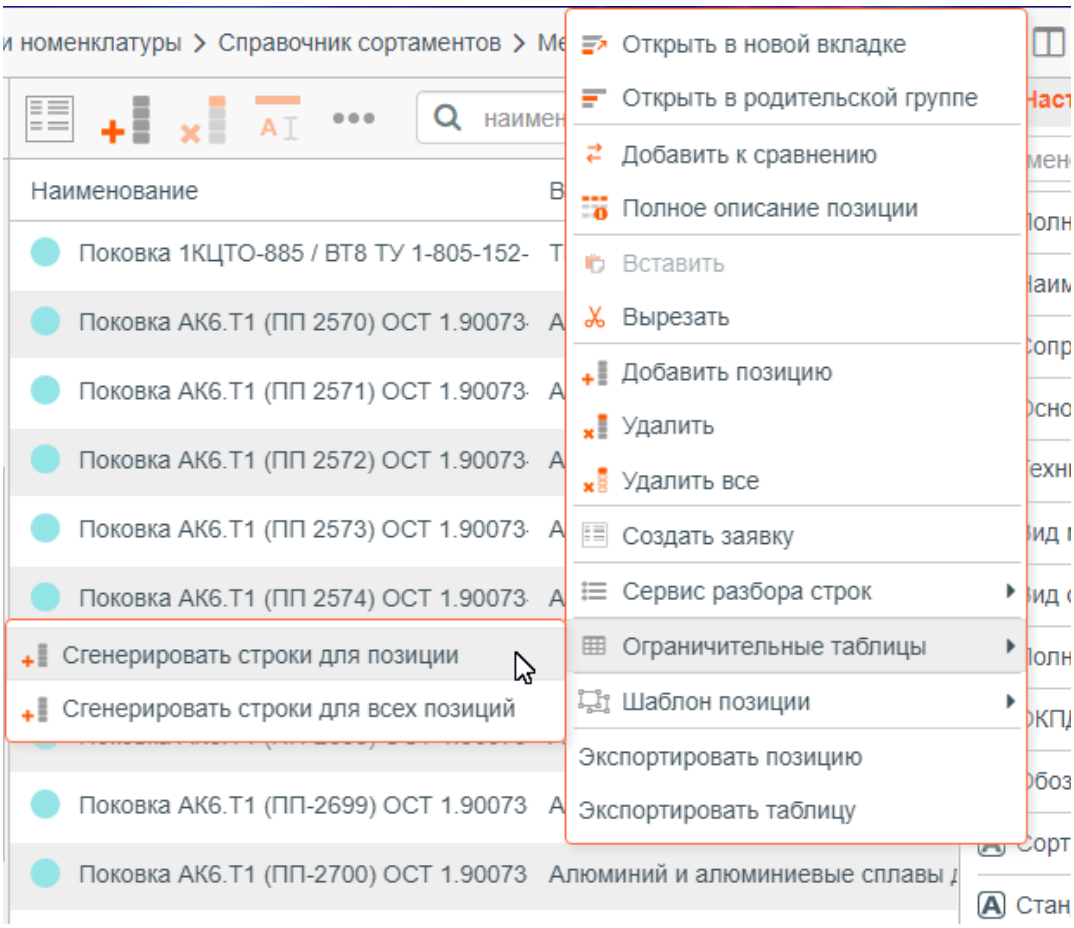


Рисунок 323 - Вызов команд генерации строк ограничительной таблицы

При этом данные выбранной строки или всех строк текущей ветки справочника будут записаны в ограничительную таблицу, которая затем может быть отредактирована и дополнена.

Аналогичным образом следует поступить, когда после создания новой ограничительной таблицы все ранее созданные позиции получили предупреждение о несоответствии ограничениям. Такое произойдет даже в том случае, если позиции заполнены правильно. Следует выполнить генерацию ограничительной таблицы по строкам с предупреждениями так, как описано выше.

3.19 Экспорт и импорт справочных данных

Для массового заполнения справочников на этапе внедрения *Системы*, а также анализа и проверки существующих данных можно воспользоваться

функционалом, предоставляемым сторонними табличными редакторами и возможностями формата XML.

3.19.1 Экспорт и импорт позиций и групп в Excel© и CSV

Для выполнения экспорта позиций в *Excel* или файл формата *CSV* необходимо выполнить следующие действия:

- установить курсор на нужной ветке справочника в дереве;
- выбрать в списке команду **Экспорт в Excel** или **Экспорт в CSV** (Рисунок 324) (для выполнения экспорта не требуется создавать пакет изменений);

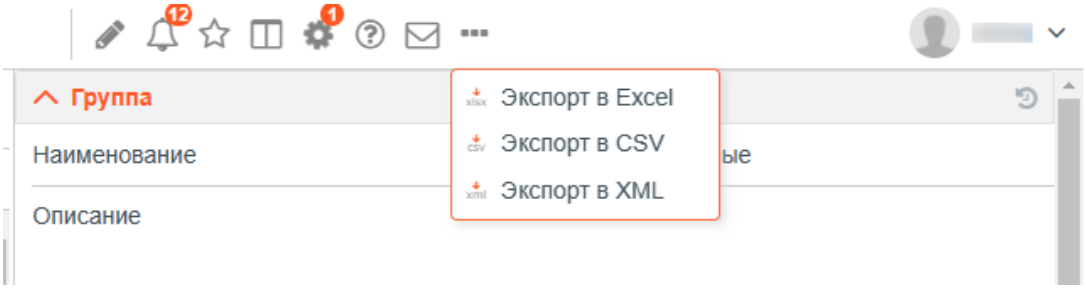


Рисунок 324 - Вызов команд экспорта

- в открывшемся диалоге Экспорт <наименование справочника> (Рисунок 325) в Excel (в CSV) следует:
  - выбрать **Тип значений**, которые будут выгружаться. Выгрузить в файл можно отдельно либо актуальные либо исторические значения;
  - настроить **Опции** в части выгрузки в файл системных атрибутов;
  - в разделе **Атрибуты** установить маркеры напротив тех атрибутов, значения которых требуется выгрузить в файл. Если какие-либо атрибуты присутствовали в таблице данных, то маркеры напротив них буду установлены автоматически. Для поиска атрибутов можно воспользоваться строкой поиска, расположенной над списком атрибутов. Установленный маркер **только выделенные** оставит в списке только помеченные к выгрузке атрибуты. После

нажатия на кнопку ***Экспортировать*** выгрузка начнется автоматически;

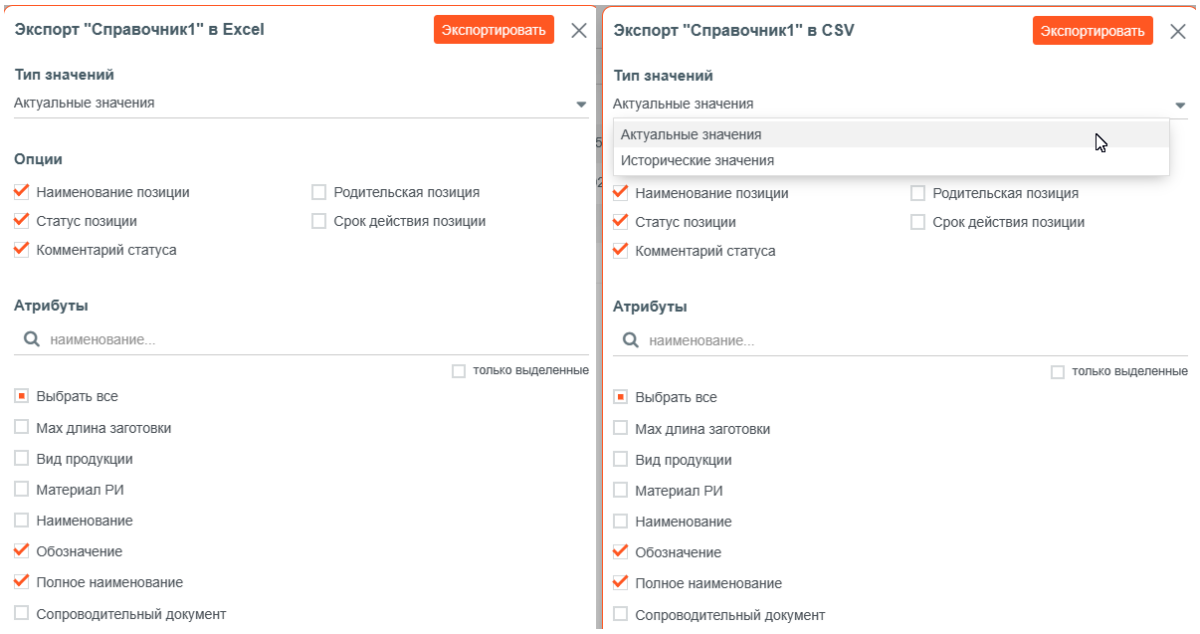


Рисунок 325 - Выбор атрибутов для выгрузки в файл

- после формирования файла нажать кнопку ***Фоновые процессы*** и, в открывшемся одноименном окне, выбрать ссылку «файл» (Рисунок 326).

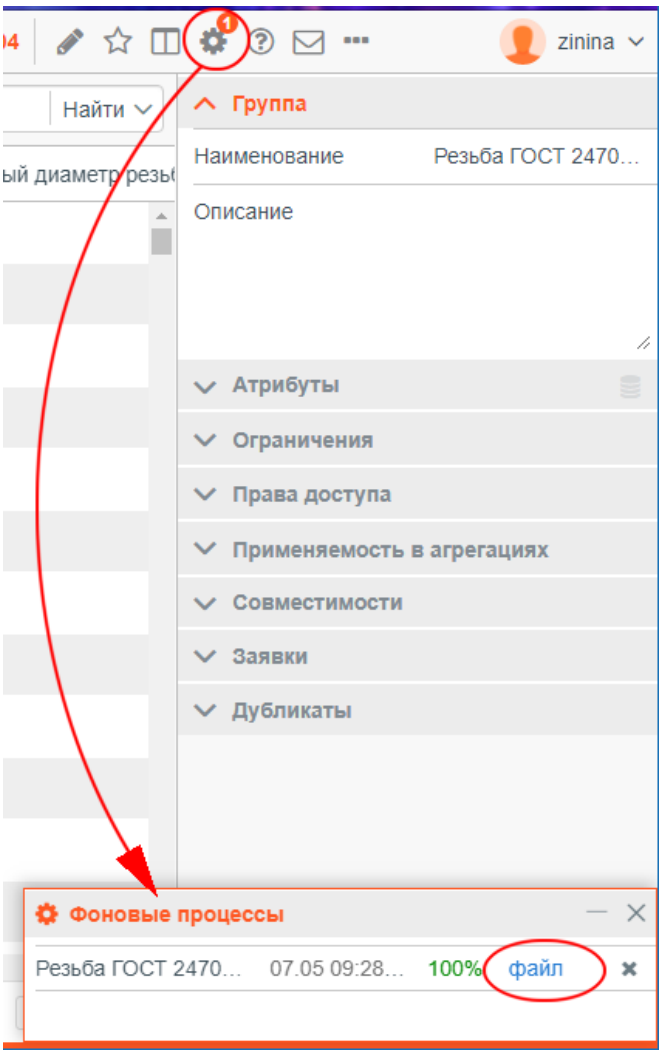


Рисунок 326 - Загрузка сформированного файла

Сформированный файл будет загружен на ПК и его можно будет открыт в приложении *Excel*. После чего выполнять редактирование и сохранять.

Процедура импорта данных из *Excel и CSV* выполняется аналогично. Для загрузки данных в справочник из файла необходимо:

- включить режим управления изменениями;
- установить курсор на ветке справочника, в которую будут загружаться данные;
- вызвать из меню команду **Импорт из Excel** или **Импорт из CSV** (Рисунок 327) и выбрать в *Проводнике Windows* требуемый файл, после чего, будет запущена процедура загрузки данных в таблицу. Причём, будут загружены только те атрибуты, для которых настроена видимость колонок на сетке данных .

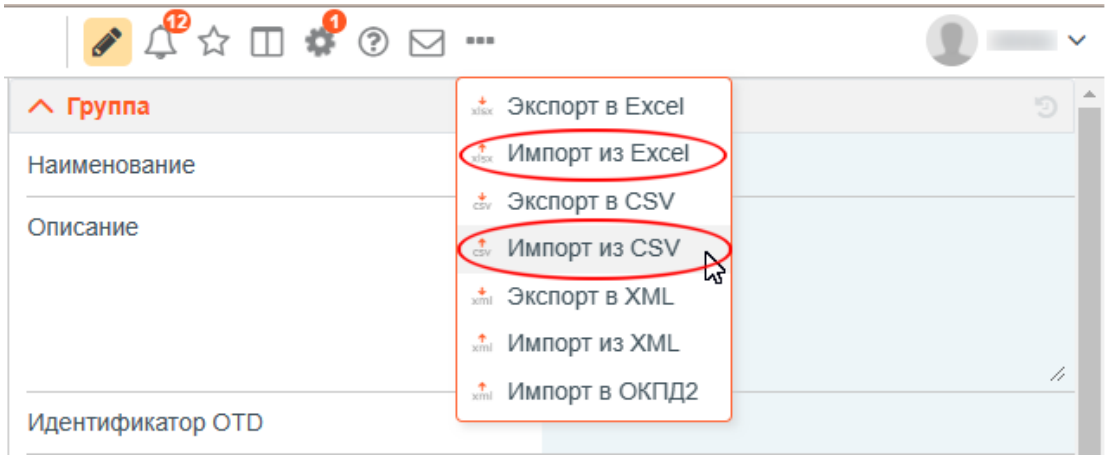


Рисунок 327 - Загрузка данных в справочник из файла

Для справочных групп доступны два вида выгрузки / загрузки в Excel:

- Экспортировать подчиненные группы в Excel;
- Экспортировать декларации в Excel.

Для выполнения экспорта подчиненных групп следует установить курсор на наименовании справочной группы и вызвать из контекстного меню команду **Экспорт -> Экспортировать подчиненные группы в Excel** (Рисунок 328). При выполнении экспорта в *Excel* выгружаются следующие колонки:

- GUID группы;
- GUID родительской группы;
- Статус;
- Наименование группы;
- Описание группы;
- Идентификатор OTD.

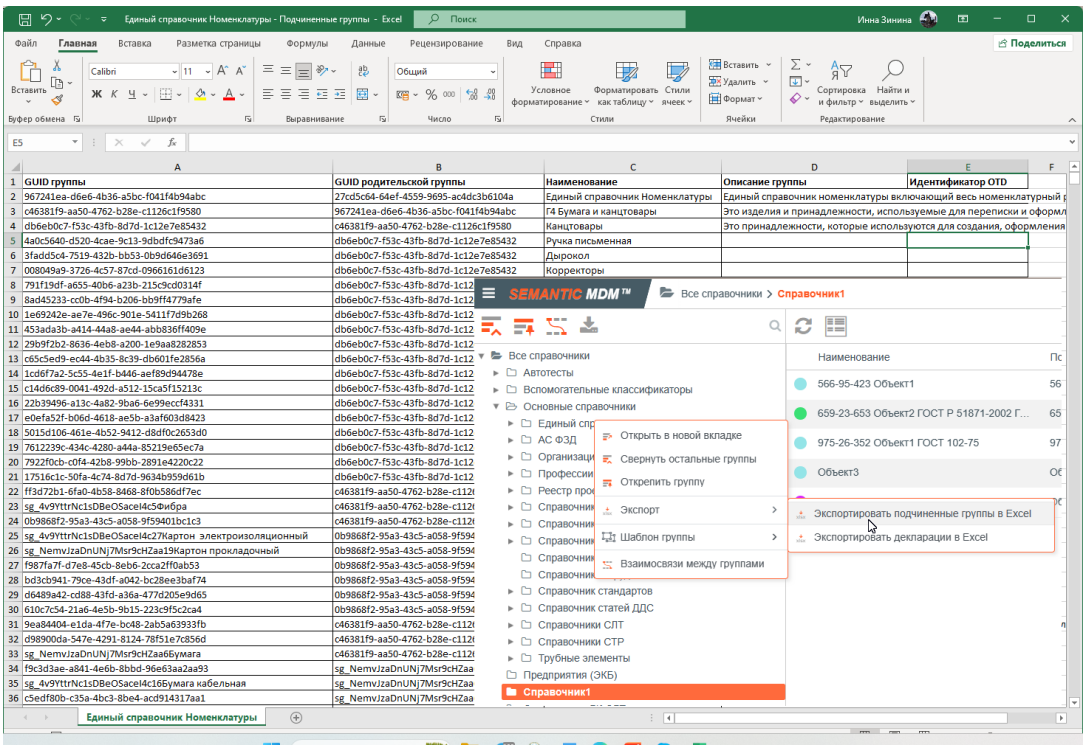


Рисунок 328 - Выполнение экспорта иерархии групп справочника

Для выполнения экспорта деклараций атрибутов следует установить курсор на наименовании справочной группы и вызвать из контекстного меню команду **Экспорт -> Экспортировать декларации в Excel**. При выполнении экспорта в *Excel* выгружаются следующие колонки:

- GUID группы;
- Наименование группы;
- GUID атрибута;
- Наименование атрибута;
- Категория атрибутов;
- Входит в полное описание;
- Отображать в таблице;
- Область;
- Маска ввода;
- Значение по умолчанию;
- Обозначение ЕИ;
- Точность.

Выполнение импорта производится в режиме управления изменениями. Следует установить курсор на корневой элемента дерева или наименование родительской группы и вызвать из контекстного меню команду **Импорт -> Импортировать группы из Excel** (Рисунок 329). Для импорта используется аналогичная структура файла, с помощью которой можно выполнить создание новой иерархии групп, выполнить реструктуризацию имеющихся групп в онтологическом классификаторе или загрузить атрибуты.

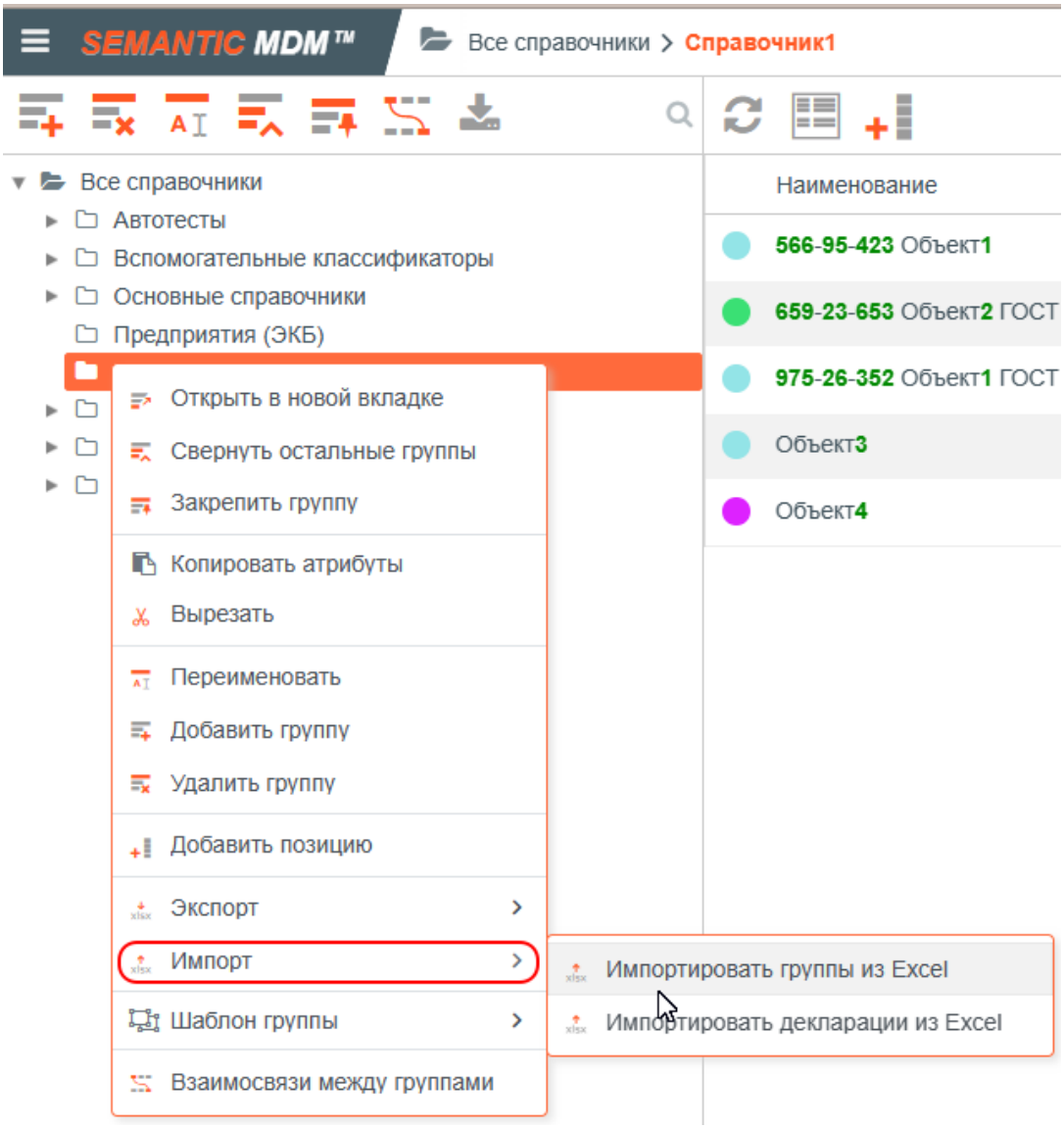


Рисунок 329 - Доступ к командам загрузки структуры справочников из файла Excel

3.19.2 Экспорт и импорт данных в формат XML

Функционал экспорта позволяет выгрузить описание:

- Всех атрибутов из классификатора атрибутов, которые использовались в модели экспортируемого справочника;
- Всей иерархии подгрупп для экспортируемой группы, а также всех зависимых групп с их иерархией, на которые имеются ссылки из деклараций экспортируемых групп. Т.е. выполняется экспорт не только основной ветки справочника, но и всех зависимых от неё;
- Всех деклараций атрибутов и формул, для экспортируемых групп;
- Структуру групп справочника вместе с ограничительными таблицами, ограничительными функциями, назначенными на группы уникальными и обязательными атрибутами;
- Совместимости между различными объектами;
- Справочников из **Конфигурации данных**, в которых используются выгружаемые группы. Кроме этого, справочники в конфигураторе доступны для экспорта в XML вместе со связанными с ними группами и позициями;
- Всех позиций, которые доступны для текущей выделенной группы с учетом наложенных поисков и фильтров;
- Пост-обработчиков, с помощью которых в *Semantic MDM* реализуется обогащение данных;
- Выделенных в таблице позиций.

Для выполнения экспорта данных в формат XML необходимо выполнить следующие действия:

- установить курсор на нужной ветке справочника в дереве;
- выбрать в списке команду **Экспорт в XML** (для выполнения экспорта не требуется создавать пакет изменений);
- в открывшемся окне (Рисунок 330) установить маркеры, напротив тех элементов структуры справочника, которые нужно выгрузить;

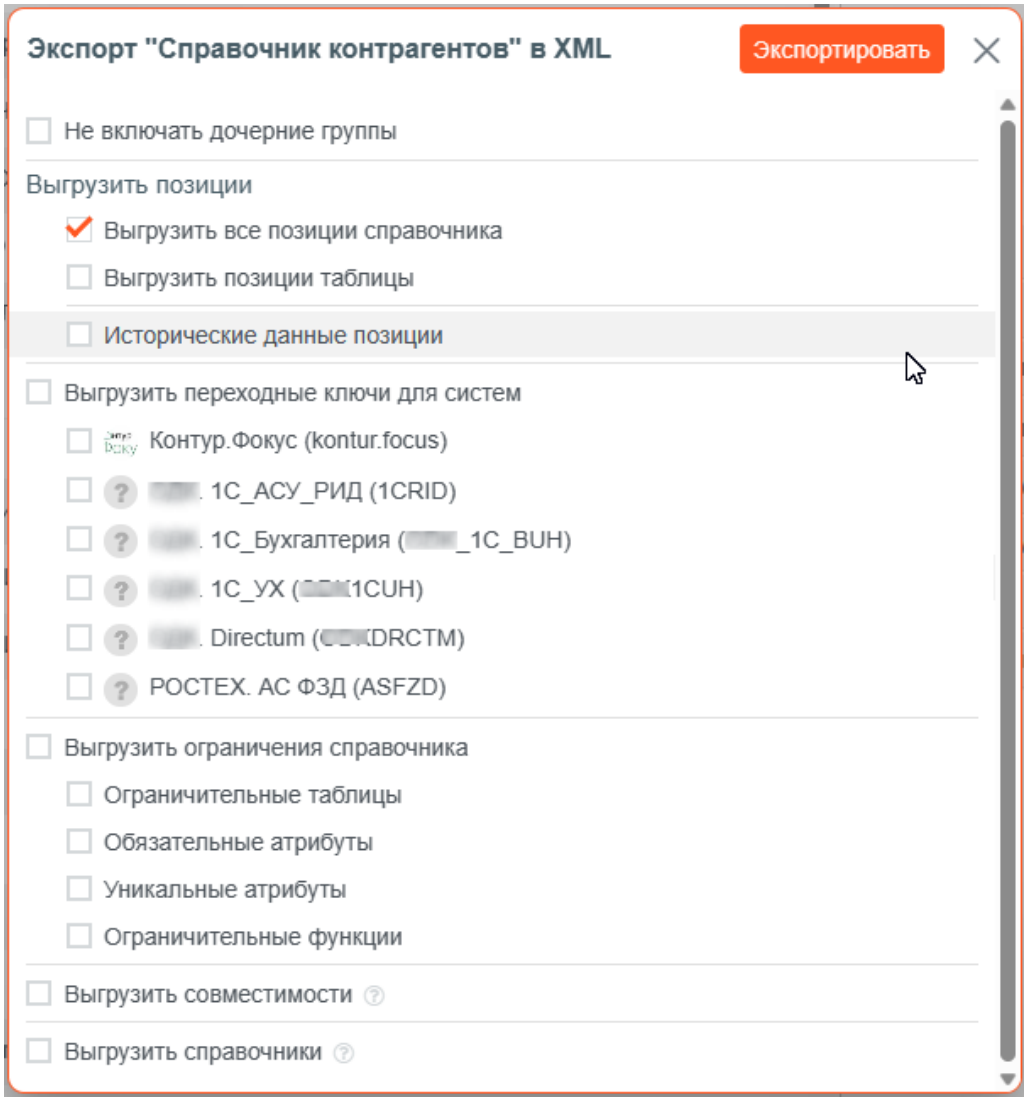


Рисунок 330 - Выбор элементов структуры для экспорта в XML

- нажать кнопку ***Экспортировать***;
- нажать кнопку ***Фоновые процессы*** и, в открывшемся одноименном окне, нажать на ссылку файл.

Сформированный файл будет загружен на ПК пользователя и его можно будет открыть и просматривать соответствующим приложением. Функционал импорта позволяет воссоздать структуру основного и зависимых справочников в том же виде, в котором она экспортировалась из источника. Функционал импорта работает в режиме корректировки и дополнения существующей структуры справочника и не выполняет никаких операций удаления. При выполнении импорта множественных агрегаций/композиций из XML производится замена массива имеющихся агрегаций/композиций на

массив агрегаций/композиций из XML. При выполнении импорта ограничительной таблицы выполняется поиск существующих колонок по GUID атрибута и, если GUID колонок не совпадают, то обновляется GUID колонки. Это позволяет избежать рассинхронизации GUID колонок при создании столбца на один и тот же атрибут в источнике и приемнике.

Процедура импорта данных из формата XML выполняется в режиме управления изменениями. Для загрузки данных в справочник из файла XML необходимо:

- включить режим управления изменениями;
- установить курсор на ветке справочника, в которую будут загружаться данные;
- вызвать из меню команду **Импорт из XML**, после чего, будет запущена процедура загрузки структуры.

3.19.3 Шаблон группы

В *Системе* можно сформировать в формате MS Word© и выгрузить шаблон с описанием группы, который включает в себя следующую информацию:

- Классификация общероссийскими и иными классификаторами. Данная информация формируется на основе классифицирующих атрибутов, для которых установлен маркер «полное описание» и допустимого списка значений в ограничительных таблицах;
- **Класс** - наименование справочной группы;
- **Описание класса** - описание справочной группы;
- Таблица из двух колонок, в которую выводятся продекларированные атрибуты, для которых установлен маркер «полное описание». В колонке **Атрибут** выводится наименования атрибута, а в колонке **Описание атрибута** выводиться его описание, выполненное в **Классификаторе атрибутов** (подробнее см. раздел 3.16).

Для генерации шаблона группы необходимо установить курсор в дереве на нужной группе и выбрать в контекстном меню команду **Шаблон группы -> Шаблон Word** (Рисунок 331). Для групп, имеющих подчиненные уровни используется команда **Шаблоны Word вложенных групп**, в этом случае в одном файле формируются сразу несколько таблиц, по одной на каждую подчиненную справочную группу. Шаблон каждой группы начинается с новой страницы документа.

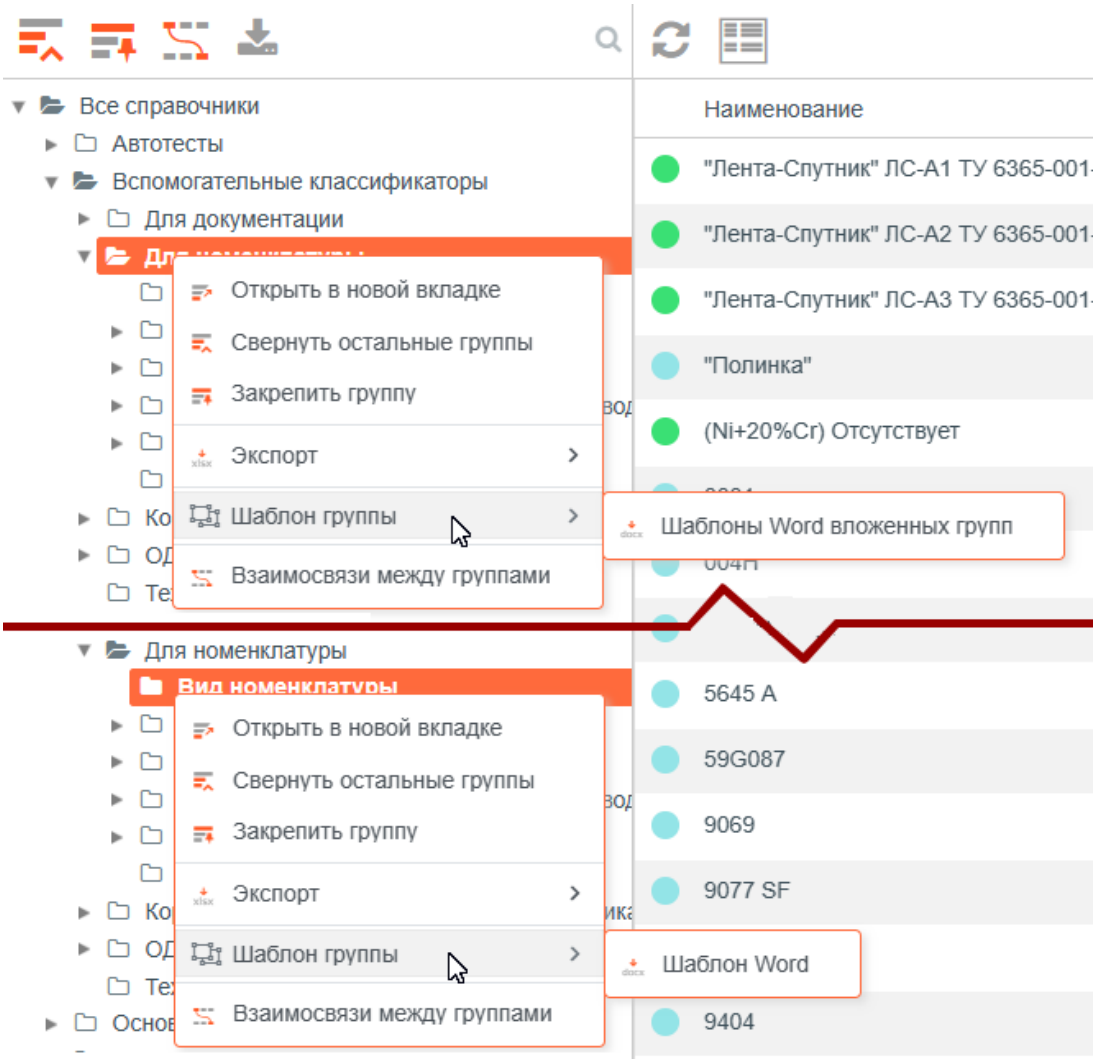


Рисунок 331 - Вызов команды для создания шаблона группы

После выбора команды **Шаблон Word (Шаблоны Word вложенных групп)** файл в формате *.docx будет сформирован автоматически и доступен для скачивания. Структура шаблона показана на Рисунок 332.

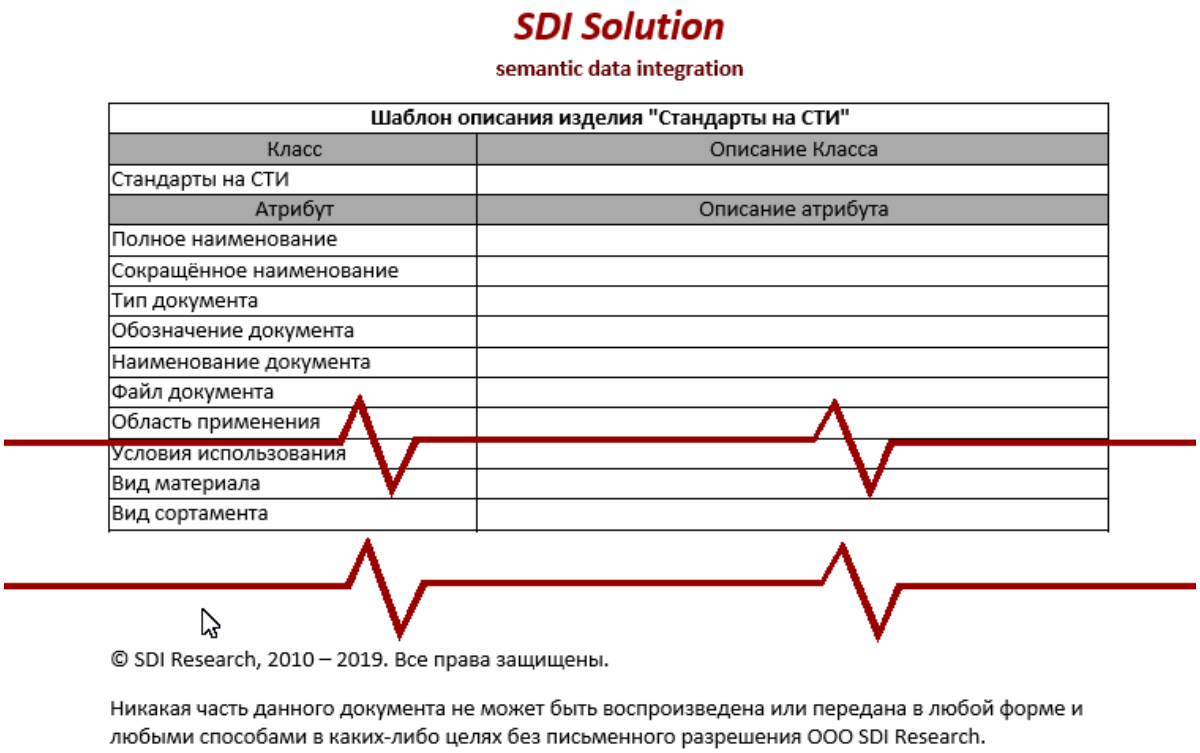


Рисунок 332 - Вид шаблона группы в формате Word

3.20 Иерархические классификаторы

Отдельный вид информационных объектов, предназначенный для классификации позиций других справочников по каким-либо признакам. В *Системе* доступен специальный функционал для создания и применения иерархических классификаторов.

Иерархический классификатор создаётся в подсистеме **Конфигурация данных** в специальном разделе **Классификаторы** (Рисунок 333). Внутри раздела классификаторы могут быть также объединены в группы.

Открыть иерархический классификатор можно только со стартовой страницы.

Структура (модель данных) конфигуратора для редактирования классификатора открывается непосредственно из представления **Конфигурация данных,** при помощи команды контекстного меню **Открыть классификатор**.

Классификатор может быть открыт в отдельной вкладке при помощи команды контекстного меню **Открыть в новой вкладке**.

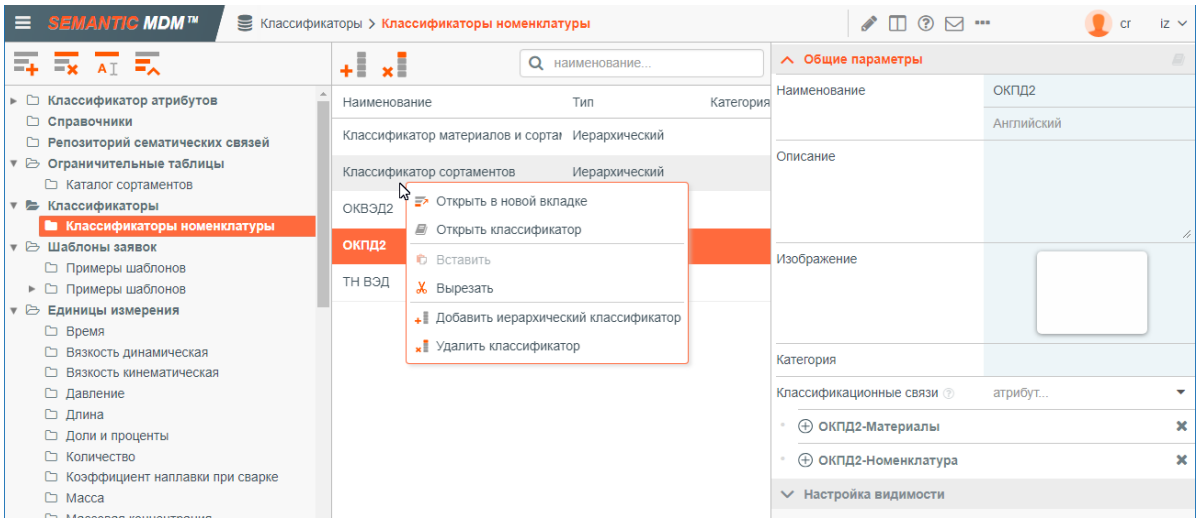


Рисунок 333 - Работа с классификаторами в подсистеме **Конфигурация данных**

3.20.1 Создание иерархического классификатора

Для того, чтобы создать новый классификатор следует:

- в разделе **Классификатор атрибутов** создать атрибут связи, который будет определять связь между классификатором и справочной группой, содержащей классифицируемые позиции. Количество атрибутов связи должно соответствовать числу классифицируемых справочных групп. Атрибуты не требуют выполнения каких-либо настроек;
  - в разделе **Классификаторы** установить курсор на нужной группе и вызвать из контекстного меню команду **Добавить иерархический классификатор** либо перейти на сетку данных и нажать кнопку ***Добавить***;
  - в карточке классификатора ввести наименование на русском и английском (опционально) языках. Дать описание справочника-классификатора в соответствующей ячейке (на усмотрение пользователя);

- в ячейке **Изображение** добавить рисунок, который будет отображаться на пиктограмме классификатора на стартовой странице;
- в ячейке **Категория** можно указать группу (наименование папки) справочников на стартовой странице, в которую будет входить создаваемый классификатор;
- в разделе карточки **Классификационные связи**, используя выпадающий список, выбрать атрибут связи, который был создан на первом шаге. Если атрибутов несколько, то выбрать их из списка последовательно;
- в настройках видимости выбрать группы или отдельных пользователей, которые могут видеть данный классификатор.
- Если в разных справочных группах атрибут, содержащий классификационный признак, носит разные наименования, то для этого в классификатор

3.20.2 Настройка классификатора

После создания нового классификатора необходимо перейти к настройке его структуры. Для чего его необходимо открыть из представления **Конфигурация данных** при помощи команды контекстного меню.

Для настройки структуры классификатора необходимо выполнить следующие действия:

- установить курсор на наименовании классификатора в таблице и вызвать из контекстного меню команду **Открыть классификатор**. Классификатор будет открыт в окне на вкладке **Структура**;
- перейти в режим управления пакетами изменений;
- установить курсор на корневом элементе (наименовании классификатора) в дереве и перейти в карточку группы;

- развернуть в карточке элемент **Классификационные связи** (Рисунок 334), где будут присутствовать один или несколько атрибутов, характеризующих классификационную связь.

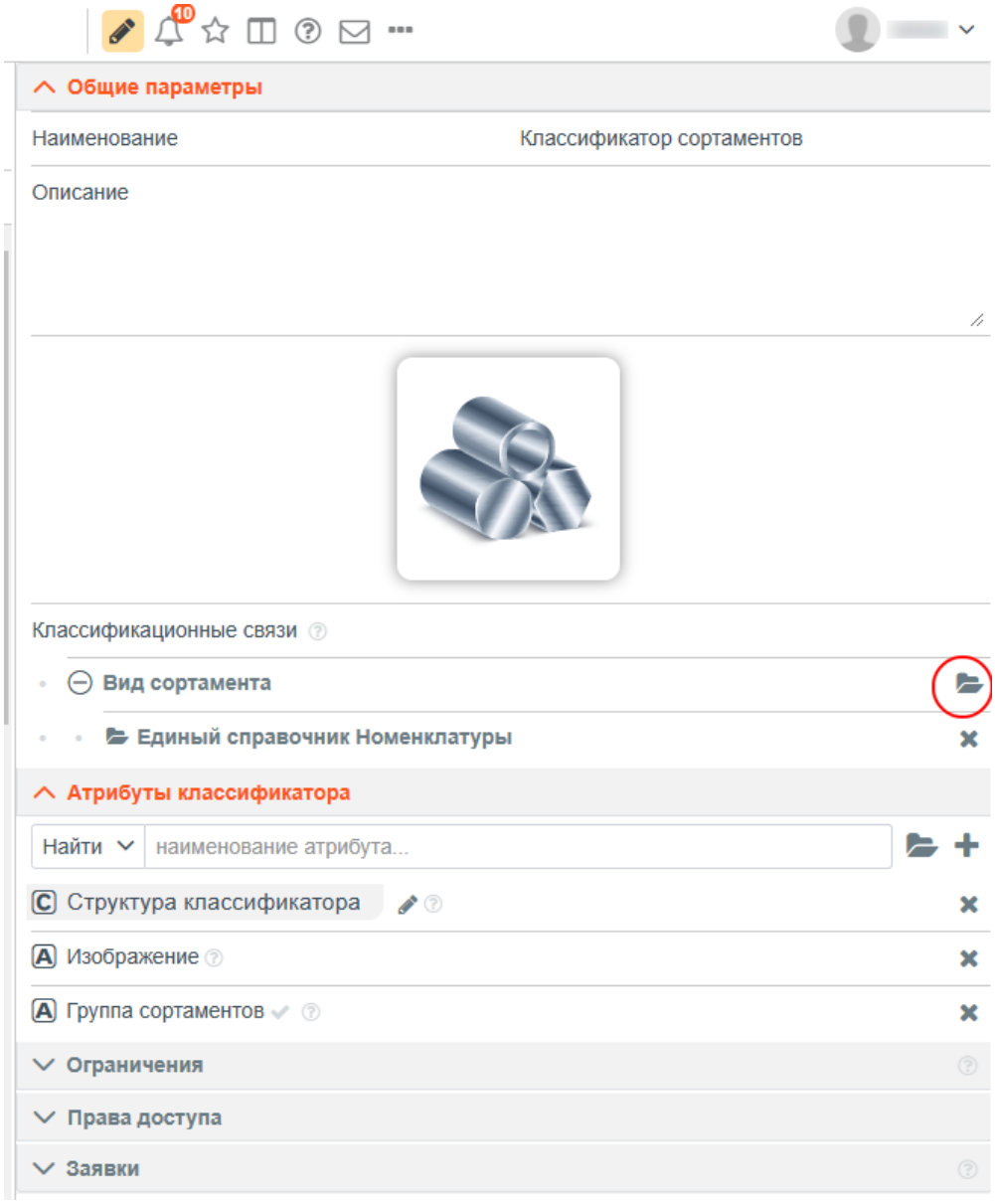


Рисунок 334 - Карточка классификатора с настроенными связями

- нажать на кнопку ***Выбрать группу*** в строке с наименованием классификационной связи и, в открывшемся окне **Группы**, выбрать одну справочную групп, позиции которой будут классифицироваться, и нажать кнопку ***Ок.*** Если необходимо добавить несколько групп, то процедуру следует повторить. Для поиска нужной группы можно

воспользоваться строкой поиска, расположенной в верхней части окна. После добавления группы на сетке данных на вкладке с наименованием классификационной связи внизу иерархического дерева появится новый элемент *Нераспределенные*, установив курсор на который можно увидеть позиции добавленной справочной группы;

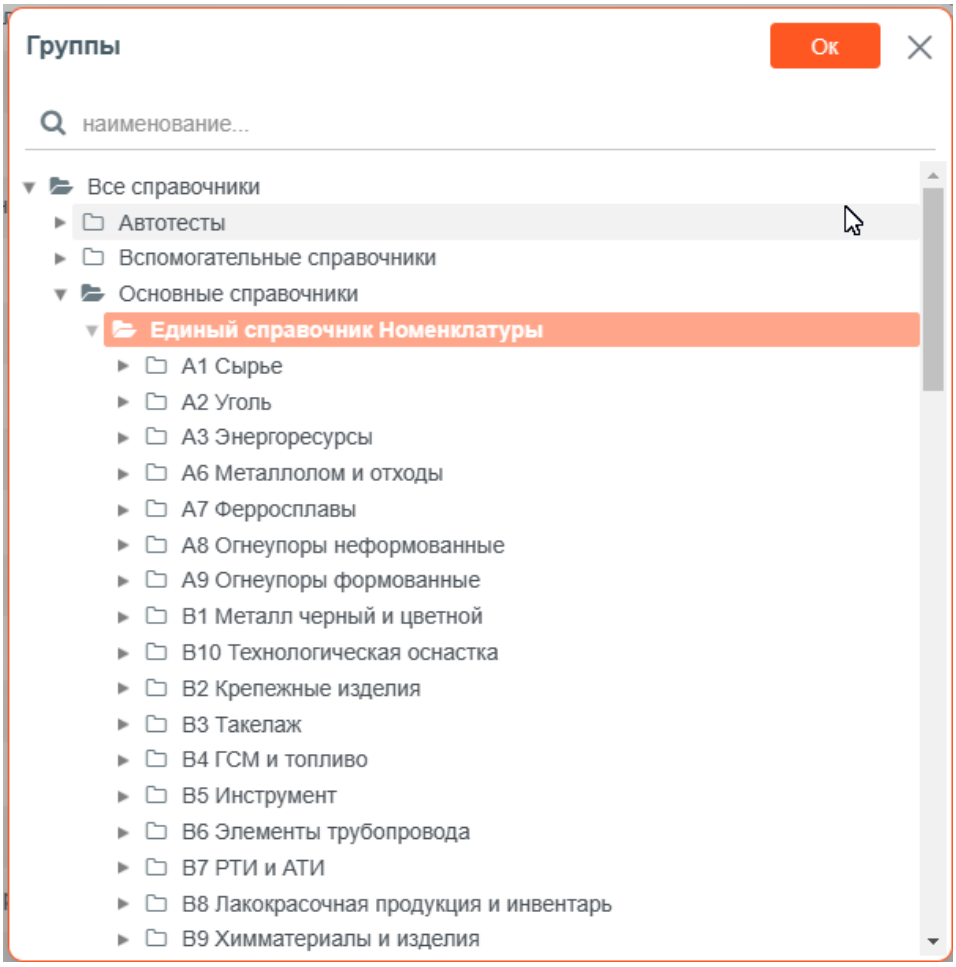


Рисунок 335 - Выбор классифицируемых справочных груп

- в разделе **Атрибуты классификатора** выбрать из выпадающего списка или **Классификатора атрибутов** либо создать новые атрибуты для классификатора и выполнить их настройку.
- в разделе **Ограничения** можно настроить ограничения, обеспечивающие качество данных - ограничительные таблицы, обязательные и уникальные атрибуты, а такжде функции проверки;
- в разделе **Права доступа** следует определить пользователей и группы,

которым разрешено взаимодействовать с этим классификатором;

- в разделе **Заявки** определить **экспертов, обрабатывающих заявки в данном классификаторе, а также маршрут согласования изменений;**
- **после настройки атрибутивного состава нужно перейти в дерево классификатора и, используя кнопку *Добавить группу классификации,*** создать дерево классификатора из необходимого количества уровней;
- выйти из режима редактирования структуры закрыв вкладку или вернувшись на предыдущую страницу.

В иерархических классификаторах доступна работа с историческими значениями. Используя исторические значения можно отобразить иерархический классификатор, актуальный на указанную дату. Получение актуальной структуры классификатора на заданную дату выполняется через расширенный поиск, при просмотре структуры классификатора (Рисунок 336).

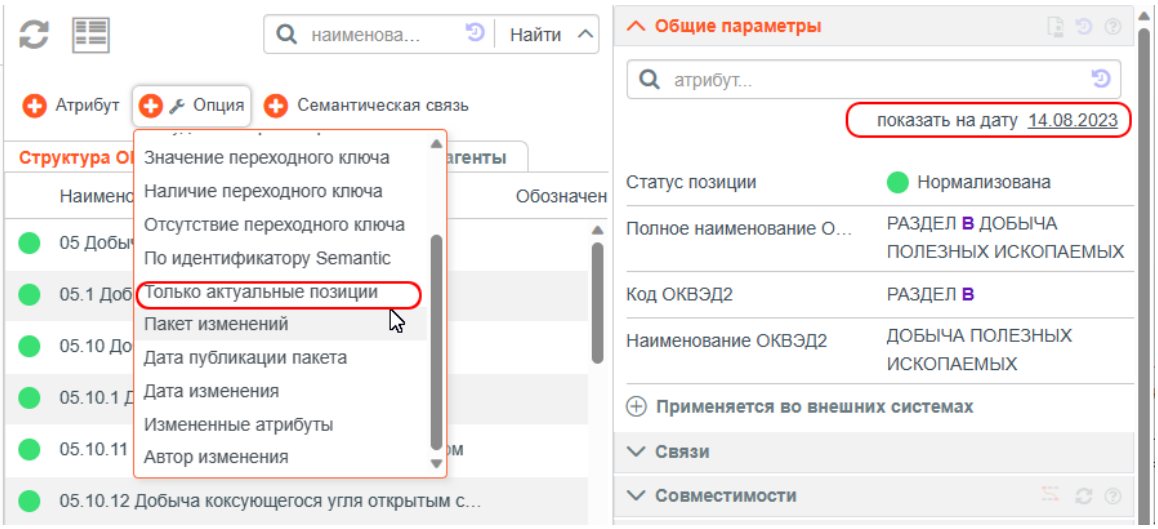


Рисунок 336 - Доступ к историческим значениям в иерархическом классификаторе

3.20.3 Классификация позиций

После создания и настройки классификатора все позиции классифицируемых справочных групп окажутся на вкладке с наименованием

классификационной связи в группе *Нераспределённые*, расположенной в конце дерева (Рисунок 337). Классификацию позиций можно выполнять двумя способами – находясь в классификаторе или находясь в справочнике (справочной группе). Действия при классификации обоими способами выполняются идентичные.

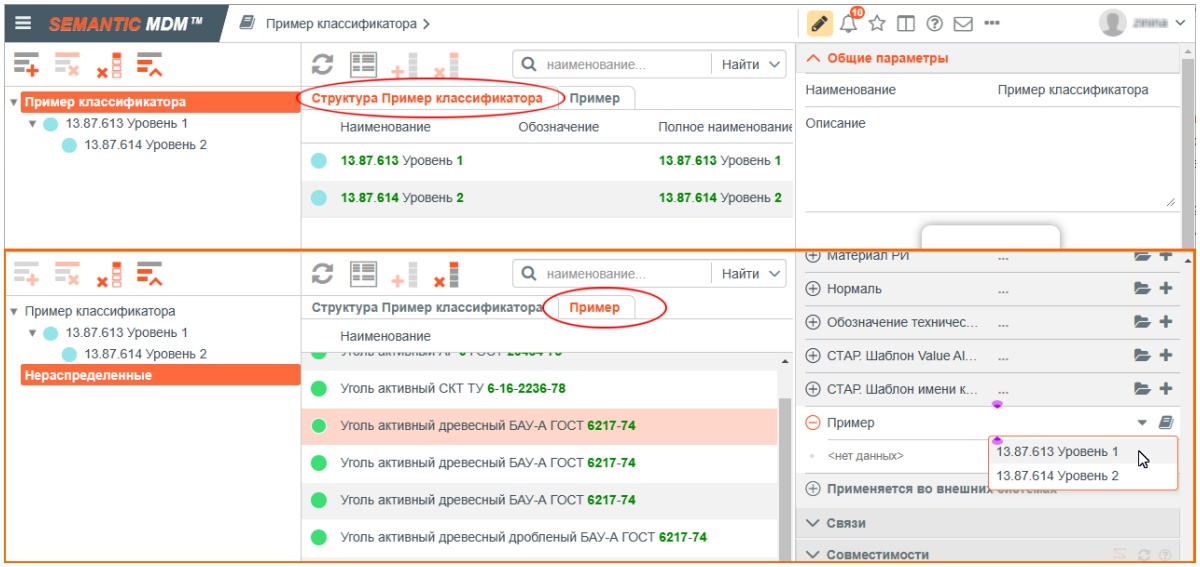


Рисунок 337 - Выполнение классификации через группу *Нераспределенные*

Чтобы выполнить классификацию через классификатор необходимо:

- на стартовой странице нажать кнопку классификатора;
- находясь в классификаторе включить режим управления пакетами изменений и перейти на вкладку с наименованием классификационной связи;
- установить курсор на группе *Нераспределенные* и перейти на сетку данных;
- на сетке данных, используя функции поиска и фильтрации отобрать необходимые позиции;
- включить в таблице отображение столбца с классификационным признаком;
- используя команду **Задать значение** выполнить присвоение классификационного признака (кода группы) отобранным позициям;

- если нужно классифицировать только одну позицию, то на нее нужно установить курсор в таблице и перейти в карточку, где выбрать значение классификационного признака из выпадающего списка.

Для того, чтобы классифицировать позиции со стороны справочника необходимо открыть нужный справочник, установить курсор на позиции в таблице, перейти в карточке позиции в разделе **Общие параметры**, выбрать ячейку с классификационной связью и выбрать значение из выпадающего списка либо нажать кнопку ***Добавить*** в ней. После чего в открывшемся окне классификатора выбрать группу (подгруппу, раздел), к которой относится текущая позиция.

При необходимости можно выполнить массовую классификацию позиций справочной группы.

По щелчку левой кнопкой мыши на классификационной группе (коде) в карточке позиции, будет открыт сам классификатор с выделением на сетке данных той позиции, из которой открыли данный классификатор.

При просмотре классификатора можно выбрать просмотр только нормализованных позиций (маркер **Только нормализованные** внизу таблицы) при этом Система отфильтрует не только классифицируемые позиции, но и дерево классификатора.

3.20.4 Экспорт и импорт классификационных групп

Структура любой ветки справочника-классификатора может быть экспортирована в форматы Excel или XML. Экспорт может производится режиме просмотра (только пользователями с административными правами). Импорт данных из указанных форматов возможен только в режиме управления пакетами изменений.

Для экспорта (импорта) структуры необходимо установить курсор на нужной ветке справочника и вызвать из дополнительного меню соответствующую команду (Рисунок 338).

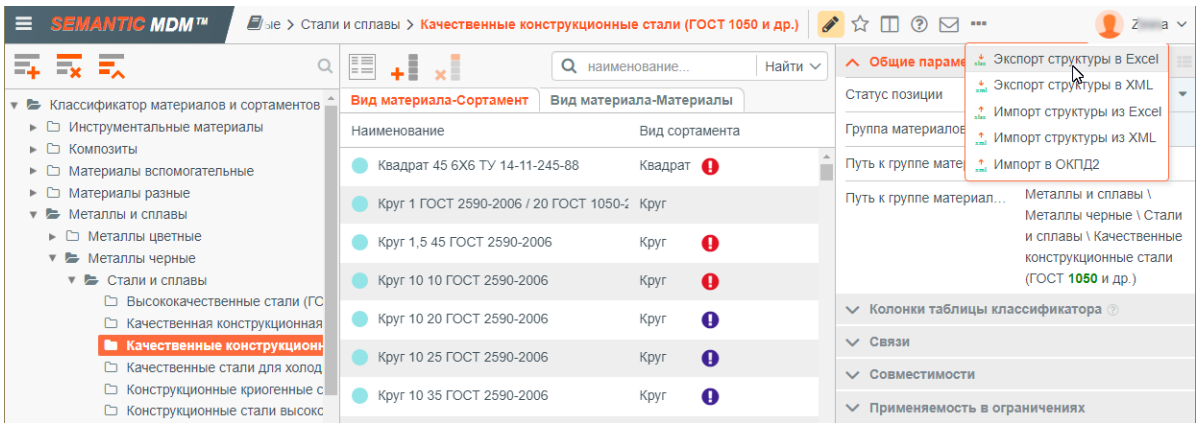


Рисунок 338 - Вызов команд экспорта (импорта) структуры классификатора

3.21 Единицы измерения

Системный справочник *Единицы измерения* (Рисунок 339) предназначен для регистрации и хранения единиц измерения, используемых в справочных группах *Semantic MDM*. В справочнике хранятся как стандартные единицы измерения (ЕИ), принадлежащие системе СИ, так и вспомогательные ЕИ, например, производительность оборудования, трудоёмкость и т.п. Использование системного справочника обеспечивает единообразие применяемых ЕИ, а также возможность их автоматического изменения и перерасчёта.

Единицы измерения объединены в группы в соответствии с измеряемыми сущностями – время, длина, плотность и т.д. Если для сущности используется несколько единиц измерения, то одна из них назначается базовой, для чего в карточке ЕИ устанавливается соответствующий маркер. Базовая единица измерения будет предлагаться автоматически при назначении данной измеряемой сущности, также она важна при экспорте данных во внешние ИС. Остальные ЕИ в группе могут быть связаны с базовой через **Коэффициент пересчёта** или **Формулу пересчёта**, которые указываются в соответствующих ячейках. Ячейка **Обратная формула пересчёта** содержит формулу, позволяющую пересчитать базовую ЕИ в текущую.

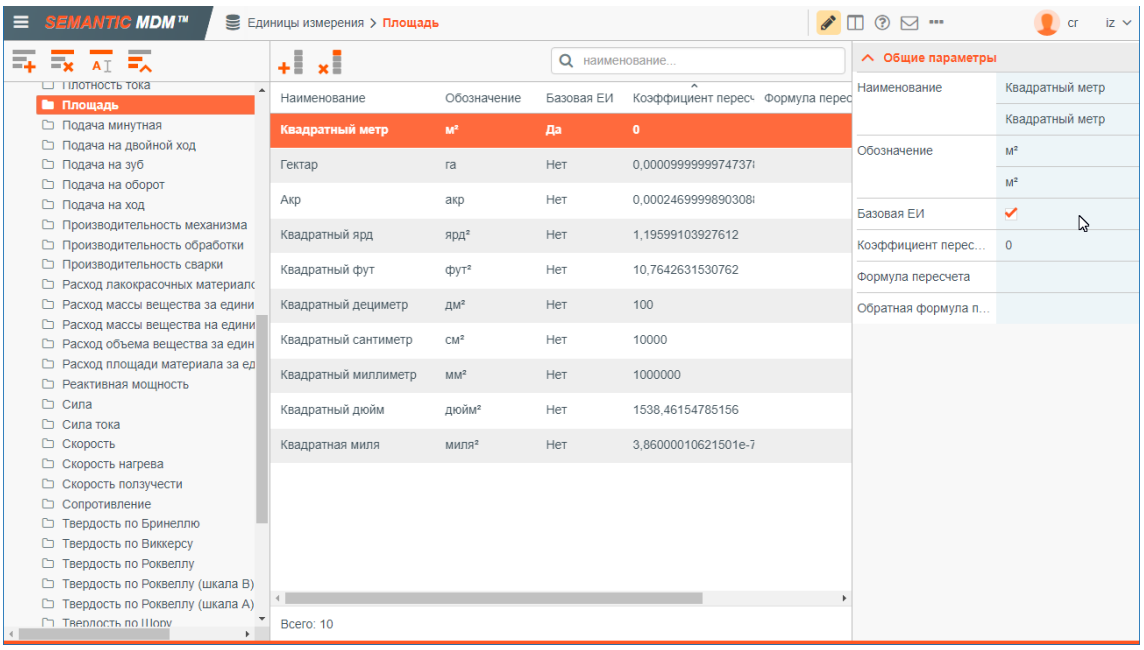


Рисунок 339 - Системный справочник *Единицы измерения*

Чтобы создать новую единицу измерения необходимо:

- перейти в системный справочник *Единицы измерения* и установить курсор на нужной группе (измеряемой сущности);
- в таблице нажать кнопку ***Добавить*** и перейти в карточку созданной ЕИ;
- в карточке заполнить наименование и обозначение ЕИ на русском и английском (опционально) языках;
- если создаваемая ЕИ является базовой для группы следует установить маркер **Базовая ЕИ** и коэффициент пересчета равный 0, в противном случае требуется ввести **Коэффициент пересчёта** или **Формулу пересчёта**;
- для случаев обратной трансформации из базовой в текущую ЕИ ввести **Обратную формулу пересчёта**.

Для справочника *Единицы измерения* доступна выгрузка и загрузка файлов формата XML. Доступ к командам **Экспорт в XML / Импорт в XML** выполняется через кнопку на основной панели инструментов (Рисунок 340).

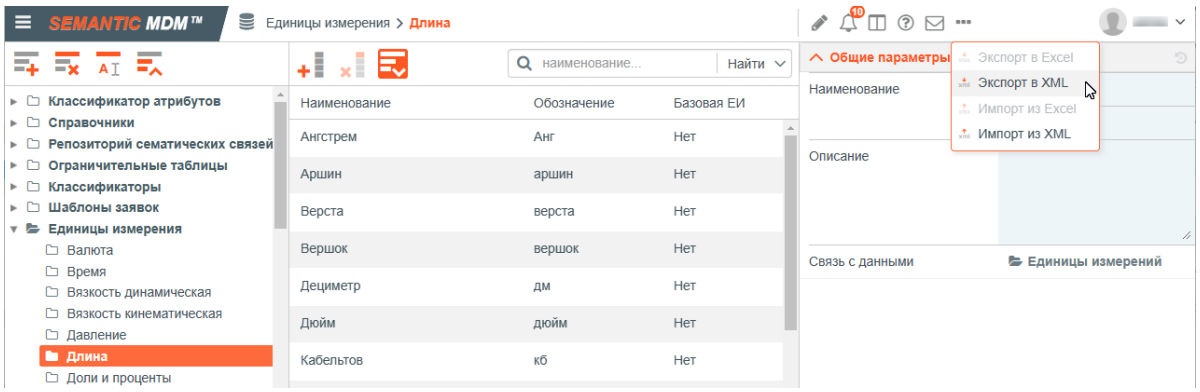


Рисунок 340 - Доступ к командам экспорта и импорта единиц измерения в файл XML

3.22 Репозиторий семантических связей. Стереотипы совместимости

Обеспечивает создание и хранение различных типов семантических связей (стереотипов совместимостей), например, аналогов. Семантические связи применяются для установления совместимости, в том числе вычисляемой по определенным заранее правилам, между различными объектами, например, сборочными единицами и входящими в них деталями, державками режущего инструмента и допустимыми для них типоразмерами режущих пластин и т.п.

Создание и просмотр стереотипа совместимости возможен с использованием **Репозитория семантических связей**, находящегося в представлении **Конфигурация данных**, либо в графической форме, которая доступна из раздела **Совместимости** карточки справочной группы**, а также в самом репозитории через кнопку *Диаграмма связей совместимости*** и в контекстного меню.

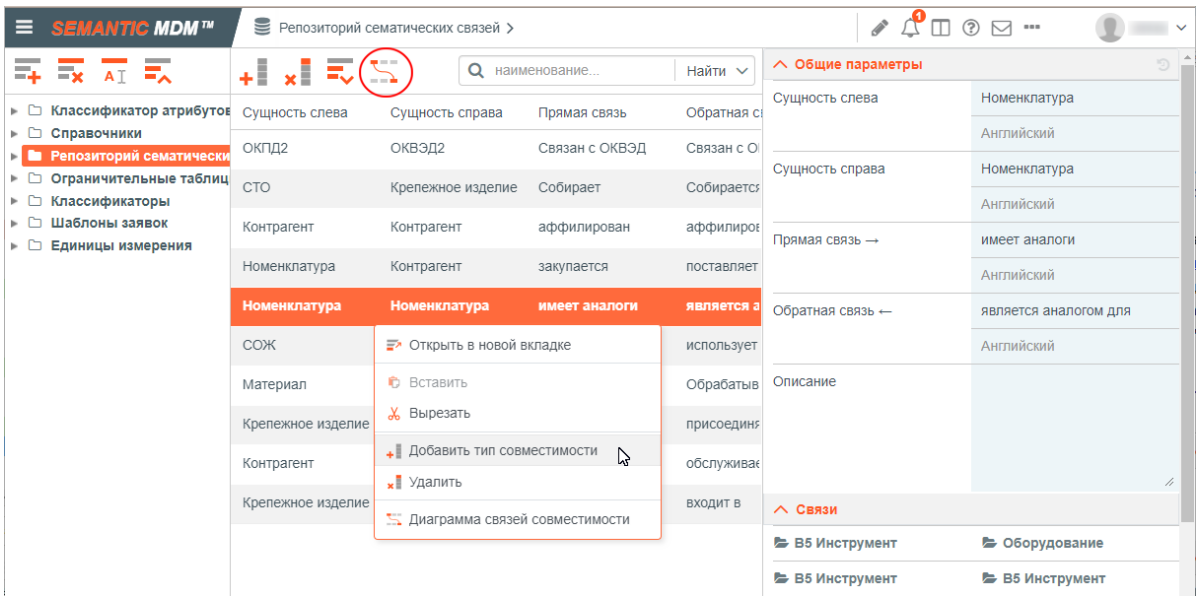


Рисунок 341 - Добавление нового стереотипа совместимости в **Репозитории семантических связей**

Для создания нового стереотипа совместимости в **Репозитории семантических связей** необходимо:

- в представлении **Конфигурация данных** перейти в раздел **Репозиторий семантических данных**;
- на сетке данных выбрать из контекстного меню команду **Добавить тип совместимости** или нажать кнопку ***Добавить*** над таблицей;
- в карточке созданной связи заполнить:
  - **Сущность слева** - общее наименование первого связываемого объекта (группы, позиции);
  - **Сущность справа** - общее наименование второго участника семантической связи (группы, позиции);
  - **Прямая связь** - наименование семантической связи в направлении слева - направо (от первого объекта ко второму). применяется для расширенного поиска. Например, *Деталь* ***входит в состав*** *-> Сборочной единицы*, здесь «входит в состав» - прямая связь;
  - **Обратная связь** - наименование семантической связи в направлении справа - налево (от второго объекта к первому).

применяется для расширенного поиска. Например, *Сборочная единица* ***включает в себя*** *->* *Деталь*, здесь «включает в себя» - обратная связь;

- **Описание** - текстовый комментарий к семантической связи, поясняющий её назначение и / или условия использования.

В подразделе **Связи** в карточке совместимости будут отображаться те группы и позиции, для которых подобная связь была установлена в справочниках.

Для создания нового стереотипа совместимости с использованием графического редактора необходимо:

- При создании из Репозитория семантических связей:
  - нажать кнопку Диаграмма связей совместимости;
  - выбрать команду Диаграмма связей совместимости из контекстного меню на сетке данных;
- При создании из карточки группы:
  - нажать кнопку *Диаграмма связей совместимости* в разделе Совместимости (Рисунок 342);



Рисунок 342 - Вызов диаграммы связей совместимости из карточки справочной группы

В открывшемся окне **Связи совместимости** пользователь может просматривать, редактировать и создавать новые стереотипы совместимости. Для этого можно воспользоваться функциями, представленными в Таблица 25.

Таблица 25 – Кнопки в окне «Связи совместимости»

Кнопка / Команда		Функция
	Увеличить масштаб	Позволяет увеличить масштаб отображения диаграммы
	Масштаб 1:1	Команда доступна только из контекстного меню. Обеспечивает отображение всех совместимостей в масштабе 1:1, при котором графический элемент имеет размер 34x10 мм (для одной связи)
	Уменьшить масштаб	Позволяет уменьшить масштаб отображения диаграммы
	Вписать в экран	Система обеспечивает такой масштаб диаграммы, при котором в окне будут видны все настроенные стереотипы совместимости
	Авторасстановка	Система располагает стереотипы совместимости автоматически в плоскости экрана. Позволяет "распутать" хаотично расположенные связи. При большом числе связей команда значительно увеличивает область для просмотра и соответственно уменьшает масштаб.

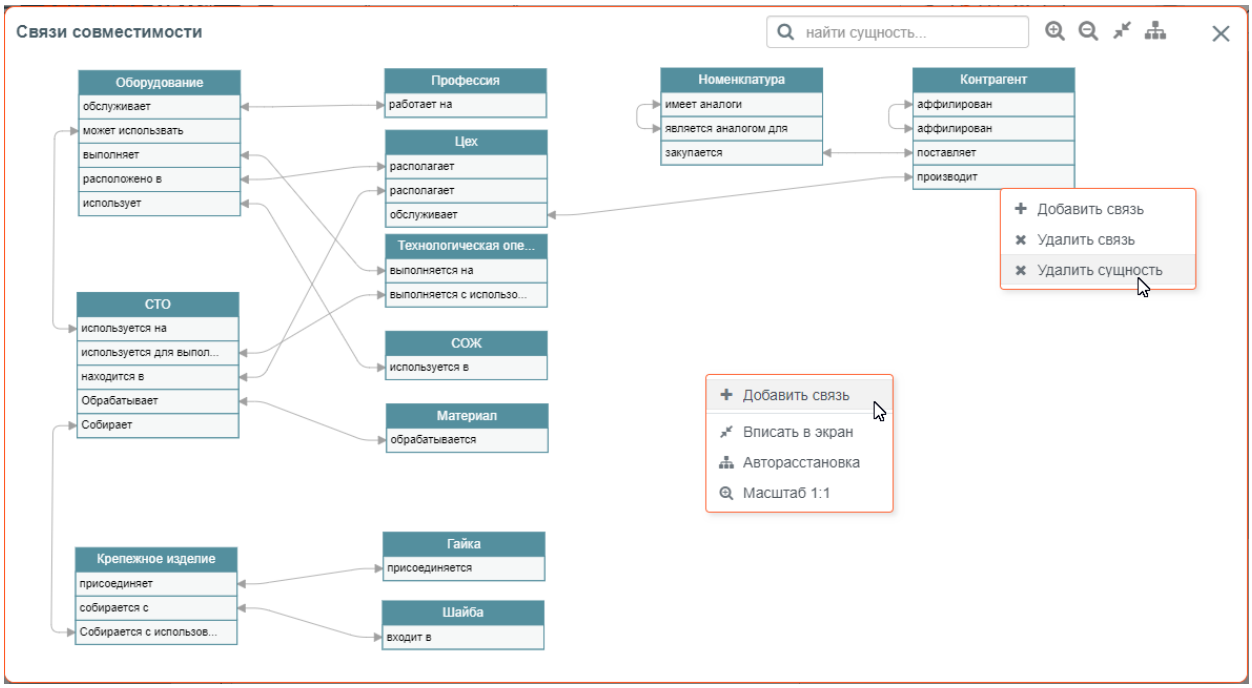


Рисунок 343 - Окно для просмотра и настройки связей совместимости в графическом виде

Команда **Добавить связь** вызывается из контекстного меню и позволяет создать новый стереотип совместимости, который представляет собой два прямоугольника, разделенных на две области и соединенных между собой двунаправленной стрелкой.

Выполнив двойной щелчок левой кнопкой мыши можно перейти в режим редактирования созданной совместимости. В верхней части прямоугольника указываются наименования сущностей (слева и справа соответственно), а в нижней наименование прямой (слева) и обратной (справа) связей. Каждая сущность может иметь неограниченное количество связей с другими сущностями, в том числе несколько видов связей между одними и теми же сущностями (Рисунок 344).

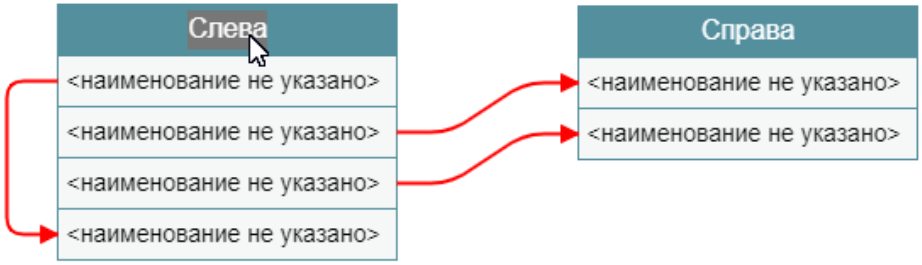


Рисунок 344 - Настройка совместимости

Если при создании новой совместимости *Система* обнаружит, что такие сущности уже есть, то она предложит подтвердить их слияние (Рисунок 345). При этом новая связь будет «дописана» к имеющейся сущности. Если слева и справа будет указана одна и та же сущность, то совместимость будет установлена внутри одной сущности - при этом стрелка будет соединять названия прямой и обратной связи.

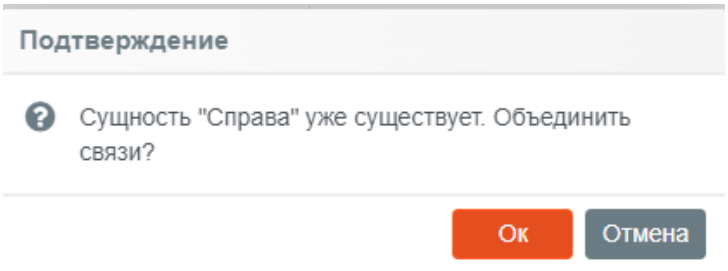


Рисунок 345 - Подтверждение слияния сущностей

Если пользователь не подтвердит слияния сущностей, то новая совместимость, как и связь создана не будет.

В случае изменения наименования сущности *Система* проверяет наличие других сущностей с идентичным наименованием. Если такие сущности есть, то будет затребовано подтверждение объединения сущностей со следующим сообщением: «Сущность с таким именем уже существует. Объединить связи?». Если пользователь подтвердит объединение, то будет выполнено объединение, путем удаления текущей сущности и переноса связей на существующую.

Удаление связи выполняется командой контекстного меню **Удалить связь**. Если для данной сущности связь была единственной, то вместе со связью будет удалена и сама сущность.

Команда контекстного меню **Удалить сущность** выполняет удаление сущности и всех её связей.

3.23 Интеграция Semantic MDM с Контур.Фокус

В *Системе* реализована интеграция с приложением *Контур.Фокус* (СКБ Контур) в части проверки данных контрагентов по *Единому государственному реестру юридических лиц* (ЕГРЮЛ). *Система* поддерживает два режима работы. Режим работы online, когда имеется постоянное, не имеющее ограничений подключение к внешней сети Интернет, а также режим работы когда внешняя и внутренняя (локальная) сети разделены между собой.

3.23.1 Работа в Контур.Фокус в режиме online

В режиме работы online с сервисом *Контур.Фокус* для создания новой позиции (контрагента) следует перейти в справочник, для которого настроена работа с сервисом (см. п. 3.31.2), после чего выполнить следующие действия:

- установить курсор на наименовании справочника или на любой позиции в таблице данных;
- нажать кнопку ***Загрузить контрагента из ЕГРЮЛ по ИНН***, расположенную над таблицей (Рисунок 346);

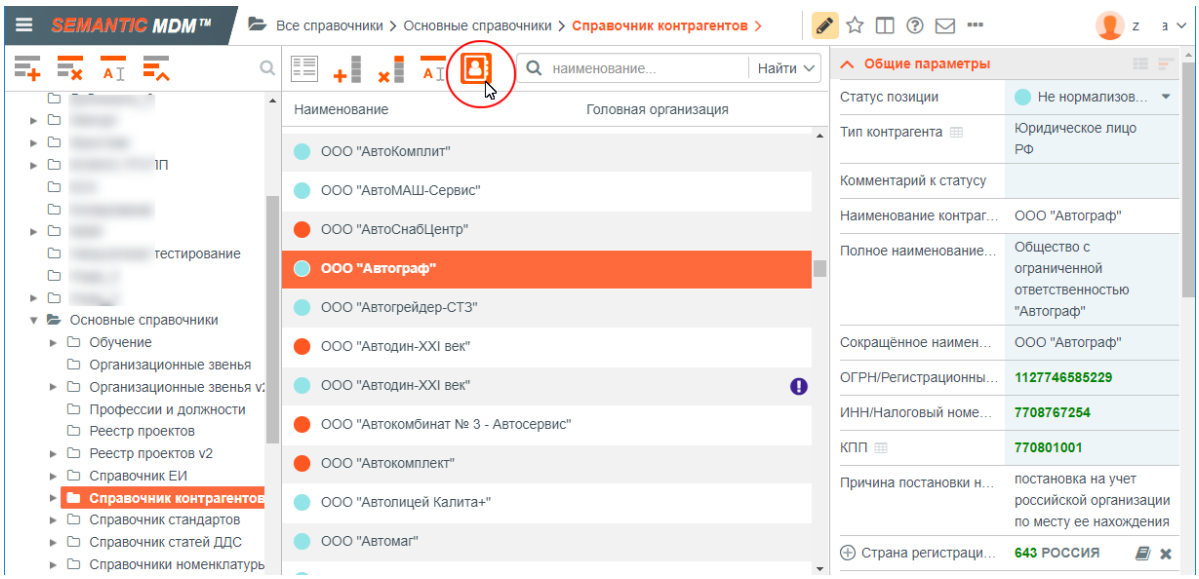


Рисунок 346 - Вызов команды загрузки контрагента из ЕГРЮЛ

- в появившемся окне **Импорт контрагента по ИНН** ввести ИНН и нажать кнопку ***Загрузить*** (Рисунок 347).

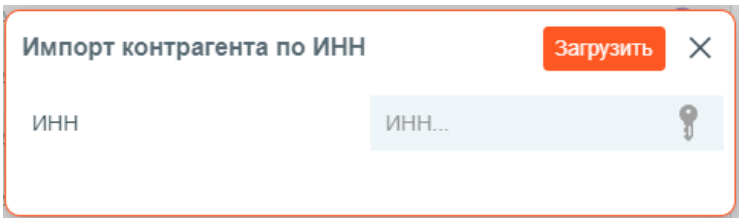


Рисунок 347 - Запрос на импорт контрагента по ИНН из *Контур.Фокус*

По нажатию на кнопку ***Загрузить*** система проверяет корректность ввода ИНН, после чего выполняет загрузку контрагента из *Контур.Фокус* и возвращает идентификатор импортированной позиции. После получения информации об импортированной позиции, курсор устанавливается в таблице данных на созданной записи. Если информацию не удалось получить из *Контур.Фокус*, то будет выдано сообщение с информацией об ошибке, которые была получена от сервиса *Контур.Фокус*.

Кроме загрузки новых данных можно в справочнике, настроенном для работы с *Контур.Фокус* можно провести консолидацию ранее созданных в нем записей с информацией из *Контур.Фокус*. Для этого необходимо:

- перейти в режим управления пакетами изменений;

- установить курсор на любой позиции в таблице данных либо выделить сразу несколько интересующих позиций (используя клавишу <Ctrl> при выделении);
- из контекстного меню выбрать команду **Консолидировать с контрагентом из ЕГРЮЛ**.

После этого *Система* получит значение из атрибута *ИНН*, проверит его на корректность, а затем выполнит загрузку контрагента из *Контур.Фокус* и вернет идентификатор импортированной позиции.

После загрузки будет открыт диалог дедубликации текущего контрагента с загруженным эталонным контрагентом, где эталоном является контрагент загруженный из *Контур.Фокус*, а дубликатом текущий выделенный контрагент.

3.23.2 Работа в условиях разделения внутренней и внешней сети

Последовательность работы с *Контур.Фокус* в случае эксплуатации изолированной внутренней сети предприятия, заключается в следующем:

- установить курсор на справочную группу с контрагентами и выбрать из контекстного меню команду **Интеграция с Контур.Фокус** ® **Сформировать файл с запросом данных** (Рисунок 348);
- либо установить курсор на позицию с записью конкретного контрагента и выбрать из контекстного меню команду **Интеграция с Контур.Фокус** ® **Запросить данные** (Рисунок 348).

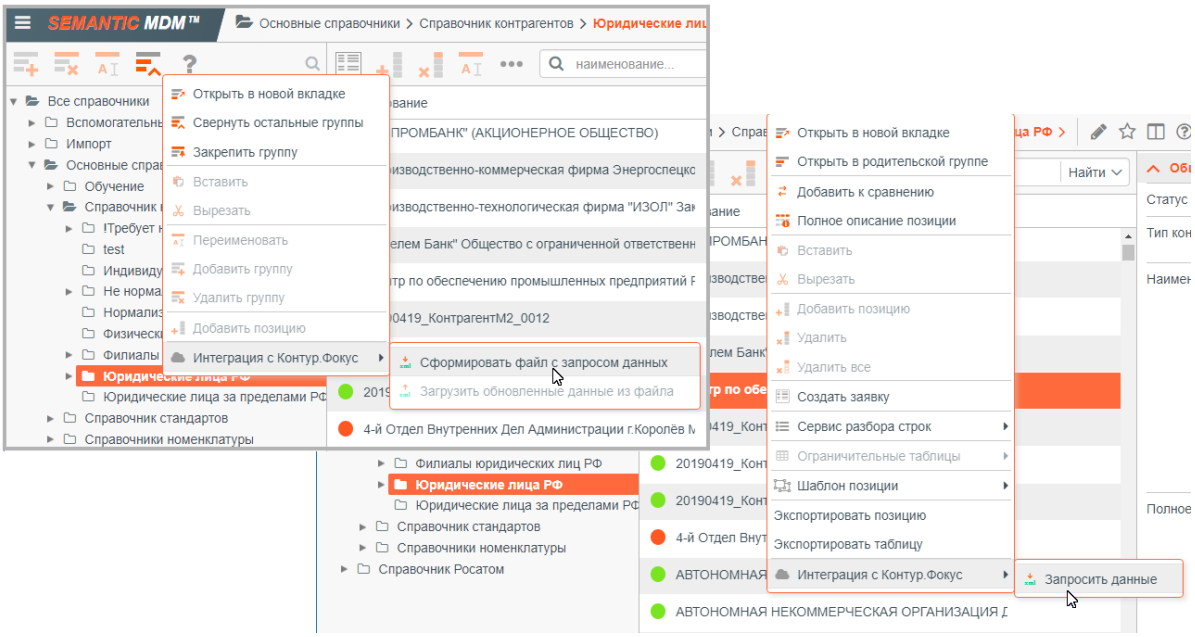


Рисунок 348 - Добавление новой семантической связи

После этого *Система* формирует файл и выгружает его на диск.

- выгруженный файл перенести на компьютер с доступом в сеть Интернет и, с помощью предустановленной программы *Контур.Фокус*, получить данные по контрагентам из ЕГРЮЛ;
- сохранить полученные данные в файл и перенести его на компьютер с *Semantic MDM*;
- выбирать из контекстного меню команду **Интеграция с Контур.Фокус ® Загрузить обновлённые данные из файла**.

Система загрузит данные по контрагентам и создаст новые позиции в справочнике *Контрагентов*.

Затем, при необходимости, может быть выполнена дедупликация контрагентов.

3.24 Интеграция справочников в САПР ТП Timeline

Semantic MDM интегрируется с системой технологический процессов Timeline (САПР ТП Timeline), что обеспечивает единое информационное пространство при конструкторском проектировании и технологической

подготовке производства на основе централизации и устранения дублирования объектов НСИ.

Основное назначение программы САПР ТП Timeline – проектирование технологических процессов для различных видов производств и формирование комплекта технологической документации в формате PDF в соответствии с ЕСТД.

Все необходимые для проектирования справочные данные хранятся в справочниках подсистемы Semantic MDM, из которых в диалоговом режиме при постоянном обращении технолога к базам НСИ в нужный момент времени извлекаются информационные объекты: оборудование, оснастка, инструменты и пр.

В результате интеграции на панели инструментов Timeline в режиме работы с техпроцессом технологу доступно окно Semantic MDM с возможностью поиска данных в справочниках: фильтрации, быстрого, фасетного и расширенного поиска по критериям, а также семантический поиск, при котором в качестве критериев выбора объектов можно задавать не только их параметры, но и взаимосвязи с другими объектами.

На Рисунок 349 показано рабочее окно программы Timeline. Интерфейс реализован в виде браузера – рабочие окна располагаются на отдельных вкладках.

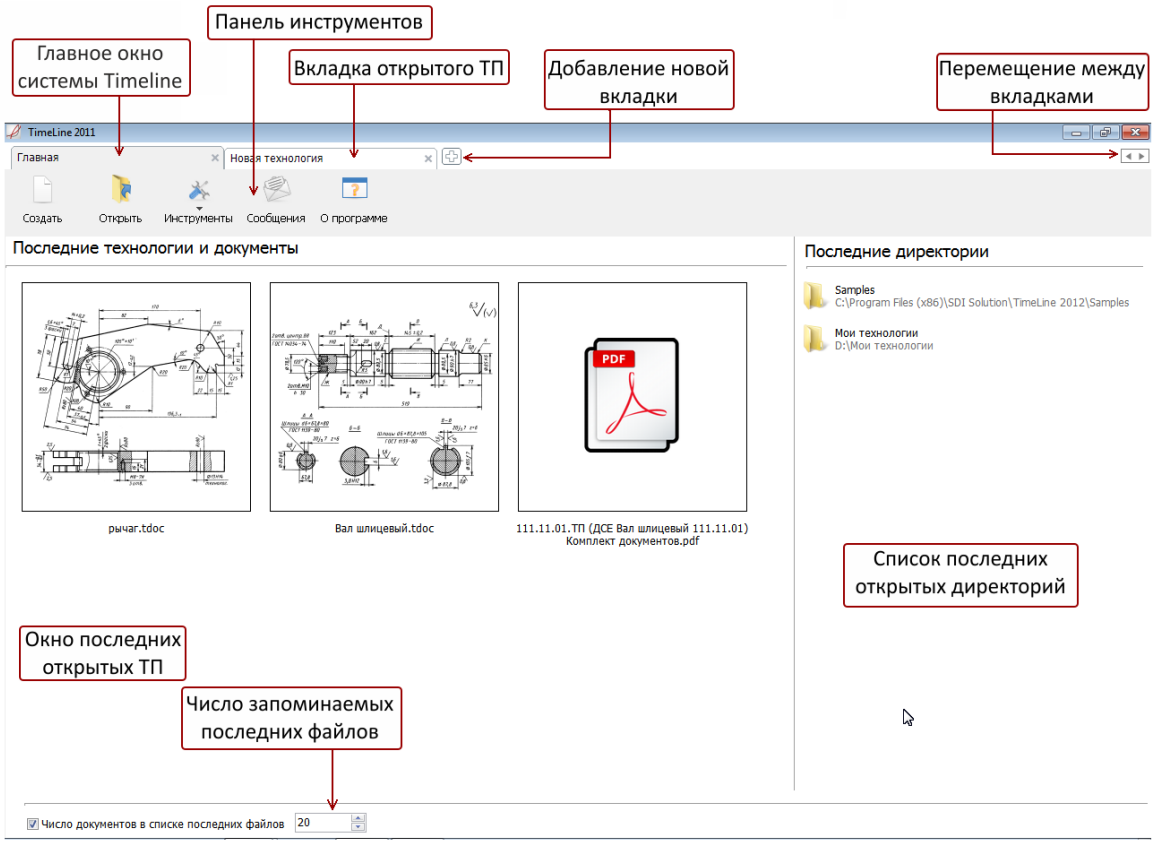


Рисунок 349 - Интерфейс главного рабочего окна

Для создания новой технологии на инструментальной панели САПР ТП Timeline расположена кнопка **Создать новый технологический процесс**, при нажатии на которую создаётся вкладка **Новая технология (**Рисунок 350**).**

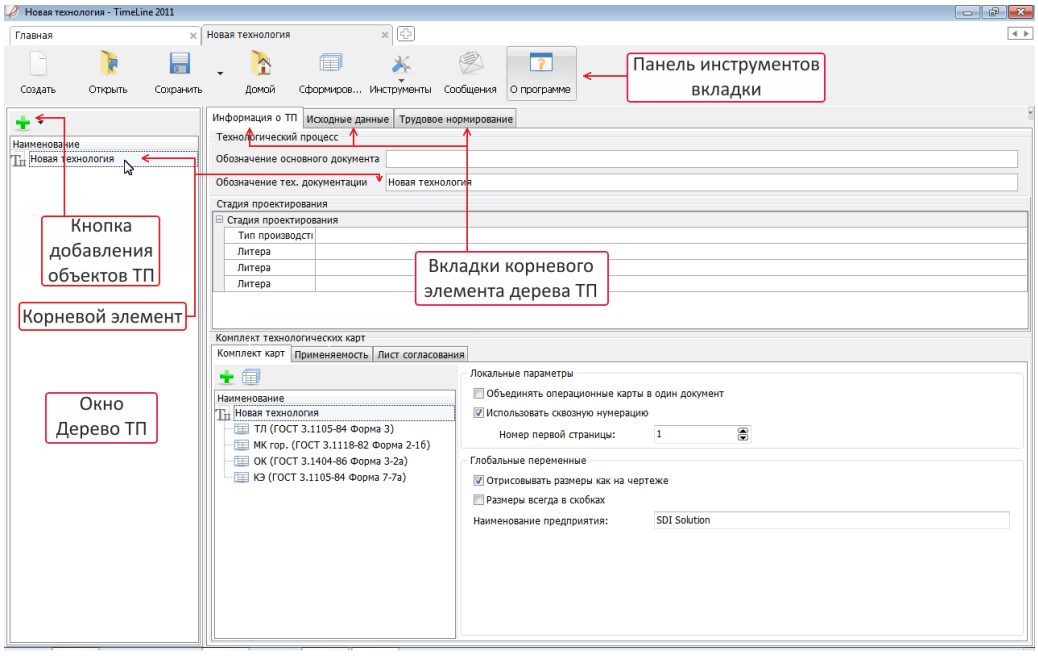


Рисунок 350 - Вкладка Новая технология

На вкладке **Новая технология** слева расположено окно **Дерево ТП**. В этом окне добавляются элементы техпроцесса – операции и переходы. Добавление элементов осуществляется с помощью кнопки **Добавить**, которая расположена на панели инструментов Дерева ТП.

На стадии проектирования ТП пользователю доступна отдельная вкладка Semantic MDM для получения данных из справочников в процессе выполнения следующих действий:

- Выбор и изменение типа производства;
- Добавление сведений о материале и заготовке;
- Создание новых операций в ТП;
- Редактирование операции;
- Создание новых переходов в операциях;
- Редактирование переходов;
- Добавление оборудования;
- Использование фильтрации;
- Добавление инструмента и оснастки.

3.24.1 Выбор и изменение типа производства

На вкладке Информация о ТП в разделе **Стадии проектирования** в ячейке *Тип производства* доступна кнопка вызова справочника ***Тип производства***, расположенная в правой части ячейки (Рисунок 351).

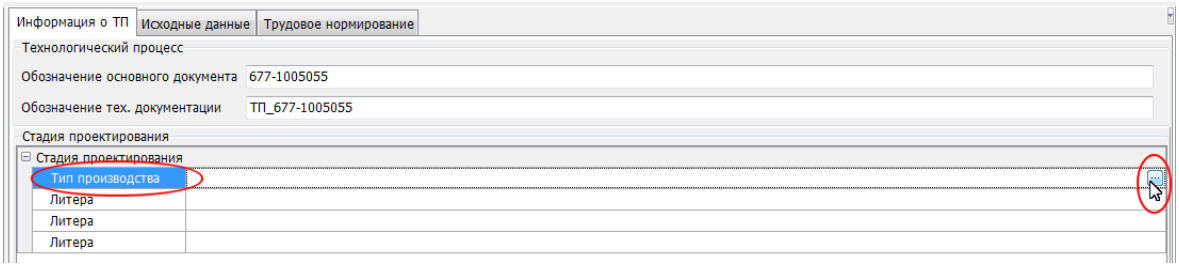


Рисунок 351 - Вызов справочника для указания типа производства

Для отображения значения типа данных в ячейке необходимо выбрать значение типа производства в *Таблице данных* в открывшемся окне

справочника ***Тип производства***, нажав кнопку **Выбрать** в правом нижнем углу окна или кнопку ✔ над таблицей данных.

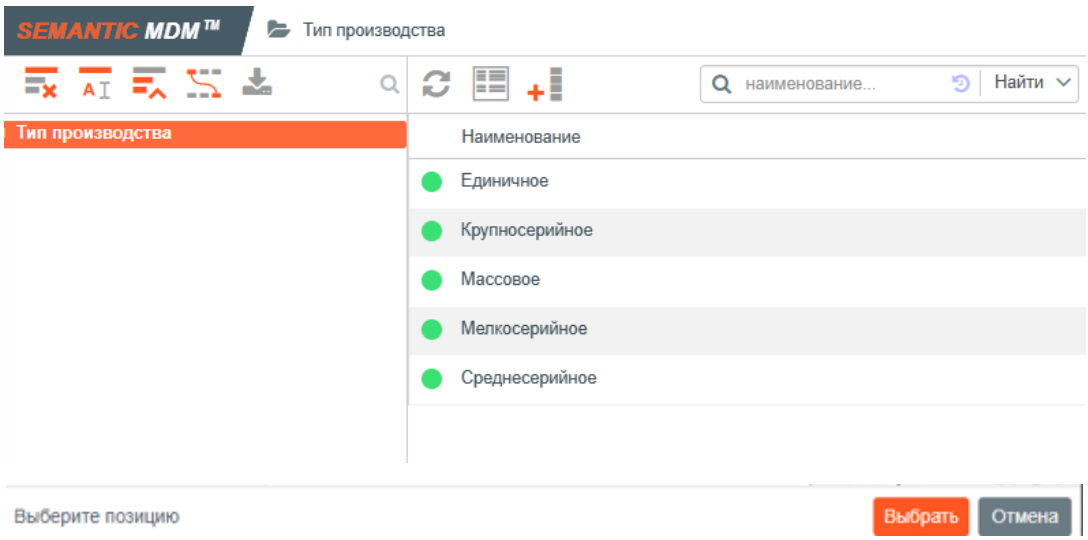


Рисунок 352 - Окно справочника **Тип производства**

Для изменения типа производства необходимо в повторно вызвать справочник и выбрать другое значение.

3.24.2 Добавление сведений о материале и заготовке

На инструментальной панели на вкладке **Исходные данные** нажать кнопку **Добавить** и выбрать ***Добавить заготовку*** (Рисунок 353 - Добавление сведений о заготовке.

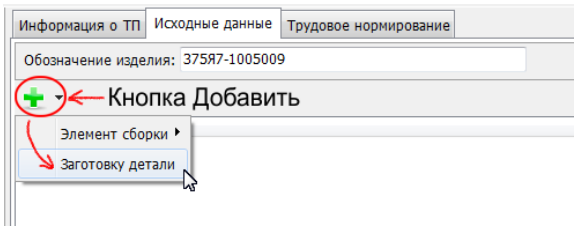


Рисунок 353 - Добавление сведений о заготовке

В открывшемся окне справочника **Материалы** найти материал заготовки (см. раздел 3.7 Поиск данных) и нажать кнопку **Выбрать** в правом нижнем углу окна или кнопку ✔ над Таблицей данных.

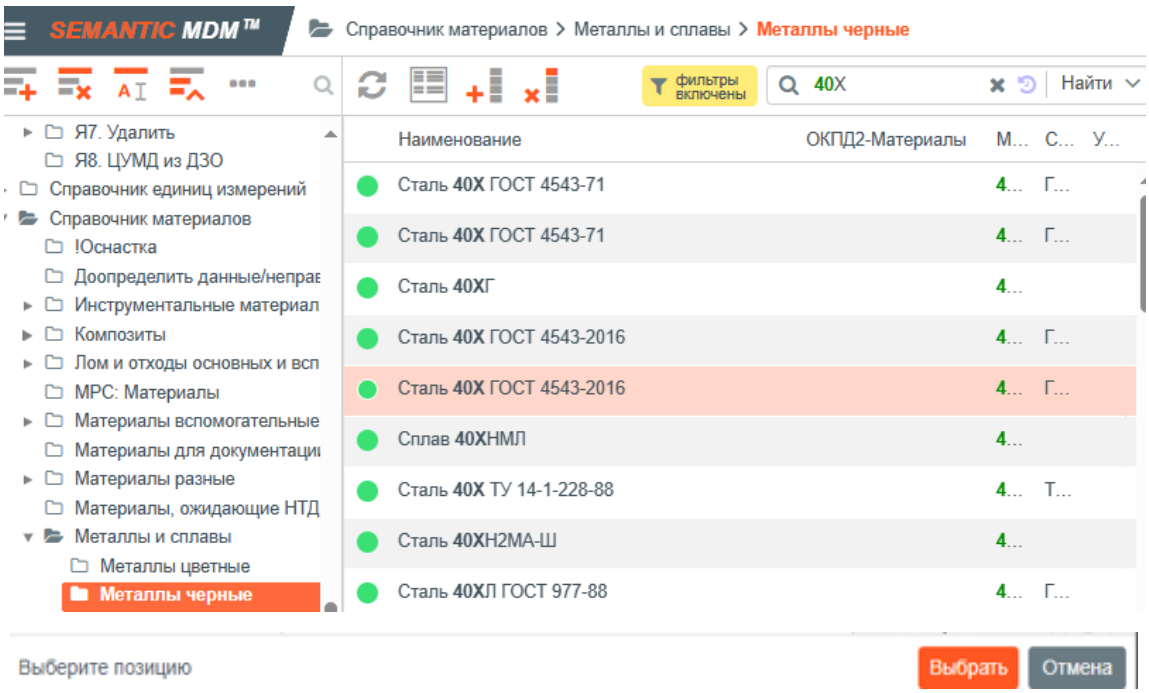


Рисунок 354 - Выбор материала заготовки

Для редактирования сведений о материале следует нажать кнопку **Выбрать заготовку** (Рисунок 354 - Выбор материала заготовки. Будет вновь открыт справочник материалов в Semantic MDM. Выбрать новый материал.

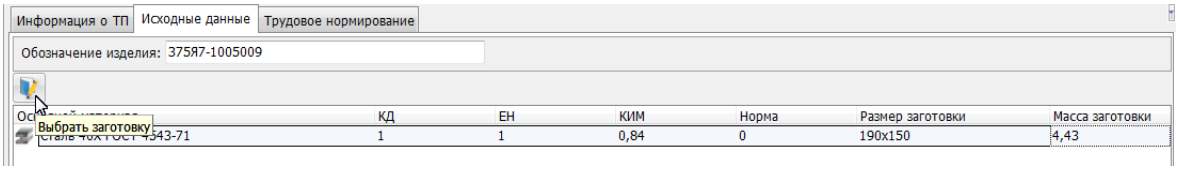


Рисунок 355 - Изменение сведений о заготовке

3.24.3Создание новых операций в ТП с использованием справочников

Для создания технологии используется окно Дерево ТП (Рисунок 350). В этом окне добавляют, удаляют и редактируют элементы структуры техпроцесса.

Для добавления новой операции в инструментальном меню **Дерева ТП** нажать кнопку **Добавить операцию** если курсор стоит на наименовании техпроцесса, или **Добавить – Добавить операцию** из контекстного меню, если курсор стоит на операции.

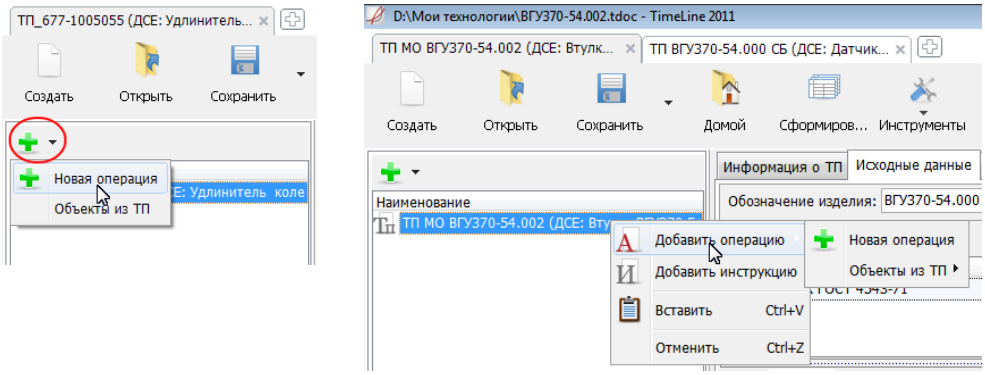


Рисунок 356 - Панель **Добавить операцию**

В выпадающем списке доступны команды:

Новая операция – позволяет добавить из базы новую операцию в технологический процесс;

Объекты из ТП – в выпадающем списке представлены для добавления операции, которые были ранее добавлены в текущий ТП. Операция добавляется в ТП без вызова Semantic MDM.

В открывшемся окне Semantic MDM – Технологические операции найти нужную операцию и нажать кнопку **Выбрать** в правом нижнем углу окна или кнопку ✔ над Таблицей данных.

Операции сгруппированы по виду обработки. Поиск нужной операции осуществляется средствами поиска Semantic MDM.

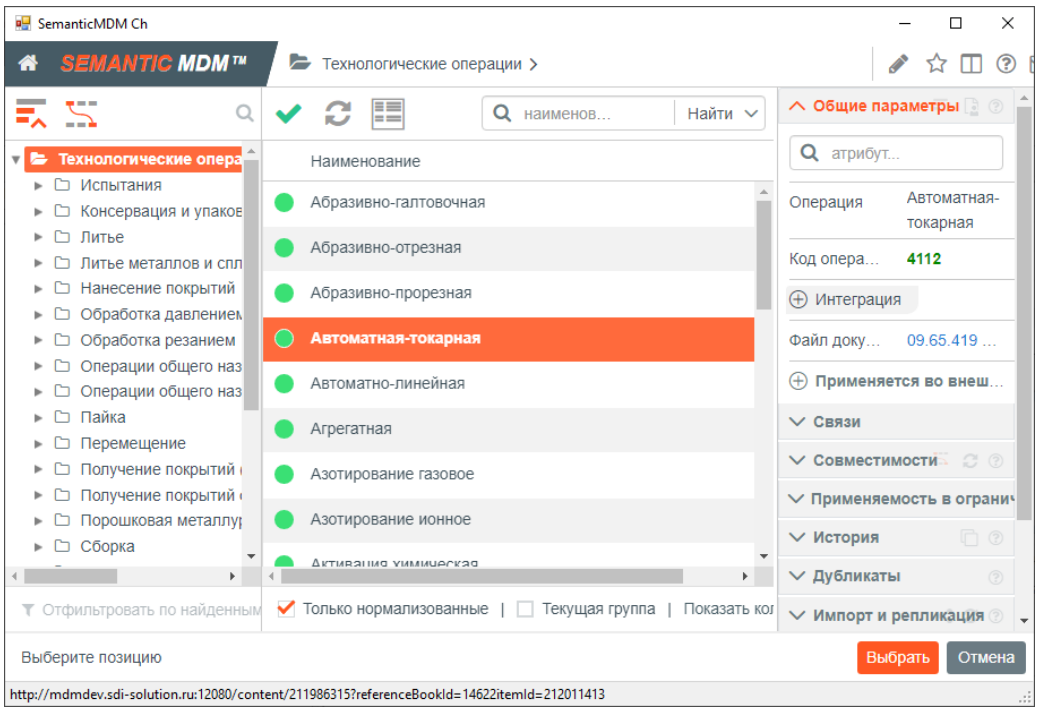


Рисунок 357 - Выбор операции в ТП

Выбранная операция добавлена в технологический процесс.

3.24.4 Редактирование операции

Любая операция ТП может быть отредактирована. Для этого используется панель инструментов дерева ТП, контекстное меню или вкладка **Информация об операции.**

Для изменения операции установить курсор на наименовании операции в *Дереве ТП* и нажать на кнопку **Изменить**, далее выбрать новую операцию.

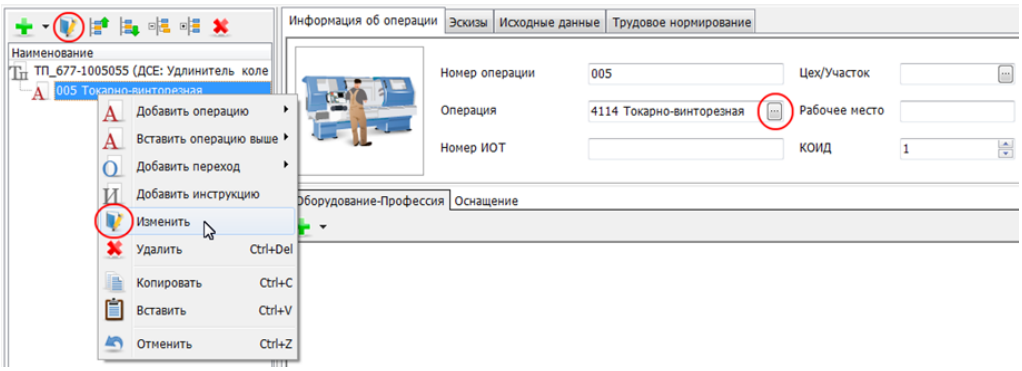


Рисунок 358 - Доступ к инструментам редактирования операций

3.24.5 Создание новых переходов в операциях

Для добавления нового перехода в операцию, нужно установить курсор на наименовании операции в *Дереве ТП*. В инструментальной панели Дерева ТП нажать кнопку **Добавить – Добавить переход**. В выпадающем меню кнопки **Добавить переход** доступны команды ***Новый переход*** и ***Объекты из ТП***, аналогично командам для добавления операций в ТП, либо в контекстном меню операции в дереве (рисунок ниже).

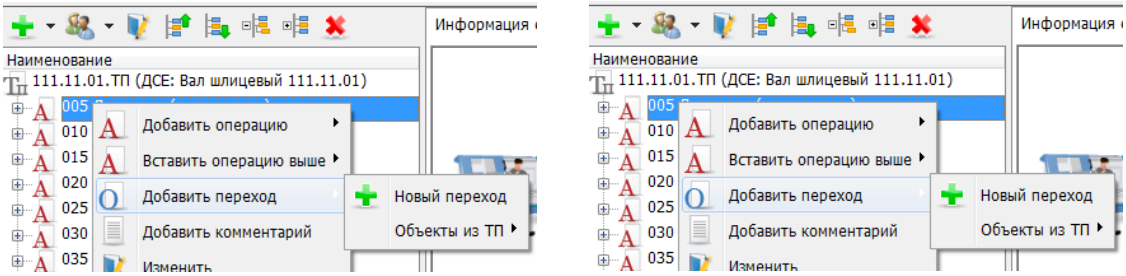


Рисунок 359 - Добавление перехода

В открывшемся окне Semantic MDM (Рисунок 360 - Главное окно справочника переходов доступны два раздела **Основные переходы** и **Вспомогательные переходы**. Двойной щелчок на любом из них открывает соответствующий справочник.

Порядок действий при добавлении в операцию основного или вспомогательного перехода одинаковый.

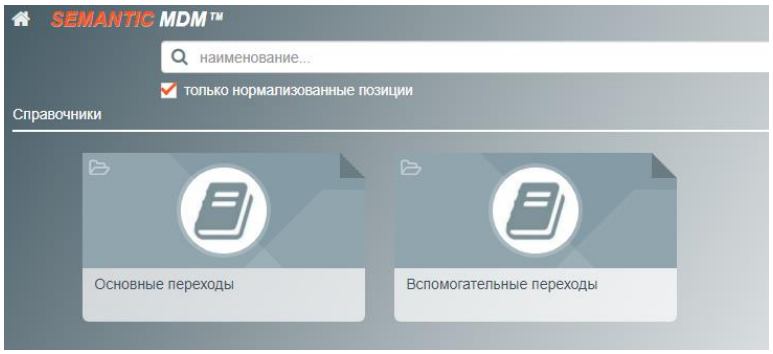


Рисунок 360 - Главное окно справочника переходов

Найти требуемый переход в справочнике и нажать кнопку **Выбрать** в правом нижнем углу окна или кнопку ✔ над таблицей данных. В операцию был добавлен новый переход.

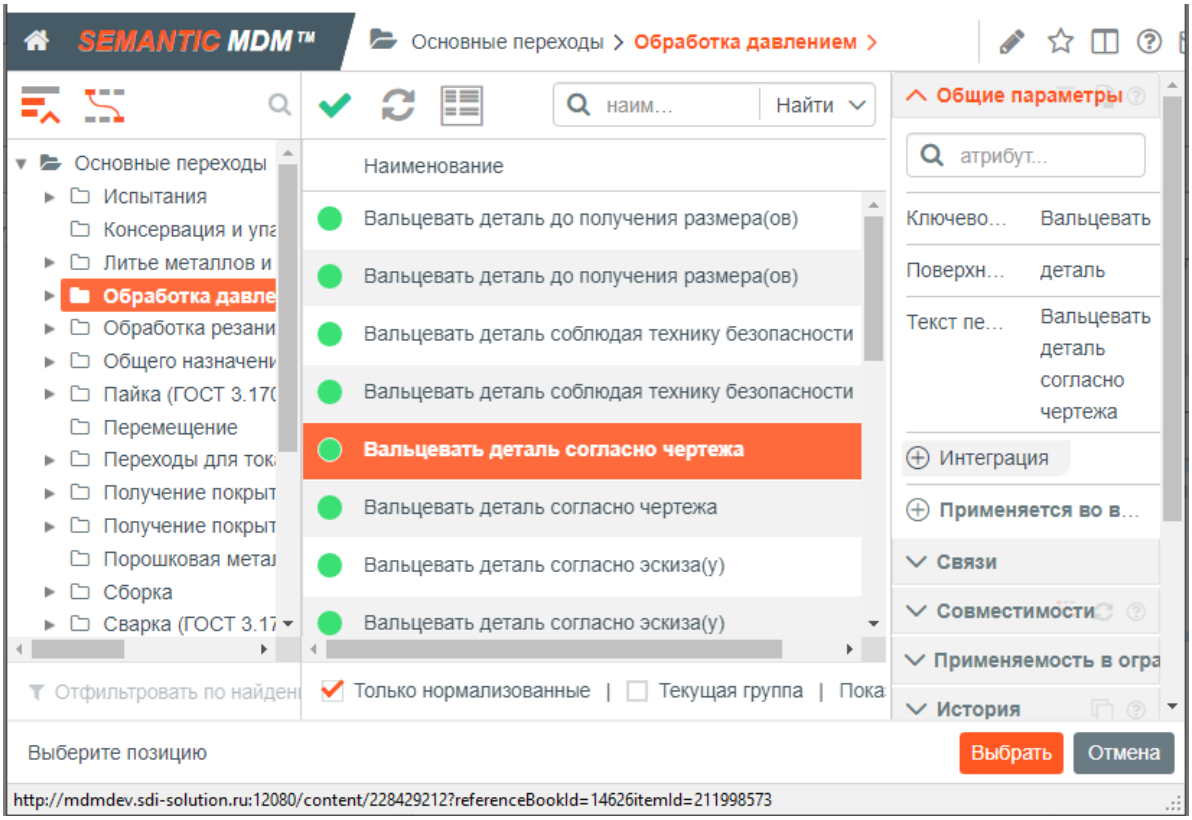


Рисунок 361 – Добавление основного перехода в операцию

Открытый справочник будет содержать только те переходы, которые возможны для данной операции. Чтобы увидеть все возможные переходы нужно отключить фильтрацию, нажав фильтры включены на панели инструментов справочника.

Настройка связей между справочниками и вариантов фильтрации объектов производится администратором системы в Semantic MDM

3.24.6 Редактирование переходов

В случае необходимости можно отредактировать текст перехода. Для этого нужно установить на нем курсор в Дереве ТП и перейти на вкладку *Текст перехода*.

Как и в случае с операцией функцию редактирования можно вызвать кнопкой **Редактировать** на панели инструментов дерева, из контекстного меню или из панели инструментов вкладки *Текст перехода*.

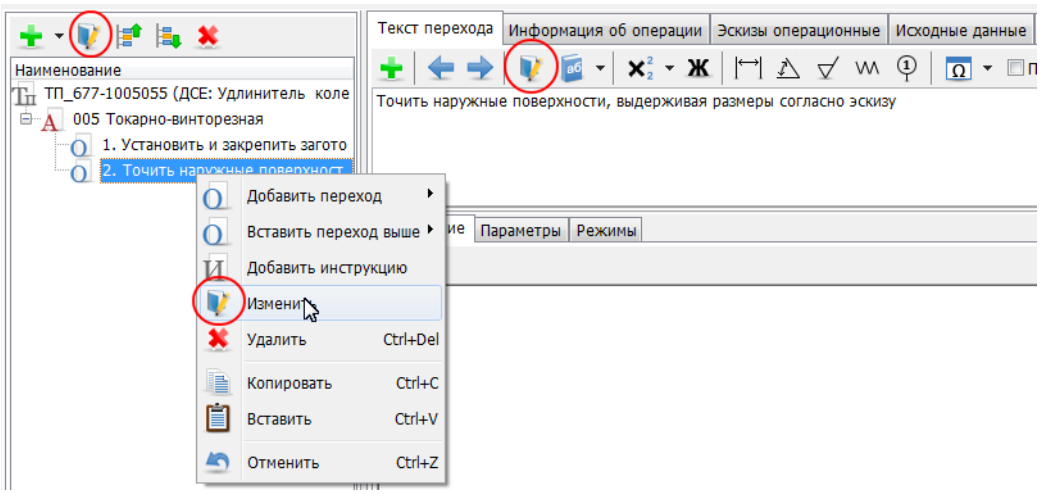


Рисунок 362 - Вызов функции редактирования перехода

Функция редактирования запускает справочник переходов, давая возможность изменить текст используя предлагаемые варианты.

3.24.7 Добавление оборудования в операцию

Установить курсор на наименовании операции в *Дереве ТП* и перейти на вкладку **Оборудование-Профессия** и нажать на панели кнопку **Добавить.** В выпадающем меню выбрать команду **Новый элемент (**рисунок ниже**).**

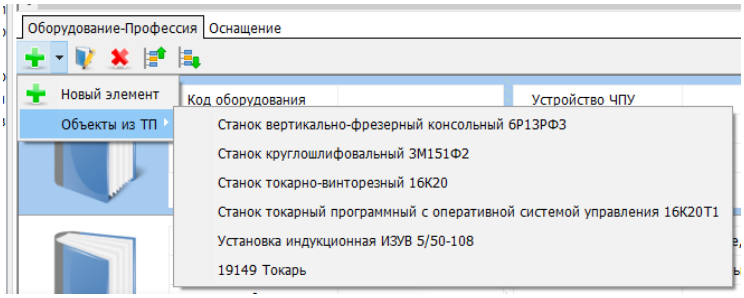


Рисунок 363 - Вызов окна справочника Semantic MDM

В открывшемся окне Semantic MDM дважды кликнуть на пиктограмме ***Оборудование***.

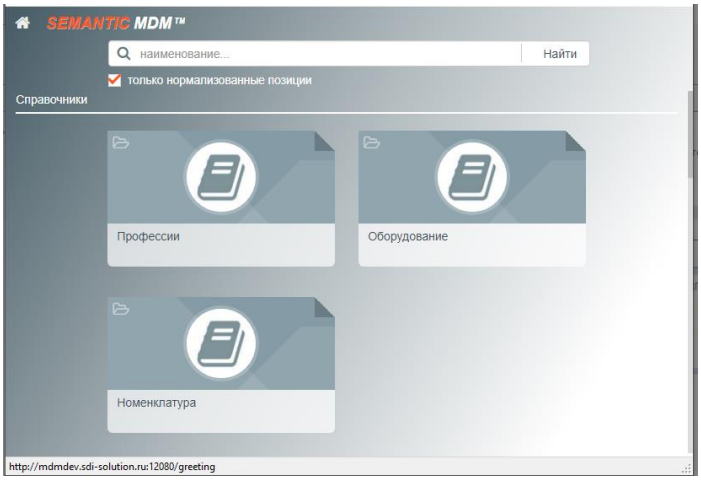


Рисунок 364 – Окно Semantic MDM для выбора оборудования

В открывшемся справочнике Оборудование, располагаются группы оборудования, с отфильтрованными позициями, на которых возможно выполнение выбранной операции. В окне справа на вкладке Атрибуты можно ознакомиться с краткой технической характеристикой выбранного оборудования.

Выбрать требуемое оборудование из справочника и нажать кнопку Выбрать в правом нижнем углу окна или кнопку ✔ над *Таблицей данных*.

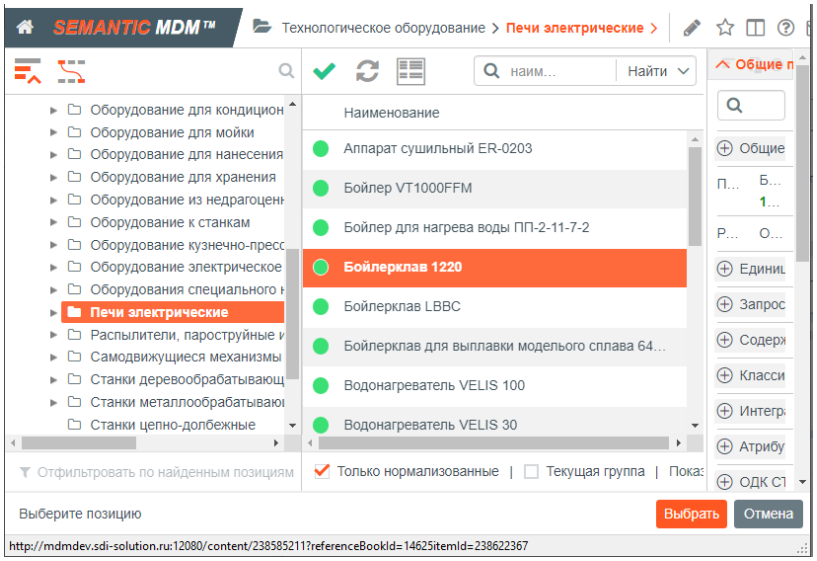


Рисунок 365 - Выбор оборудования

3.24.8 Добавление оснастки

Добавление приспособлений, мерительного и вспомогательного инструмента и материалов производится на вкладке **Оснащение**, расположенной радом с вкладкой **Оборудование-Профессия**. На вкладке может быть добавлено столько экземпляров оснастки, сколько требуется пользователю.

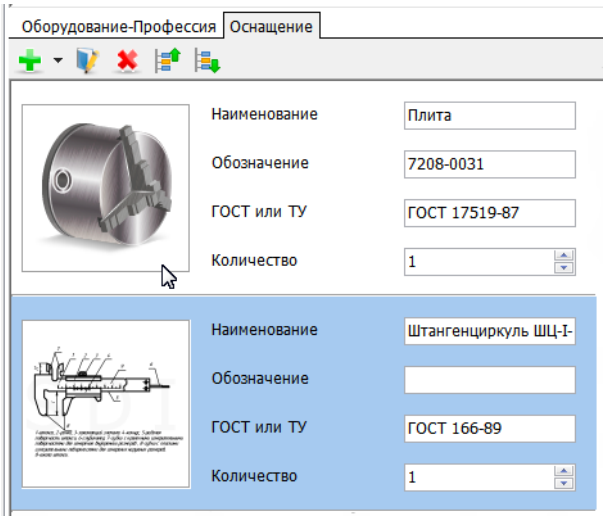


Рисунок 366 -Вкладка оснащение с экземплярами оснастки

Добавление оснащения в техпроцесс выполняется аналогично добавлению оборудования.

3.25 Отчеты

В *Semantic MDM* присутствует интеграция с подсистемой отчетов SemanticReports. Интеграция позволяет разработать отчет с использованием инструментария SemanticReports и организовать его формирование и просмотр в специальном представлении **Отчёты**. В данном разделе *Системы* доступны шаблоны отчетов для формирования, а также ранее сформированные отчеты. Подсистема отчетов доступна только для пользователей, которые имеют одну из привилегий:

- Формирование отчетов;
- Менеджер заявок;

- Системный администратор.

Все доступные в подсистеме отчеты условно делятся на два вида:

- Системные – вспомогательные отчеты, которые нужны для отображения информации в панелях dashboard Semantic или используются в пользовательских отчетах;
- Пользовательские – отчеты, предназначенные для использования пользователями.

Пользовательские отчеты делятся на следующие категории:

- Отчеты по заявкам – данные отчеты доступны для формирования в подсистеме **Журнал заявок** (модуль управления заявками (МУЗ) на вкладке **Отчеты**;
- Отчеты по импорту и репликации – данные отчеты доступны для формирования в подсистеме **Импорт и репликация** (модуль импорта и репликации (МИР) на вкладке **Отчеты**;
- Иные категории – это категории, которые определяются пользователями и позволяют в подсистеме отчетов выполнять сортировку и фильтрацию отчетов по категориям, для удобства использования.

Чтобы отчеты SemanticReports были доступны в подсистеме отчетов Semantic MDM, требуется выполнить определенные настройки, описание данных настроек представлено в п. 3.31.7.

3.25.1 Формирование отчетов

Для формирования отчетов по разным аспектам деятельности сотрудников в *Semantic MDM* следует перейти в представление **Отчёты** на вкладку **Доступные** (Рисунок 367).

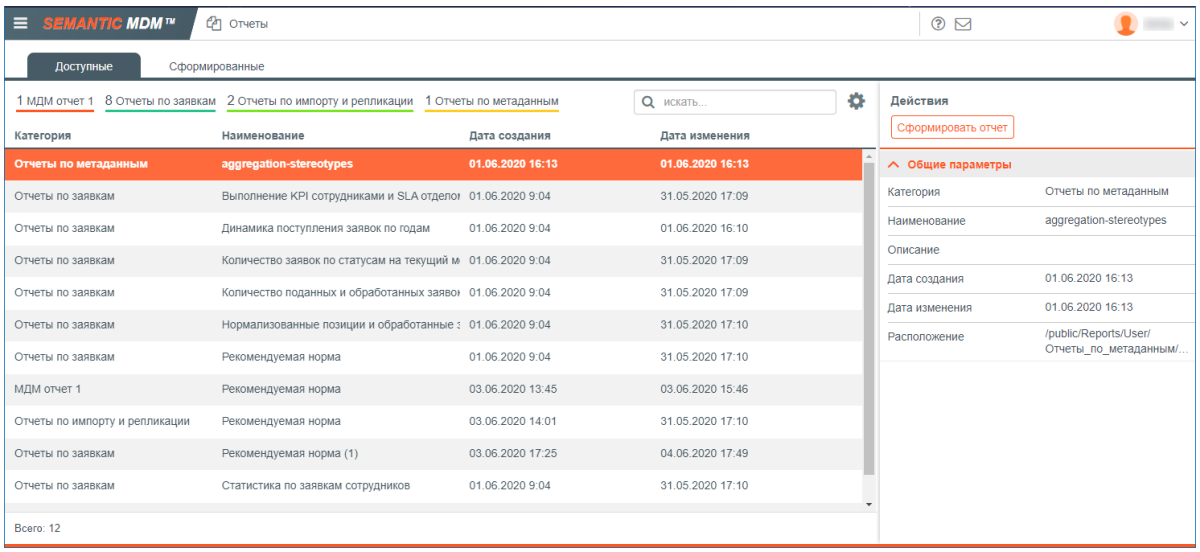


Рисунок 367 - Вкладка **Доступные** в представлении **Отчёты**

На вкладке **Доступные** представлена таблица с загруженными в *Систему* шаблонами отчетов, которые могут быть сформированы по запросу пользователя. Над таблицей находится строка, в которой расположены фильтры для доступа к различным категориям отчетов с указанием количества отчетов в каждой категории. Здесь же находится ячейка быстрого поиска, а также кнопка ***Настройки таблицы***, которая позволяет настроить количество видимых столбцов на сетке данных. В таблице также доступен фасетный поиск и сортировка по столбцам.

Справа от таблицы располагается карточка отчета с его описанием и адресом, где расположен шаблон. Над карточкой находится кнопка ***Сформировать отчет***, нажатие на которую запускает процедуру формирования отчета. В зависимости от настроек шаблона пользователю может быть предложено выбрать дополнительные атрибуты для создания отчета - временной период, справочную группу, пользователей и т.д.

Таким образом, для формирования отчета следует установить на нем курсор в таблице и нажать кнопку ***Сформировать отчет***. После этого выбрать дополнительные значения для формирования и нажать кнопку ***Сгенерировать*** (Рисунок 368). Готовый отчет будет размещен на вкладке **Сформированные**.

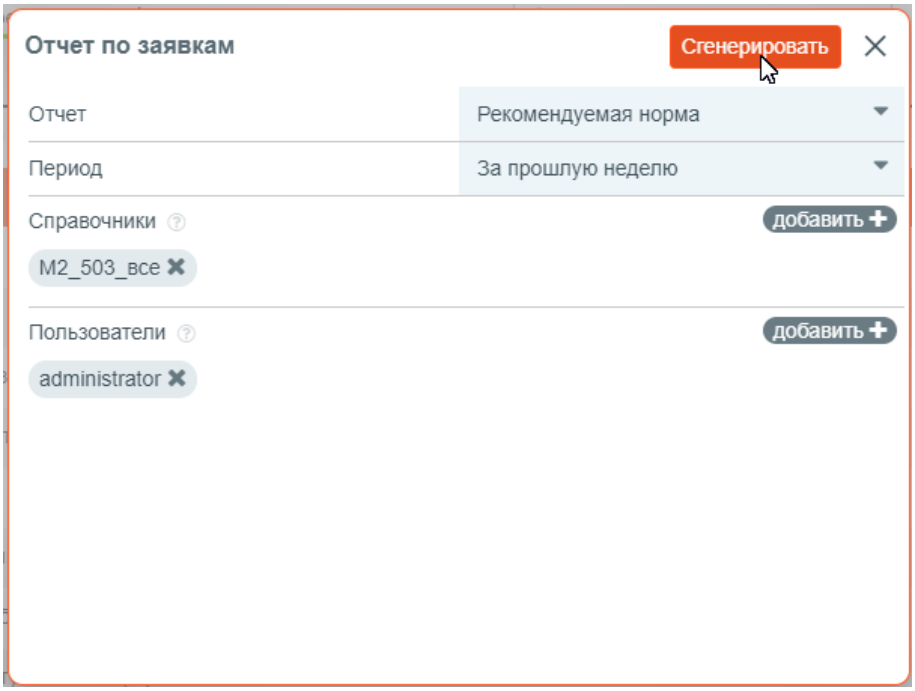


Рисунок 368 - Настройка отчета и запуск генерации

3.25.2 Просмотр и выгрузка отчетов

Готовые отчеты располагаются на вкладке **Сформированные** (Рисунок 369) представления **Отчёты**.

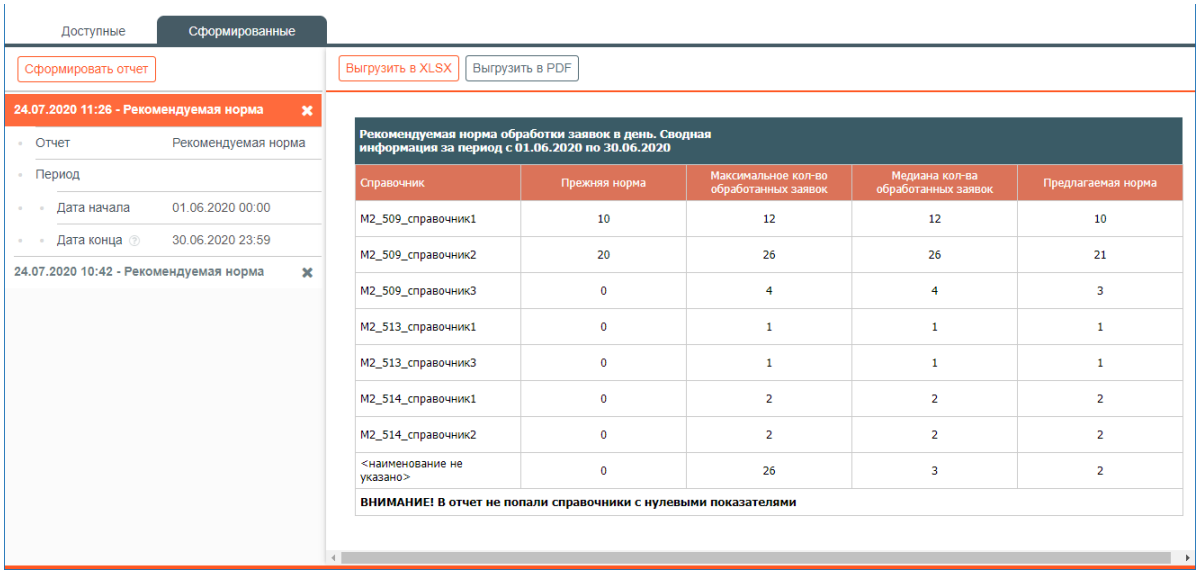


Рисунок 369 - Просмотр сформированных отчетов

В левой части вкладки располагается список готовых отчетов с указанием даты и времени формирования отчета. Установив курсор на

интересующем отчете в списке можно ознакомится с дополнительными значениями атрибутов отчета, а в правой части вкладки с содержанием отчета.

Ненужные отчеты могут быть удалены при помощи кнопки ✖, находящейся в строке с названием отчета. Для удаления необходимо нажат кнопку ***Ок*** во всплывающем окне подтверждения (Рисунок 370).

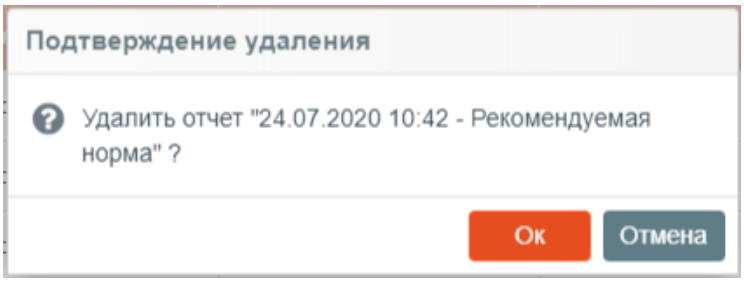


Рисунок 370 - Подтверждение удаления отчета

Сформированный отчет может быть выгружен в виде файла Excel или в формате PDF. Для этого над полем отчета располагаются две кнопки - ***Выгрузить в XLSX*** и ***Выгрузить в PDF*** соответственно.

3.25.3 Отчёты по заявкам

На вкладке **Отчёты** подсистемы **Журнал заявок** можно генерировать различные виды отчётов по заявкам, в том числе отчет по заявителям, исполнителям и согласующим, а также выполнение KPI сотрудниками, количество нормализованных позиций и обработанных заявок, статистика по заявкам сотрудников, количество поданных и обработанных заявок.

Для того, чтобы сформировать отчёт следует:

- в подсистеме **Журнал заявок** перейти на вкладку **Отчёты**;
- нажать кнопку ***Сформировать отчёт***;

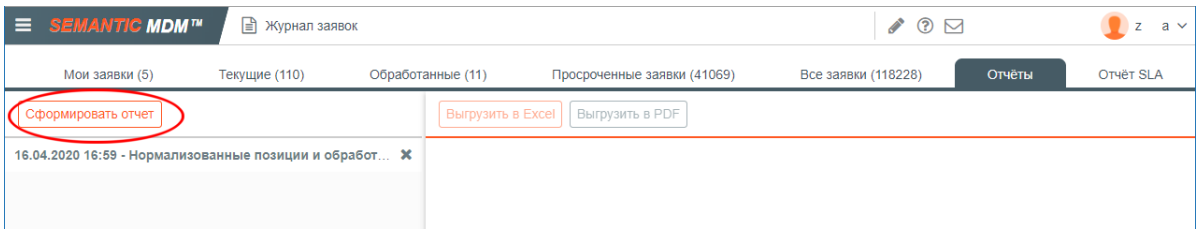


Рисунок 371 - Вкладка **Отчёты**

- в открывшемся окне **Отчёт по заявкам** (Рисунок 372) из выпадающего списка выбрать тип отчета;

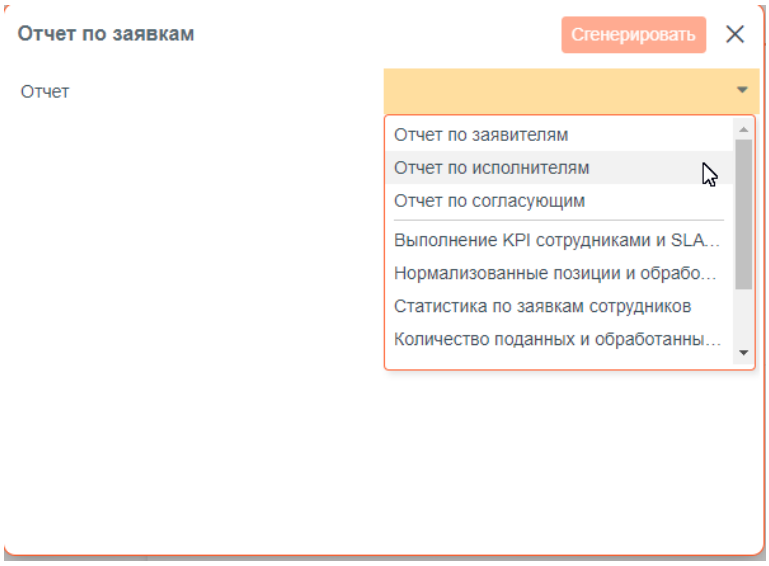


Рисунок 372 - Окно **Отчёт по заявкам** (выбор типа отчета)

В зависимости от выбранного типа отчета пользователю будут доступны разные настройки.

Для отчетов по заявителям, исполнителям и согласующим отличия будут только в части доступных для настройки полей отчета (Рисунок 373).

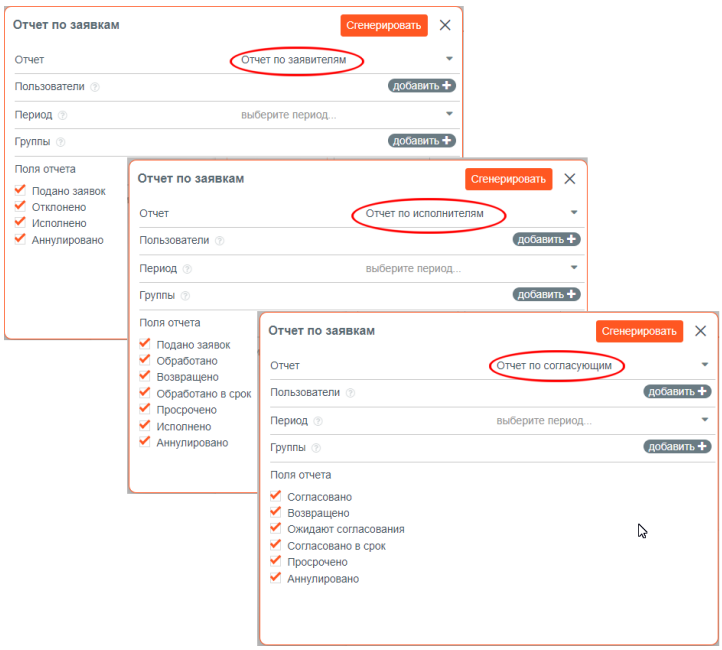


Рисунок 373 - Настройка отчетов по видам пользователей

- **Пользователи** – с помощью кнопки ***Добавить*** можно открыть список пользователей и выбрать необходимых. Для выбора пользователя достаточно установить маркер напротив его имени (логина) и нажать кнопку ***Ок*** (Рисунок 374);

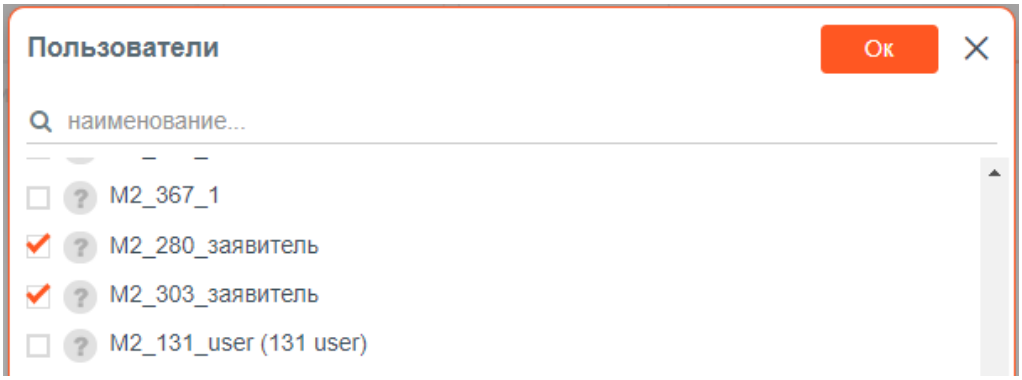


Рисунок 374 - Выбор пользователей для получения отчёта

- Указать **Период** – время, за которое требуется сформировать отчёт. Для выбора доступны – конкретная дата, интервал, а также варианты «*вчера*», «*за прошлую неделю*», «*за эту неделю*», «*за прошлый месяц*» и т.п.;

- в разделе **Группы** можно добавить (используя дополнительное окно группы, которое вызывается кнопкой ***Добавить***) справочные группы, по которым подавались и выполнялись заявки;
- в разделе **Поля отчёта** маркерами отмечаются интересующие сведения. Набор сведений зависит от роли пользователя.

Для *Заявителя*:

- Количество поданных заявок;
- Количество отклонённых заявок;
- Количество исполненных заявок;
- Количество аннулированных заявок;

Для *Исполнителя*:

- Количество поданных заявок;
- Количество обработанных заявок;
- Количество возвращённых заявок;
- Количество заявок, обработанных в срок;
- Количество просроченных заявок;
- Количество исполненных заявок;
- Количество аннулированных заявок;

Для *Согласующего*:

- Количество согласованных заявок;
- Количество возвращённых заявок;
- Количество заявок, ожидающих согласования;
- Количество заявок, согласованных в срок;
- Количество просроченных согласований;
- Количество аннулированных заявок.

Для статистических отчетов (Рисунок 375) возможна только настройка пользователей, выбор справочников, а также периода, за который следует сформировать отчет. Выбор пользователей и справочников осуществляется с помощью кнопки ***Добавить***. Интервал времени выбирается из выпадающего списка.

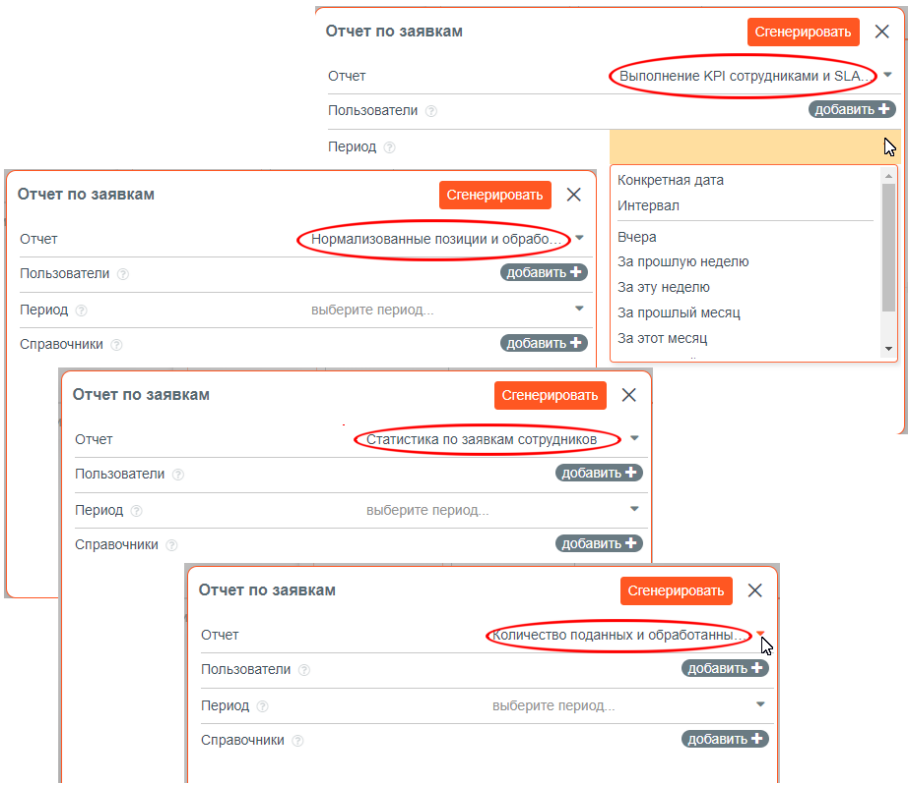


Рисунок 375 - Настройки статистических отчетов

- после выполнения всех необходимых настроек следует нажать кнопку ***Сгенерировать***.

Процесс генерации отчёта будет отображён в фоновых процессах. Сформированный отчёт будет размещён в левой части окна. Щелчком левой кнопки мыши по сформированному отчёту или по ссылке Посмотреть в окне **Фоновые процессы**, можно ознакомиться с содержимым отчёта, которое будет представлено в виде таблицы в правой части рабочего окна (Рисунок 376).

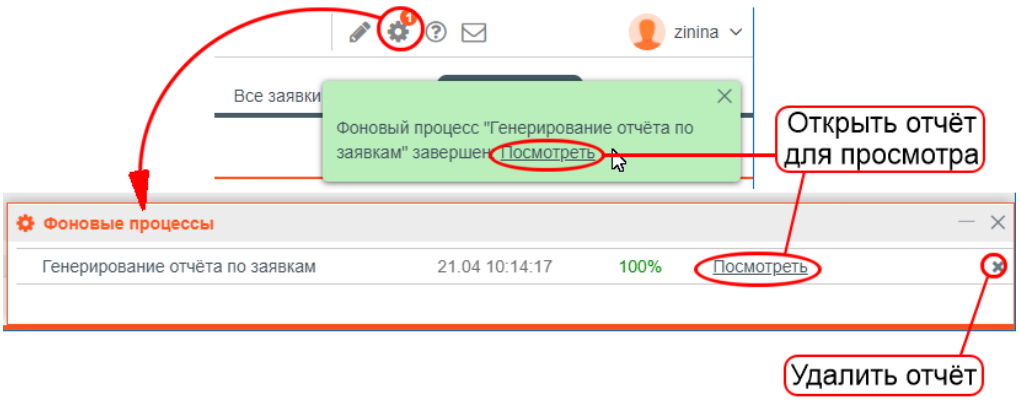


Рисунок 376 - Доступ к сгенерированным файлам отчёта по заявкам

В разделе **Фоновые процессы** можно удалить сгенерированный отчёт, нажав на кнопку ✖ в соответствующей строке.

Для отчетов по пользователям доступен экспорт сформированного файла в Excel. Для этого используется кнопка ***Выгрузить в XLSX***, расположенная над таблицей. Для статистических отчетов кроме Excel доступен экспорт в файл формата PDF. Кнопка ***Выгрузить в PDF*** также располагается над таблицей (Рисунок 377).

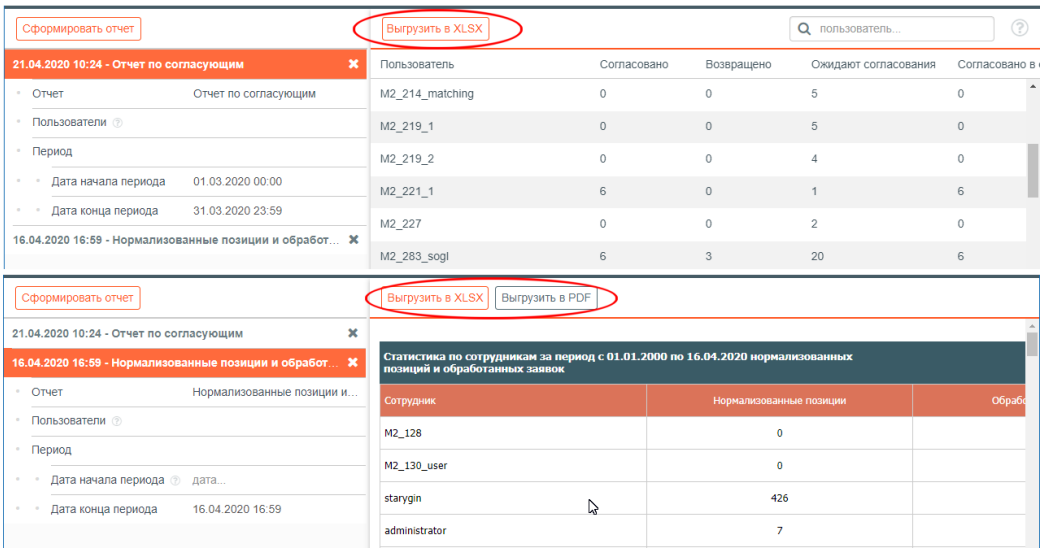


Рисунок 377 - Вид отчетов при просмотре и расположение кнопок экспорта

В таблицах с отчётом по пользователям можно выполнить сортировку записей щелчком левой кнопки мыши по любому из заголовков столбцов. Первый щелчок выполняет сортировку по возрастанию, второй – по убыванию. Также в таблице можно выполнить быстрый поиск по пользователю, если отчёт формировался по всем пользователям.

3.26 Качество данных

Представление (подсистема) **Качество данных** (Рисунок 378) обеспечивает решение следующих задач:

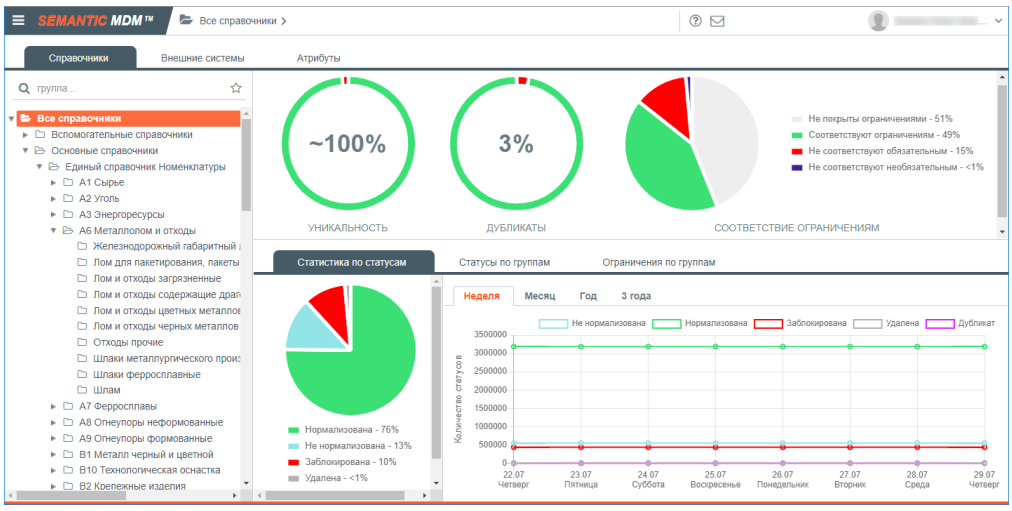


Рисунок 378 - Вид представления **Качество данных** (в разрезе справочников)

- Отслеживание качества данных по позициям, внесенных в *Semantic MDM*;
- Отслеживание прогресса наследования и нормализации позиций в рамках справочника, предприятия, внешней ИС;
- Отслеживание качества данных по атрибутам группы;
- Отслеживание качества данных по каждому из атрибутов индивидуально;
- Выявление позиций, не покрытых правилами проверки качества данных;
- Выявление дублирующих значений атрибутов;
- Выявление атрибутов дубликатов.

Данная подсистема позволяет оперативно просмотреть аналитическую информацию для любой выделенной группы, в следующих разрезах:

- Справочники – предоставляет аналитику по всем позициям текущей группы;
- Внешние системы – предоставляет аналитику по всем позициям, связанным с выбранной внешней системой или внешними системами выбранного предприятия;
- Атрибуты – предоставляет аналитику по атрибутам, продекларированным в выбранной группе;

- Атрибут - предоставляет аналитику по выбранному атрибуту.

Доступ к любому из срезов данных, осуществляется через собственную вкладку. Для выбранных атрибутов, вкладки сохраняются в рамках выбранной группы их наследованием ниже по иерархии.

3.26.1 Справочники

На вкладке **Справочники** (Рисунок 379) для быстрой оценки качества данных доступны следующие быстрые показатели:

- *Уникальность* – позволяет оценить насколько позиции являются уникальными в рамках выделенной группы. Данный показатель доступен только при наличии для группы ограничений на уникальность одного или нескольких атрибутов;
- *Дубликаты* – позволяет по ранее запущенным поискам дубликатов оценить количество позиций дубликатов для текущей группы;

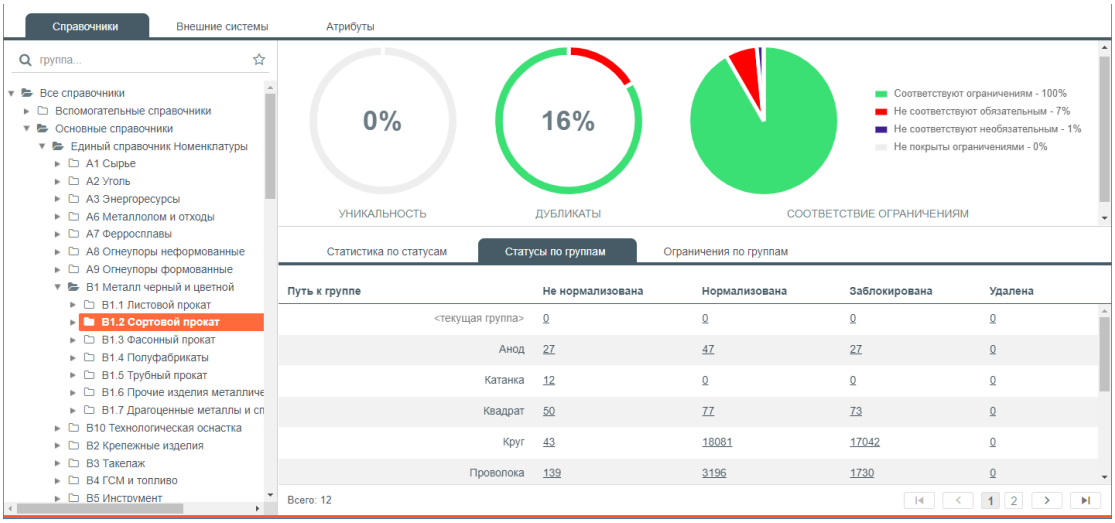


Рисунок 379 - Качество данных в разрезе справочных групп

- *Соответствие ограничениям* – позволяет быстро оценить покрытие позиций ограничениями и их соответствия этим ограничениям. Для более детальной информации, ниже данного показателя доступна вкладка **Ограничения по группам**, где отображается информация о

соответствии позиций ограничениям для каждой из подчиненных групп, в которых есть позиции (Рисунок 380).

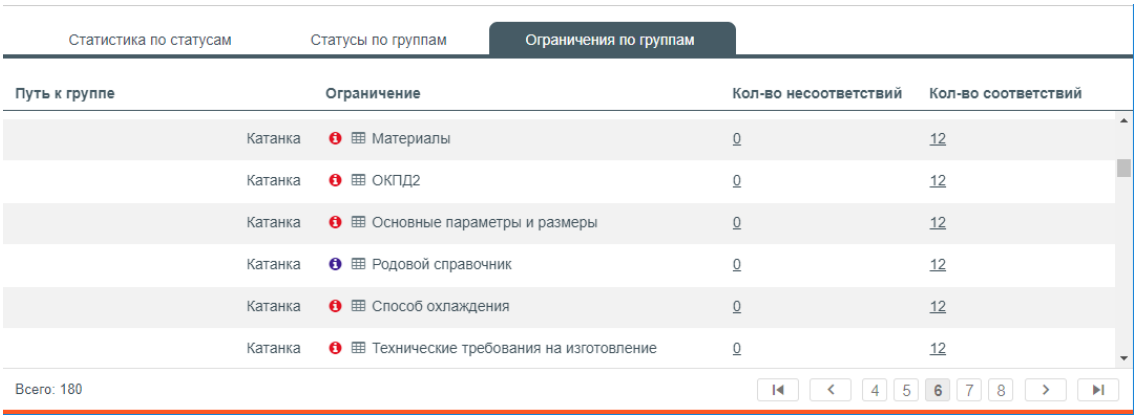


Рисунок 380 - Вкладка **Ограничения по группам** для справочников

Для отслеживания текущих результатов и прогресса наследования и нормализации позиций по справочнику на вкладке **Статистика по статусам** (Рисунок 381) отображается информация по статусам позиций на текущий момент времени для выделенной группы, а также статистическая информация за неделю, месяц, год, 3 года. Для получения более детальной информации по статусам позиций в разрезе дочерних групп, предусмотрена специальная вкладка «Статусы по группам».

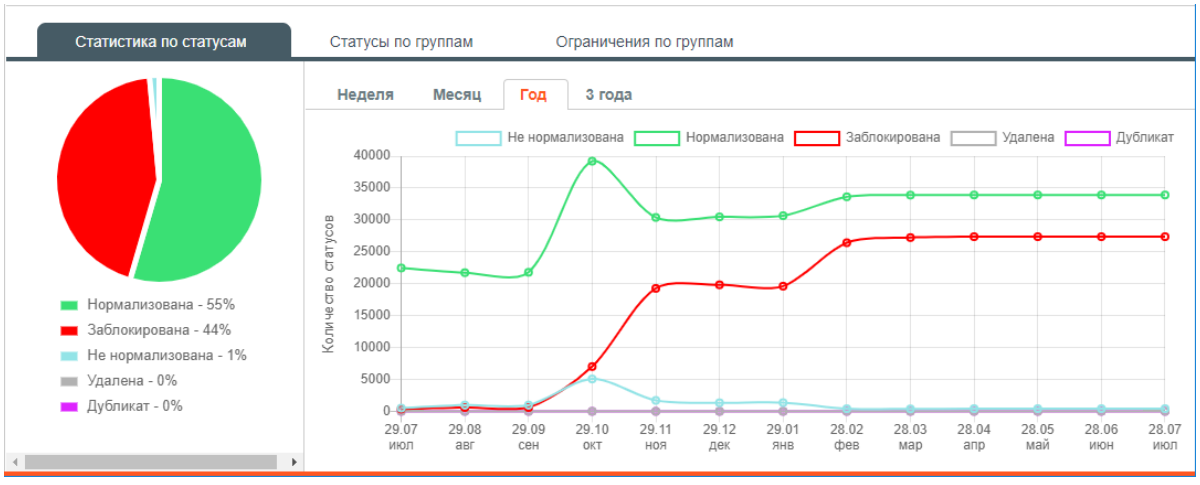


Рисунок 381 - Качество данных справочника со статистикой по статусам

3.26.2 Внешние системы

На вкладке **Внешние системы** (Рисунок 382) для быстрой оценки качества данных доступны следующие быстрые показатели:

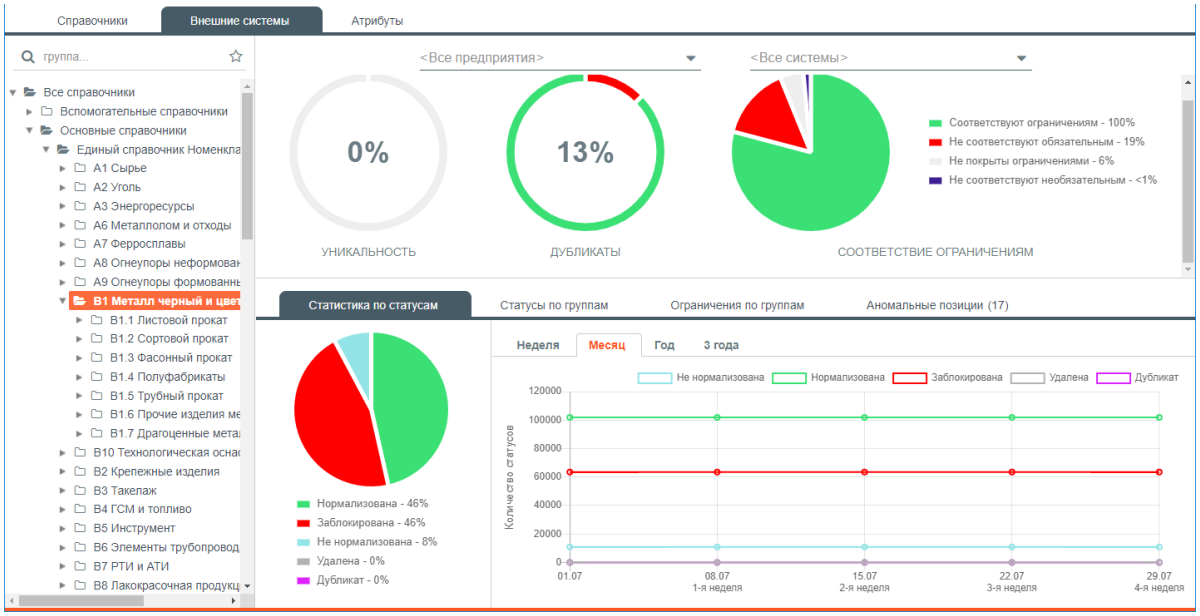


Рисунок 382 - Представление качества данных по внешним системам

- *Уникальность* – позволяет оценить насколько позиции являются уникальными в рамках выделенной группы. Данный показатель доступен только при наличии для группы ограничений на уникальность одного или нескольких атрибутов;
- *Дубликаты* – позволяет по ранее запущенным поискам дубликатов оценить количество позиций дубликатов для текущей группы;
- *Соответствие ограничениям* – позволяет быстро оценить покрытие позиций ограничениями и их соответствия этим ограничениям. Для более детальной информации, ниже данного показателя доступна вкладка **Ограничения по группам**, в которой отображается информация о соответствии позиций ограничениям для каждой из подчиненных групп, в которых есть позиции.

Над быстрыми показателями расположены фильтры, позволяющие выбрать предприятие, для которого проводится анализ, а также конкретную

внешнюю ИС. Фильтры могут быть применены как совместно, так и по отдельности (Рисунок 383).

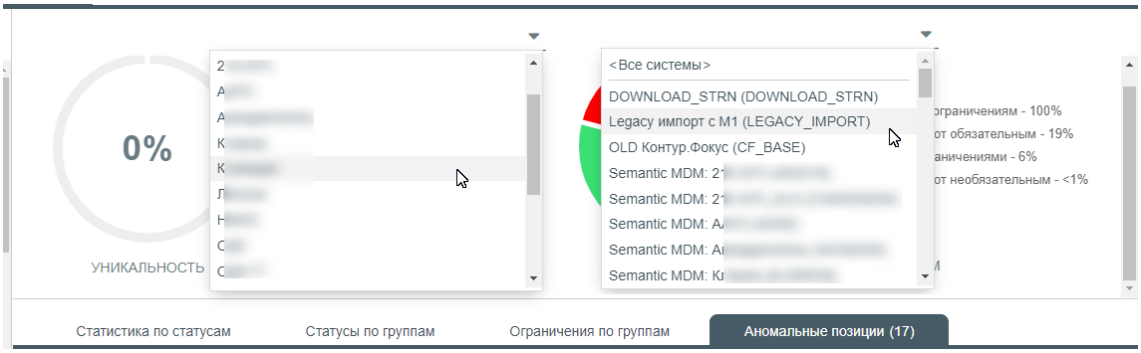


Рисунок 383 - Фильтрация качества данных по предприятию и внешней ИС

Для отслеживания текущих результатов и прогресса наследования и нормализации позиций по внешней системе, на вкладке **Статистика по статусам** отображается информация по статусам позиций на текущий момент времени для выделенной группы, а также статистическая информация за неделю, месяц, год, 3 года. Для получения более детальной информации по статусам позиций в разрезе дочерних групп, предусмотрена специальная вкладка **Статусы по группам**.

Вкладка **Аномальные позиции** (Рисунок 384) содержит сведения о позициях, имеющих более одного переходного ключа с одной внешней ИС.

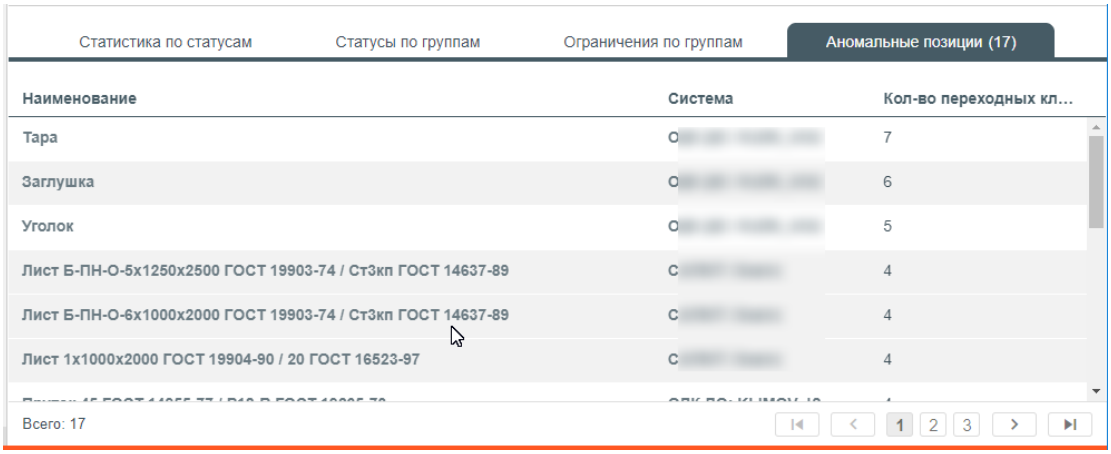


Рисунок 384 - Статистика по аномальным позициям в разрезе внешних систем

3.26.3 Атрибуты

На вкладке **Атрибуты** (Рисунок 385) представления **Качество данных** для быстрой оценки качества данных доступны следующие быстрые показатели:

- *Пустые атрибуты* – позволяет отслеживать пустые атрибуты, продекларированные для текущей группы. По данному показателю можно отфильтровать список атрибутов, отображенных в таблице ниже. Данный показатель должен стремиться к нулю;
- *Не покрываются огр. таблицами* – позволяет отслеживать не покрытые ограничительными таблицами атрибуты и фильтровать список атрибутов в таблице ниже. Данный показатель должен стремиться к общему количество атрибутов продекларированных в текущей группе;
- *Вычисляемые атрибуты* – позволяет контролировать количество вычисляемых атрибутов. Большое количество вычисляемых атрибутов может замедлять редактирование данных, поэтому важно следить, чтобы не появлялись лишние вычисляемые атрибуты. По данному показателю можно отфильтровать список атрибутов, отображенных в таблице ниже;
- *Всего атрибутов* – позволяет увидеть сколько всего продекларировано в текущей группе атрибутов и увидеть их разбивку по типам данных с возможностью фильтрации по ним списка атрибутов, отображенных в таблице ниже. Для фильтрации нужно щелкнуть левой кнопкой мыши по зеленой полоске рядом с типом атрибута.

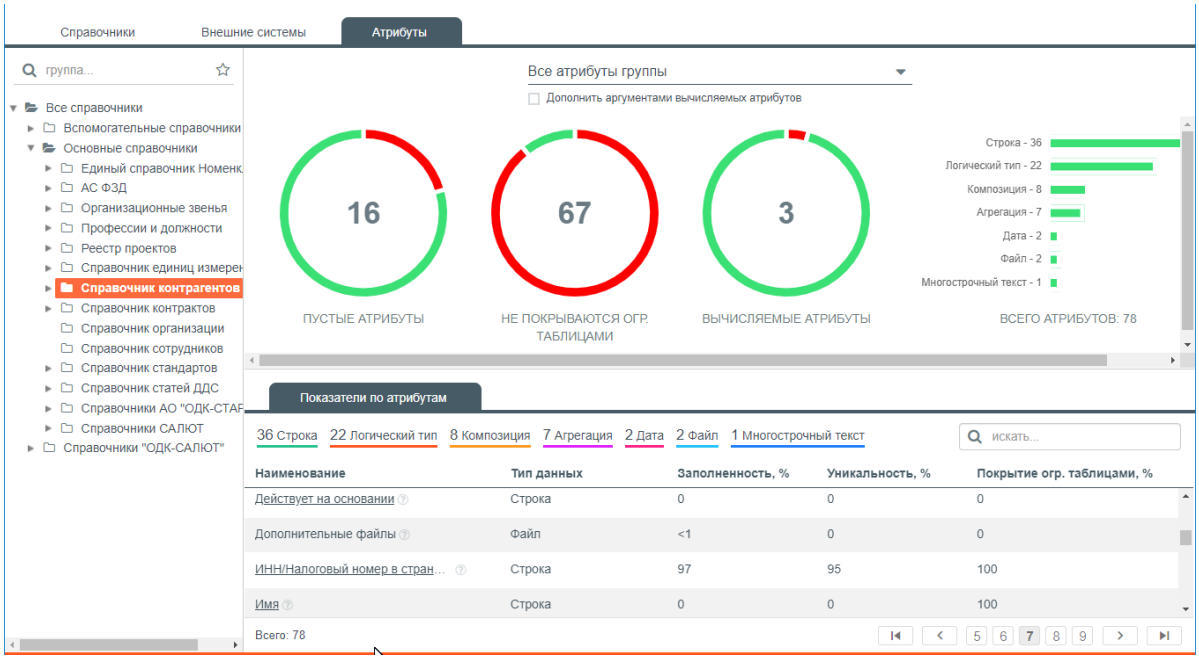
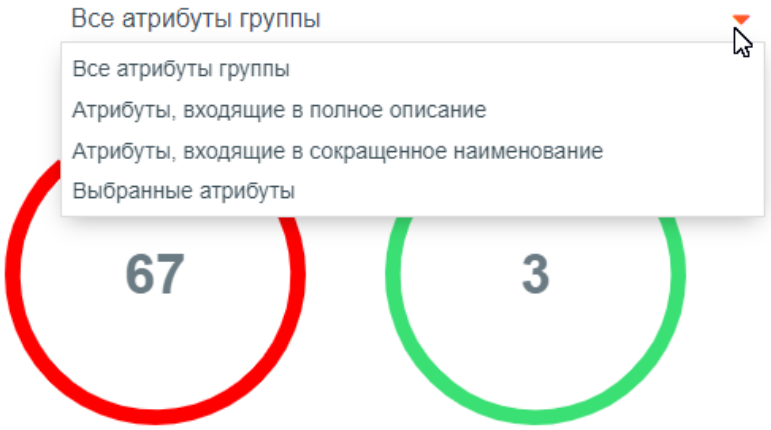


Рисунок 385 - Внешний вид вкладки **Атрибуты** в представлении **Качество данных**

Над быстрыми показателями расположен фильтр позволяющий включить в анализ:

- *Все атрибуты группы* - статистика касается всех атрибутов, продекларированных в выбранной справочной группе;
- *Атрибуты, входящие в полное описание*;



Фильтрация атрибутов

- Атрибуты, входящие в сокращенное наименование (системное наименование) позиции;

- Выбранные атрибуты - после выбора этого фильтра необходимо нажать на кнопку **Выбрать атрибуты** (Рисунок 386) и, в открывшемся окне **Выберите атрибуты**, маркерами отметить интересующие атрибуты и нажать кнопку **Ок**. Количество выбранных атрибутов будет указано на пиктограмме рядом с кнопкой.

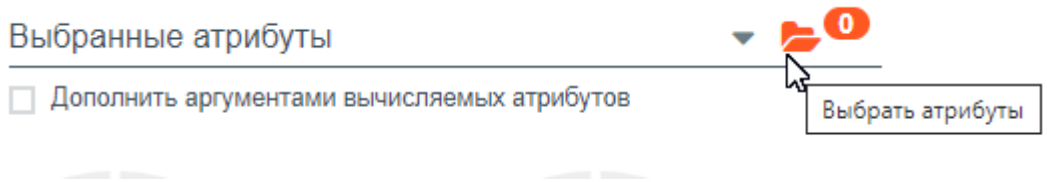


Рисунок 386 - Вызов окна для выбора атрибутов

Опция **Дополнить аргументами вычисляемых атрибутов** обеспечивает включение в анализ атрибутов, на основе которых вычислялось полное наименование позиции.

Для более детального анализа атрибутов, в таблице на вкладке **Показатели по атрибутам** отображен список продекларированных в текущей группе атрибутов с выводом в столбцах следующих показателей:

- *Тип данных* - показывает какой тип данных имеет атрибут. Над таблицей располагается фильтр, по типу данных с указанием названия типа и количества атрибутов ему соответствующего;
- *Заполненность* – позволяет в процентах оценить насколько заполнен атрибут значениями в имеющихся в текущей группе позициях
- *Уникальность* – позволяет в процентах оценить уникальность значений в позициях. Например, уникальность атрибута «Номенклатурный код» будет максимальной, а атрибута «Тип номенклатуры» минимальный
- *Покрытие ограничительными таблицами* – позволяет оценить в процентах покрытие значений атрибута ограничительными таблицами в дочерних группах.

3.26.4 Выбранный атрибут

По нажатию на наименование атрибута в таблице на вкладке **Атрибуты**, будет открыта дополнительная вкладка **<Имя атрибута>** (Рисунок 387), в которой будет отображена детальная информация для анализа данных по этому атрибуту. Для анализа качества данных будут доступны следующие быстрые показатели:

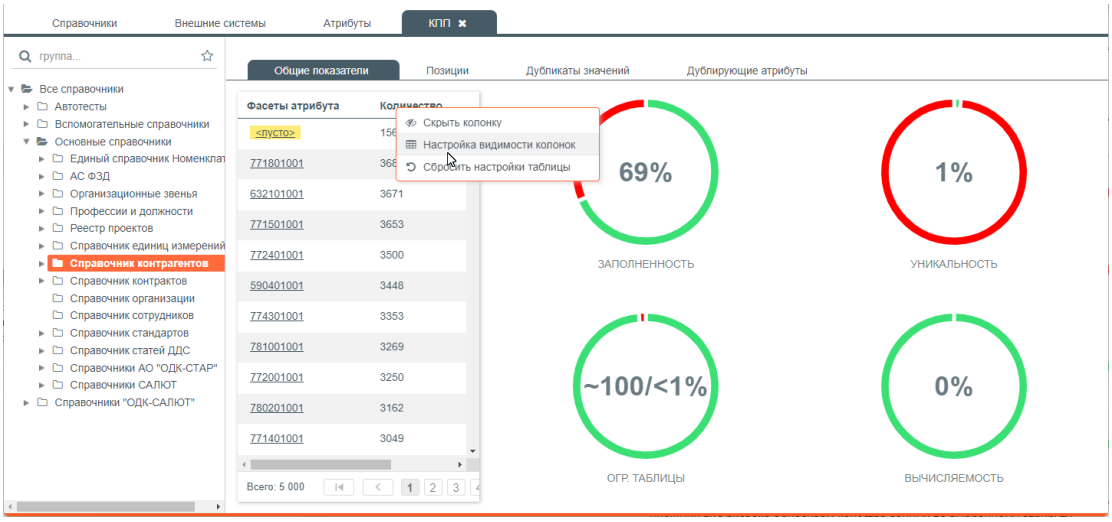


Рисунок 387 - Внешний вид вкладки с анализом качества данных по выбранному атрибуту

- *Заполненность* - позволяет в процентах оценить насколько заполнен атрибут значениями в имеющихся в текущей группе позициях;
- *Уникальность* – позволяет в процентах оценить уникальность значений в позициях. Например, уникальность атрибута «Номенклатурный код» будет максимальной, а атрибута «Тип номенклатуры» минимальный;
- *Огр. таблицы* – позволяет оценить процентах покрытия значений атрибута ограничительными таблицами в дочерних группах;
- *Вычисляемость* – позволяет определить процент значений, которые были вычислены для данного атрибута через формулу, поскольку в некоторых дочерних группах значение может быть вычисляемым, а в некоторых нет.

Кроме быстрых показателей, для выбранного атрибута будут отображены фасеты со значениями этого атрибута. В таблице с фасетами значений можно настроить отображение колонки с количеством значений. Наличие колонок со значениями атрибута и количеством каждого из значений в данных, позволяет выполнять сортировку по любой из колонок. Для анализа качества атрибута в различных разрезах используются дополнительные вкладки:

- **Позиции** – на данной вкладке можно посмотреть какие позиции содержат выбранный значение (фасет) (Рисунок 388);

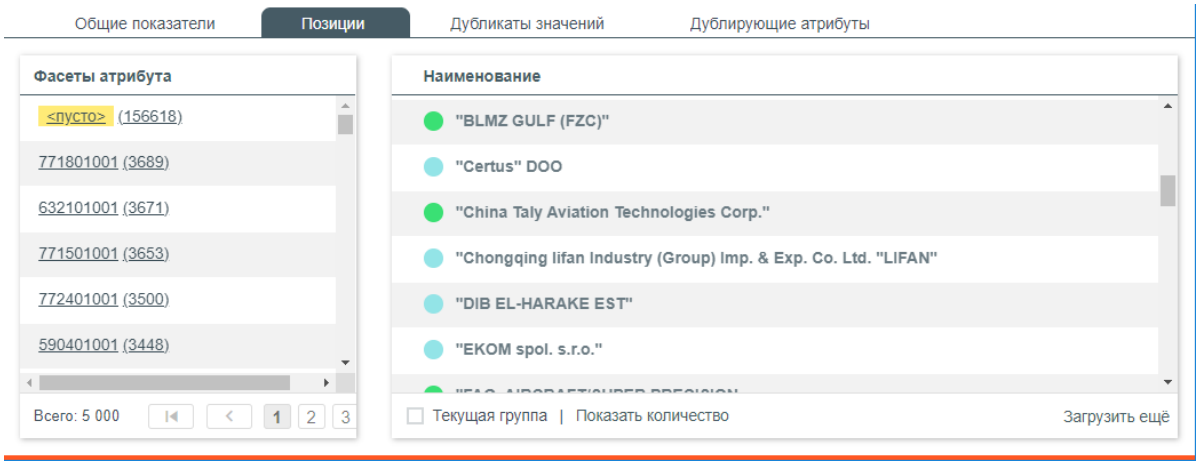


Рисунок 388 - Позиции, содержащие анализируемый атрибут

- **Дубликаты значений** – на данной вкладке можно увидеть какие для атрибута есть потенциальные дубликаты значений, которые различаются различным написанием схожих букв латиницы и кириллицы, или имеют двойные или концевые пробелы;
- **Дублирующие атрибуты** – позволяет отобразить атрибуты, с которыми есть пересечение по значениям атрибутов с другими атрибутами этого же типа данных. Поиск дубликатов выполняется не только по сходству значений в позициях, но и сходству значений в ограничительных таблицах. Дубликаты агрегаций определяются как по сходству ссылок, так и по сходству наименований агрегируемых позиций;

- **Дублирующие атрибуты (ML)**[20] – позволяет определить с помощью подсистемы машинного обучения дубликаты атрибутов со схожими значениями (Рисунок 389);

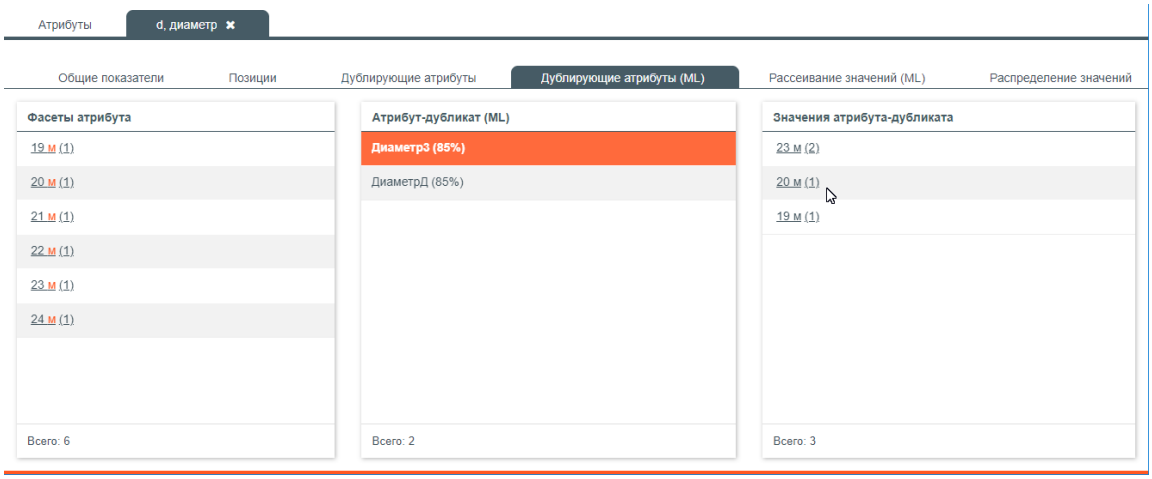


Рисунок 389 - Определение атрибутов-дубликатов при помощи машинного интеллекта

- **Рассеивание значений (ML)** – позволяет увидеть разброс значений в двухмерном пространстве и выделить аномальные значения. Данная вкладка (Рисунок 390) помогает выявить аномальные значения атрибута и группы скопления значений, которые могут семантически иметь один смысл, но записан в значения атрибутов по разному. Например: «Холоднотянутый» как «ХТ», «Холоднокатанный» как «ХК»;

[20] *Вкладки **Дублирующие атрибуты (ML)** и **Рассеивание значений (ML)** доступны только при использовании подсистемы машинного обучения (Machine Learning, сокращенно ML). Подсистема машинного обучения является дополнительной опцией и в состав базовой MDM не входит.*



Рисунок 390 - Рассеивание значений, определенное для атрибута машинным интеллектом

- **Распределение значений** – позволяет для числовых атрибутов на графике увидеть распределение числовых значений и выявить аномальные значения атрибутов (Рисунок 391).

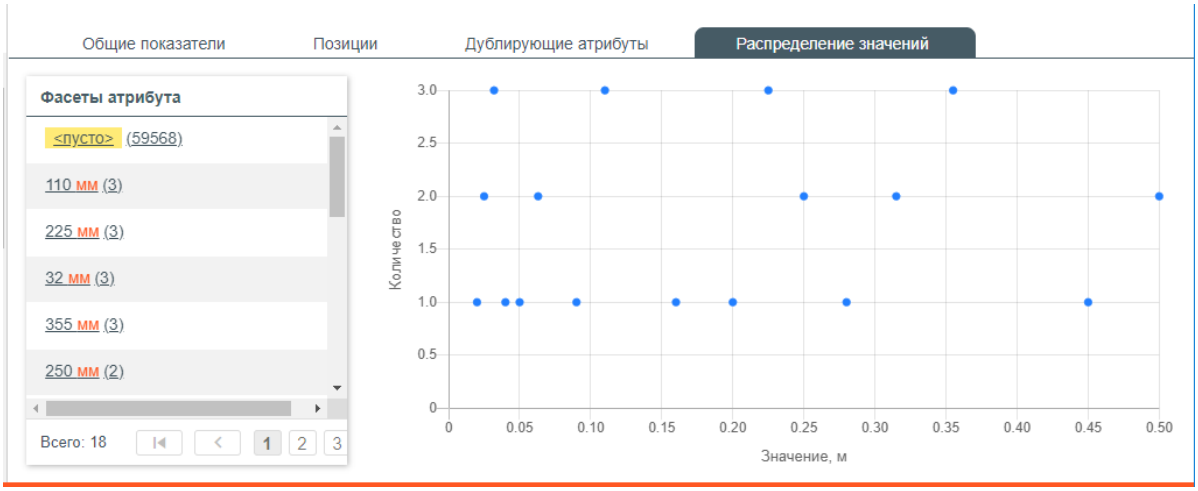


Рисунок 391 - Распределение значений атрибута

Открытая вкладка с выбранным атрибутом запоминается для текущей группы и при выделении этой или дочерней группы, вкладка с данным атрибутом автоматически будет появляться в панели вкладок.

3.27 Машинное обучение

Сервис машинного обучения (Machine Learning - ML) является дополнительной опцией *Semantic MDM* и предназначен для автоматизации работы экспертов НСИ. ML-сервис необходим для работы мастера создания позиций, автоматической обработки заявок, нормализации больших объемов данных, оценки качества данных в *Системе*. Система машинного обучения основана на нейронных сетях и для её развития и повышения эффективности её работы необходимо проводить постоянное обучение на нормализованных данных. Источниками нормализованных данных (мастер-данных) может выступать сам *Semantic MDM* или внешние ИС, качеству данных в которых пользователь доверяет. Чем больше объем данных, проходящих через ML-сервис, тем выше качество работы всех подсистем, связанных с ним.

Подсистема управления сервисом машинного обучения (подсистема **Машинное обучение**) является встроенным подключаемым расширением (см. п. 3.31.4) *Системы,* предназначена для организации, контроля процесса и оценки результатов обучения ML-сервиса и позволяет решать следующие задачи:

- Выполнять настройку режима группировки данных по каждому из атрибутов;
- Регистрировать внешние источники данных, откуда выполняется загрузка данных для обучения;
- Выполнять загрузку данных из *.csv файлов;
- Осуществлять просмотр текущего состояния пакетов обучения;
- Отслеживать процесс обучения пакетов;
- Выполнять запуск обучения выбранных или всех пакетов вручную;
- Выполнять остановку процесса обучения вручную;
- Настраивать расписания запуска автоматического переобучения;
- Оценивать точность обучения.

ML-сервис обучается на загруженных по каждому из атрибутов данных, сгруппированных в пакеты обучения. Группировка данных в пакет обучения,

осуществляется в соответствии с выбранным режимом группировки для атрибута:

- **Для всех групп** – все данные, загруженные для атрибута, группируются в один пакет и будут предсказываться всегда, для любого введенного наименования;
- **Автоматически по группам** – все данные, загруженные для атрибута, группируются автоматически в пакеты обучения по одной или нескольким справочным группам. Если уникальные значения атрибута для двух разных справочников *Semantic MDM* пересекаются более чем 60%, то данные для этих двух групп объединяются в один пакет обучения и так далее, пока не будут проанализированы все данные атрибута по группам Semantic MDM.

Кроме данных, загруженных из *.csv файлов, источником данных для машинного обучения может выступать сам *Semantic MDM*. Загрузка данных в сервис машинного обучения ML выполняется через специально зарегистрированную внешнюю систему и обменный класс в подсистеме импорта и репликации. Загрузка данных для обучения из Semantic MDM в сервис машинного обучения ML позволяет создать самообучающуюся систему, на основе нормализованных данных из справочников *Системы* и данных из заявок пользователей.

В подсистеме **Машинное обучение** пользователь имеет возможность работать с пятью вкладками:

- **Мониторинг и управление** - выполняется настройка расписания обучения, настройка пакетов, а также запускается, останавливается и контролируется процесс переобучения;
- **Данные для обучения** - на вкладке выполняется настройка подключений к источникам данных, а также их загрузка / выгрузка из файлов CSV;
- **Данные для проверки** – на вкладке отображены данные, на основе которых выполняется проверка;

- **Общая точность предсказания** – вкладка содержит график с консолидированной информацией о точности предсказания по проверяемым наименованиям с возможностью сравнения текущей точности с точностью на выбранную дату;
- **Точность предсказания по группам** – вкладка содержит информацию о точности предсказания по проверяемым наименованиям для каждой из имеющихся групп.

3.27.1 Мониторинг и управление

На вкладке **Мониторинг и управление** пользователю доступны следующие действия:

- просмотр текущего состояния по имеющимся пакетам обучения;
- отслеживание процесса обучения;
- ручной запуск обучения выбранных пакетов и запуск полного переобучения;
- ручная остановка процесса обучения;
- настройка расписания автоматического запуска переобучения;
- настройка объединения данных по атрибуту.

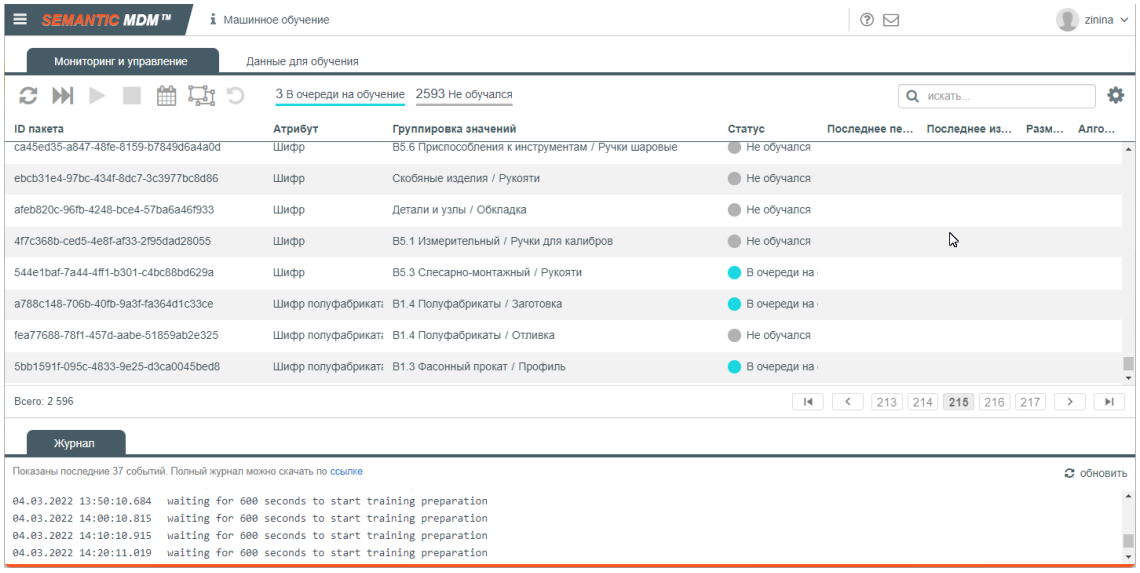


Рисунок 392 - Вкладка **Мониторинг и управление** представления Машинное обучение

На вкладке присутствует таблица с пакетами обучения, панель инструментов для работы с пакетами обучения и журнал процесса (расположен в нижней части окна). Состав колонок таблицы является настраиваемым и содержит следующие данные:

- **ID пакета** - идентификатор пакета обучения;
- **Атрибут** - атрибут *Semantic MDM,* для которого сформирован пакет обучения. Актуальное наименование атрибута определяется по его идентификатору (GUID) и отображается в виде ссылки на этот атрибут в **Классификаторе атрибутов**;
- **Группировка значений** - список онтологических групп, для которых значения атрибута были объединены в один пакет. Наименование группы формируется по шаблону: <наименования текущей группы>+< \ >+ <наименование родительской группы> и отображается в виде ссылки на группу с возможностью перехода на нее. Для «сквозного» атрибута в колонку выводится системное значение – *Для всех групп*;
- **Статус** - текущий статус пакета обучения:
  - **Не обучался** - пакет ни разу не обучался. Данный статус устанавливается после создания нового пакета;
  - **В очереди на обучение** - пакет запланирован к обучению и будет скоро переобучен. Данный статус устанавливается пользователем или автоматически, после наступления запланированного переобучения;
  - **Требует переобучения** - пакет ранее был обучен, но по нему поступили новые данные или был изменен состав проверочных данных, поэтому его надо переобучить. Данный статус устанавливается автоматически после изменения исходных или проверочных данных, если текущий статус соответствует **Обучение завершено**. Если текущий статус соответствует **Обучается**, то по завершению обучения *Система* проверит наличие даты в поле “Последнее изменение данных” и если дата

там не пустая, то будет установлен статус **Требует переобучения**, в противном случае **Обучение завершено**;

- **Обучение отложено** - обучение пакета было отложено пользователем. Данный статус устанавливается пользователем вручную;
- **Обучается** - пакет в данный момент находится в процессе переобучения;
- **Ошибка обучения** - пакет обучался, но во время обучения произошла ошибка и он не смог доучиться до конца;
- **Обучение завершено** - пакет переобучен;

- **Последнее переобучение** - дата и время последнего переобучения. Если пакет в данный момент обучается, то в этом поле отображается прогрессбар текущего процесса переобучения. Если пакет ни разу не обучался, то поле остается пустым. После завершения переобучения *Система* запишет в данное поле дату и время завершения процесса.
- **Последнее изменение данных** - дата и время последнего изменения обучаемых или проверочных данных для пакета. Эта дата и время автоматически записываются после изменения обучаемых или проверочных данных для пакета. Перед запуском процесса переобучения пакета, данная дата сбрасывается;
- **Размер пакета** - одно из значений:
  - **Большой** - больше миллиона наименований на один атрибут;
  - **Средний** - больше 100 тысяч наименований на один атрибут;
  - **Маленький** - меньше 10 тысяч наименований на один атрибут;
- **Алгоритм** - наименование алгоритма, на основании которого был обучен пакет. Алгоритм выбирается автоматически на основе размера пакета.

По умолчанию все пакеты в таблице отсортированы по наименованию атрибута. Сортировка доступна по всем полям, за исключением столбца **Группировка значений**.

Фасетная фильтрация (Рисунок 393) доступна по следующим полям: **Атрибут**, **Группировка значений**, **Статус**, **Алгоритм**. Фасетная фильтрация по столбцу **Группировка значений** выполняется через специальный диалог с «худым» деревом, в котором можно выбрать как конечные, так и родительские группы. При выборе родительской группы, в результат попадут все пакеты, в которых есть хотя бы одна конечная группа, входящая в выбранную родительскую. В фасетной фильтрации по группам присутствует системное значение «Все группы», с помощью которого можно оставить в списке пакеты, которые обучаются сразу на данных для всех групп. Для быстрой фильтрации по статусам, фасеты по статусам также отображаются над таблицей сверху.

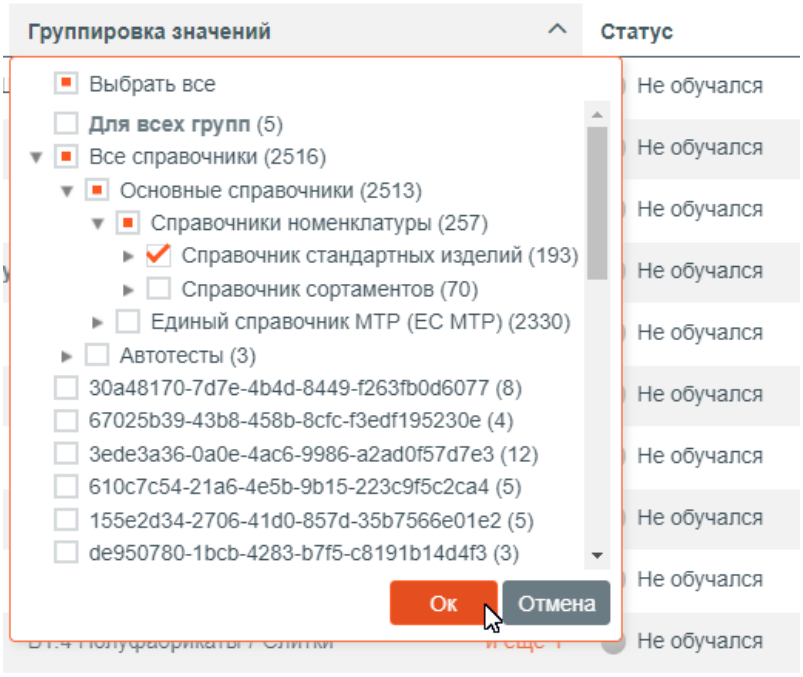


Рисунок 393 - Фасетная фильтрация с «худым» деревом

Для выполнения поиска по таблице, над таблицей справа имеется поле быстрого поиска, с помощью которого можно выполнять поиск по идентификатору пакета, идентификатору атрибута и наименованию атрибута.

Для управления процессом обучения используются кнопки и команды контекстного меню, представленные в таблице Таблица 26:

Таблица 26. Кнопки для управления вкладкой «Мониторинг и управление»

Кнопка / Команда		Функция
	Обновить таблицу	Команда для обновления таблицы
	Переобучить все	Команда открывает окно подтверждения, в котором пользователь должен выбрать режим переобучения – **Переобучить все пакеты** - будут переобучены все пакеты, кроме пакетов со статусом **Обучение отложено**. Если установлен маркер **Только неактуальные**, то для данного режима запускается переобучение всех пакетов со статусами: **Не обучался**, **Требует переобучения**, **Ошибка обучения**
	Переобучить	Команда ставит выбранные в таблице пакеты в очередь на переобучение независимо от их текущего статуса
	Отменить обучение	Команда применяется только к выбранным в таблице пакетам со статусами: **Обучается**, **В очереди на обучение**
	Настроить расписание переобучения	Открывает окно настройки расписания запуска автоматического переобучения, которое идентично команде **Переобучить все** с режимом “Переобучить только неактуальные пакеты”. Расписание настраивается и записывается в формате CRON.
	Настройка группировки значений атрибутов	Открывает окно с таблицей атрибутов для настройки режима группировки атрибутов. Для выбора доступны два режима: *Автоматически по группам* и *Для всех групп*.
	Отложить / вернуть обучение	Команда позволяет установить статус **Отложить обучение**, для выбранного пакета со статусом: **Не обучался**, **Требует переобучения**, **Ошибка обучения**. Команда позволяет вернуть процесс обучения для выбранного пакета со статусом **Обучение отложено**, при этом для пакета будет восстановлен его исходный статус

Каждый пакет обучения в таблице содержит либо значения одного атрибута, сгруппированные сразу для всех групп, в этом случае атрибут считается “сквозным”, либо значения сгруппированные автоматически по нескольким онтологическим группам справочника, в этом случае атрибут считается «простым». Для «простого» атрибута может быть сформировано несколько пакетов обучения. Формирование пакетов обучения для “простого” атрибута выполняется автоматически, во время загрузки новых значений для данного атрибута. Объединение значений онтологических групп в один пакет обучения происходит при вхождении как минимум половины значений минимальной онтологической группы, в другую более крупную онтологическую группу.

В таблице ниже описаны процессы изменения статусов для пакетов обучения.

Таблица 27. Статусная модель пакетов обучения

Исходный статус	Новый статус	Событие, влекущее смену статуса
-	**Не обучался**	Создается *Системой* автоматически для нового пакета обучения, после загрузки данных для обучения
Не обучался	**В очереди на обучение**	• Создается *Системой* при автоматическом переобучении по расписанию; • Запускается пользователем при постановке в очередь на обучение кнопками ***Переобучить***, ***Переобучить все***
	Обучение отложено	Устанавливается пользователем при помощи кнопки ***Отложить / вернуть обучение*** для предотвращения автоматического переобучения пакета
В очереди на обучение	**Обучается**	Запускается автоматически при освобождении ресурсов для обучения следующего пакета обучения
	Отменен	Устанавливается пользователем по нажатию на кнопку ***Отменить обучение***
Обучается	**Требует переобучения**	Устанавливается автоматически, если изменились данные для обучения или проверки, во время обучения. То есть когда после успешного завершения обучения, дата и время записанные в поле **Последнее изменение данных** оказались не пустыми.
	Ошибка обучения	Устанавливается автоматически, если произошла ошибка во время обучения
	Обучение завершено	Устанавливается автоматически, если обучение успешно завершилось
Обучение завершено	**Требует переобучения**	Устанавливается автоматически, если изменились данные, на которых обучается или по которым проверяется пакет обучения
Обучение отложено	**Требует переобучения**	Устанавливается пользователем по нажатию на кнопку ***Отложить / вернуть обучение***. *Система* возвращает исходный статус
	Не обучался	
Отменено	**В очереди на обучение**	Запускается пользователем постановка в очередь на обучение через кнопками ***Переобучить***, ***Переобучить все***
Требует переобучения	**Обучение отложено**	Устанавливается пользователем по нажатию на кнопку ***Отложить / вернуть обучение***
	В очереди на обучение	• Создается *Системой* при автоматическом переобучении по расписанию; • Запускается пользователем при постановке в очередь на обучение кнопками ***Переобучить***, ***Переобучить все***

Под таблицей располагается панель с журналом переобучения по пакетам и журнала основного процесса, управляющего обучением. Если в таблице выделен пакет, то будут показаны последние 1500 строк из журнала обучения данного пакета. Если в таблице не выделено ни одного пакета, то в журнале будут отображены последние 1500 строк из журнала работы процесса, управляющего обучением. По ссылке, расположенной рядом с именем заголовка вкладки **Журнал**, будет автоматически запущен процесс скачивания выбранного журнала на ПК пользователя. Журнал сохраняется в файле формата TXT.

3.27.1.1 Настройка группировки значений атрибутов

Чтобы выполнить группирование значений атрибута в пакет для проведения обучения необходимо:

- нажать кнопку ***Настройка группировки значений атрибутов*** или вызвать аналогичную команду из контекстного меню;
- в окне **Настройка группировки значений атрибутов (**Рисунок 394**)** с помощью сортировки, просмотра списка или ячейки быстрого поиска найти интересующий пакет обучения;
- в ячейке Режим группировки нужного пакета обучения выбрать из выпадающего списка один из режимов:
  - **Автоматически по группам** - значения атрибута будут автоматически сгруппированы по нескольким онтологическим группам справочника. Такой атрибут считается «простым». Для «простого» атрибута может быть сформировано несколько пакетов обучения. Формирование пакетов обучения для «простого» атрибута выполняется автоматически, во время загрузки новых значений для данного атрибута. Объединение значений атрибута в один пакет обучения происходит при вхождении как минимум половины значений атрибута

минимальной онтологической группы, в другую более крупную онтологическую группу;

- **Для всех групп** - содержит значения одного атрибута, сгруппированные сразу для всех групп. Такой атрибут считается «сквозным»;

- нажать кнопку ***Ок***. Новые режимы группировки атрибута будут переданы в подсистему ML. Для отказа от выполненных настроек достаточно закрыть окно **Настройка группировки значений атрибутов** без подтверждения выполненных действий.

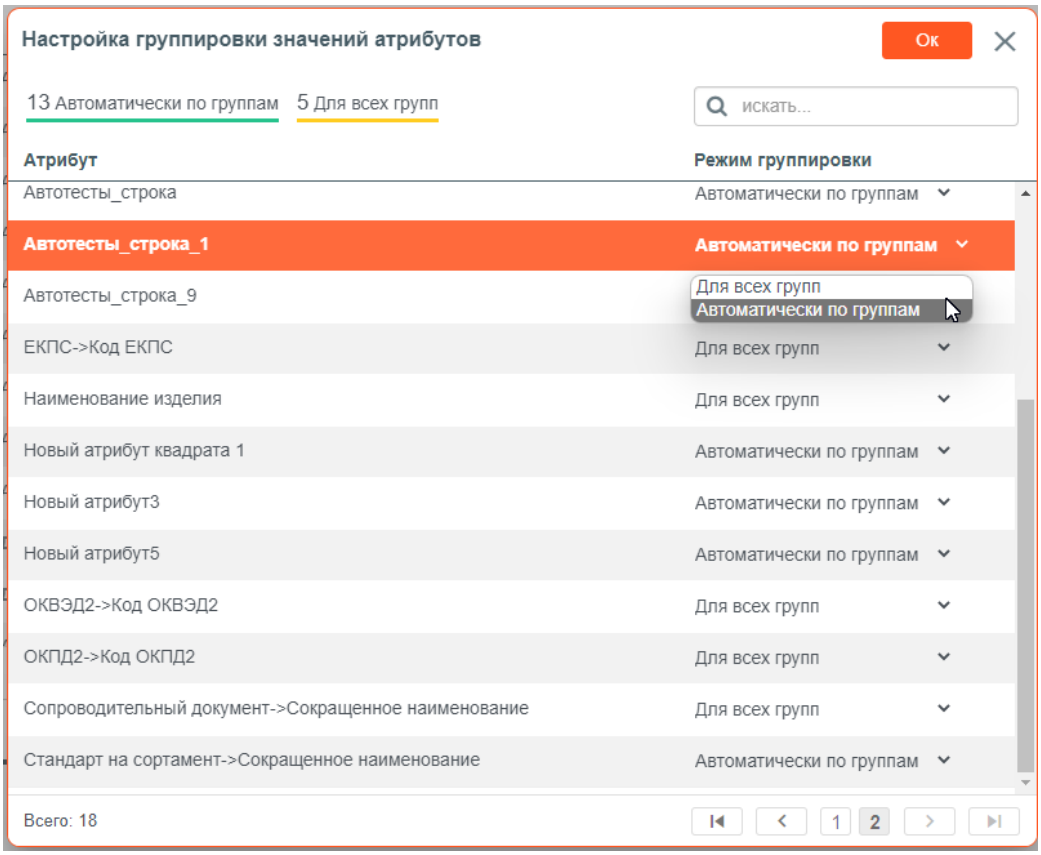


Рисунок 394 - Выполнение настройки режима группирования атрибутов

3.27.1.2 Настройка расписания переобучения

Для того, чтобы настроить расписания обучения / переобучения необходимо:

- нажать кнопку Настроить расписание переобучения;

- в открывшемся окне Расписание переобучения (Рисунок 395) выбрать временной интервал переобучения. Для выбора доступны следующие периоды:
  - **ежеминутно** - в выпадающем списке в ячейке **Каждую** выбрать интервал времени запуска переобучения от 1 до 59 минут;
  - **ежечасно** - в выпадающем списке в ячейке **Каждый** выбрать интервал времени запуска переобучения от 1 до 23 часов и от 0 до 59 минут;
  - **ежедневно**:
    - в выпадающем списке в ячейке **Каждый** выбрать интервал времени запуска переобучения от 1 до 31 дня, от 0 до 23 часов и от 0 до 59 минут;
    - в варианте **Каждый рабочий день** выбрать интервал времени запуска переобучения от 0 до 23 часов и от 0 до 59 минут;
  - **еженедельно** - установить маркеры в ячейках напротив дней недели, отводящихся на выполнение переобучения - с понедельника по воскресенье включительно, а также время запуска переобучения с 0 до 23 часов и от 0 до 59 минут;
  - **ежемесячно:**
    - с 1 по 31 день каждого 1 - 12 месяца в 0 - 23 часа и 0 - 59 минут;
    - в первую - пятую или последнюю неделю, в понедельник - воскресенье каждого 1 - 12 месяца в 0 - 23 часа и 0 - 59 минут;
  - **ежегодно:**
    - в 1 - 31 день указанного 1 - 12 месяца в 0 - 23 часа и 0 - 59 минут;
    - в первую - пятую или последнюю неделю, в понедельник - воскресенье указанного 1 - 12 месяца в 0 - 23 часа и 0 - 59 минут;
  - **выражение** - установка времени в формате описания CRON;
- нажать кнопку Ок. Новое расписание переобучения будут передано в подсистему ML. Для отказа от выполненных настроек достаточно

закрыть окно **Расписание переобучения** без подтверждения выполненных действий.

Настройки отображаются в ячейках **Текущее** и **Новое**, расположенных в центре окна настройки.

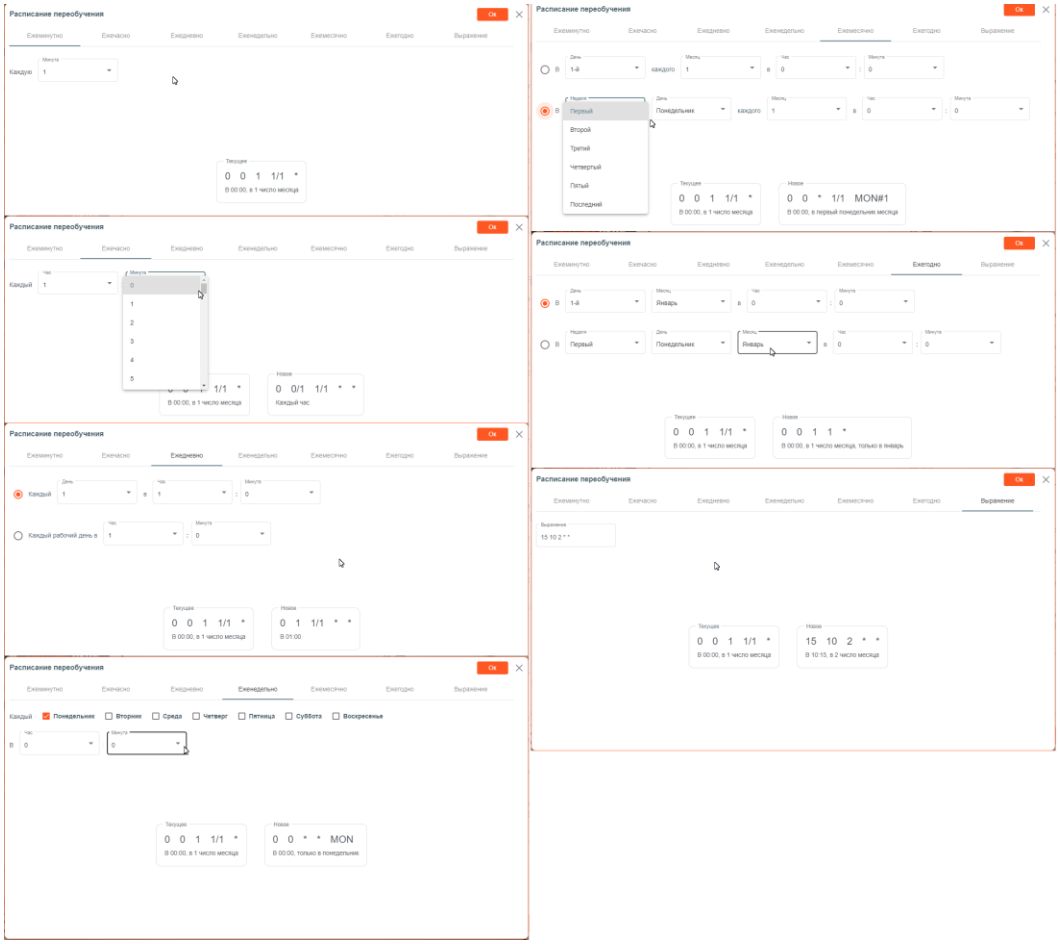


Рисунок 395 - Варианты настройки расписания переобучения

3.27.1.3 Управление процессом обучения

Процесс обучения выполняется по расписанию, если оно настроено. В случае, если расписание настроено не было или есть необходимость срочного переобучения пакетов, в ML-сервисе предусмотрены следующие команды для ручного запуска:

- **Переобучить все** - по данной команде поднимается окно **Подтверждение**, в котором пользователь должен выбрать режим переобучения:

- **Переобучить все пакеты** - за исключением пакетов со статусом “Обучение отложено”;
- **Только неактуальные** - установленный маркер обеспечивает запуск переобучения всех пакетов со статусами: **Не обучался**, **Требует переобучения**, **Ошибка обучения**.

После выбора режима обучения, пользователь должен либо подтвердить переобучение с выбранным режимом с помощью кнопки ***Ок***, либо отказаться от переобучения с помощью кнопки ***Отмена*** (Рисунок 396). После этого все выбранные пакеты встанут в очередь на переобучение, а при наличии свободных ресурсов обучение начнется незамедлительно.

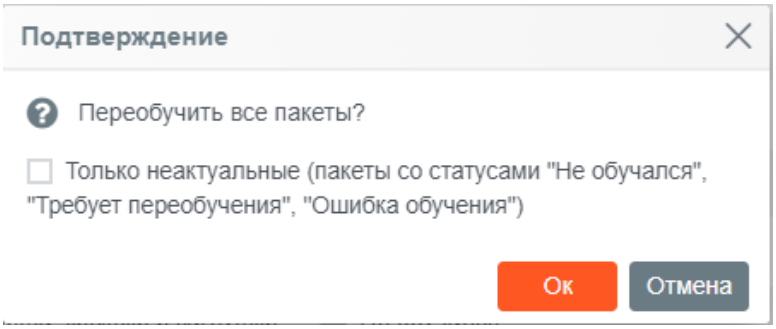


Рисунок 396 - Запуск переобучения

- Отложить обучение / Вернуть обучение - данная команда устанавливает статус Отложить обучение для выбранного пакета с одним из статусов: Не обучался, Требует переобучения, Ошибка обучения.

Если выбран пакет со статусом **Обучение отложено**, то команда получит название **Вернуть обучение**, при этом для пакета будет восстановлен его исходный статус.

3.27.1.4 Данные для обучения

На вкладке **Данные для обучения** (Рисунок 397) пользователю доступны следующие действия:

- Просмотр данных, на основе которых выполняется обучение;

- Быстрый поиск в данных;
- Просмотр альтернативных наименований;
- Просмотр значений атрибутов, которым обучается подсистема обучения;
- Загрузка и выгрузка данных для обучения в csv;

  Загрузка альтернативных наименований из csv.

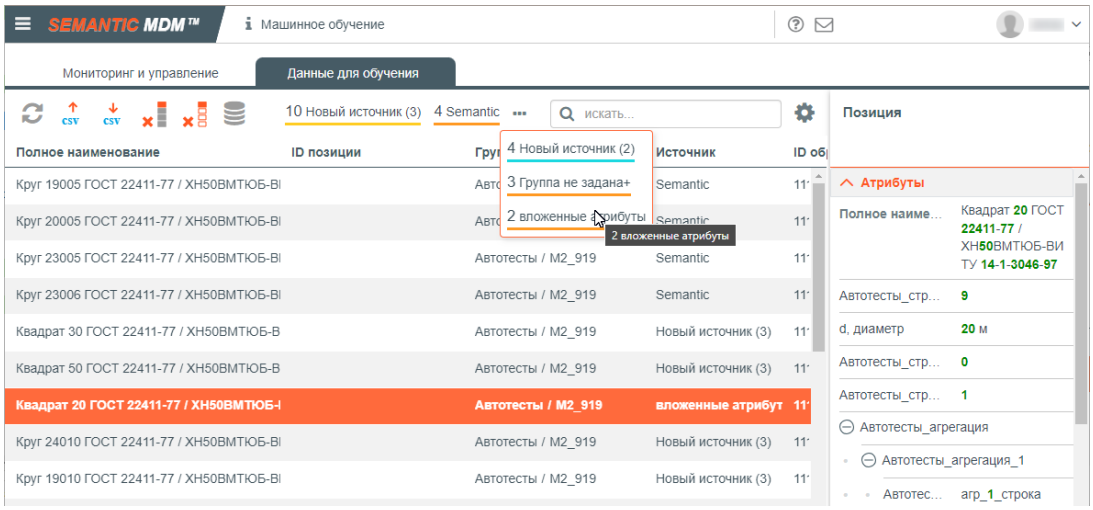


Рисунок 397 - вкладка **Данные для обучения** в представлении Машинное обучение

На вкладке присутствует таблица с загруженными справочными позициями, панель инструментов для загрузки / выгрузки данных и карточка с описанием позиции, входящей в состав загруженных данных. Состав колонок таблицы является настраиваемым (см. п. 3.5.2):

- **Полное наименование** - уникальное полное наименование позиции;
- **ID позиции** - GUID позиции в Semantic MDM. Идентификатор может отсутствовать, если позиция была загружены не из Semantic MDM;
- **Группа** - короткий путь к справочной группе, состоящий из наименования конечной группы и наименования родительской группы. По щелчку правой кнопкой мыши на группе она будет открыта в новой вкладке браузера;
- **Источник** - наименование источника, из которого было загружена позиция со значениями атрибутов;

- **ID образца** - внутренний идентификатор позиции, автоматически присваиваемый ML-сервисом.

В таблице доступны:

- сортировка по колонкам: **Группа**, **Полное наименование** и **Источник**. По умолчанию включена сортировка по колонке **Группа** (выполняется по наименованию короткого пути);
- фасетная фильтрация доступна по колонкам **Группа** и **Источник**. При раскрытии диалога с фасетами отображается «худое» дерево групп, с возможностью выбора конечных или промежуточных групп в иерархии. Для быстрой фильтрации, фасеты по группам отображаются также над таблицей сверху;
- быстрый поиск по таблице выполняется по следующим данным: *GUID, Полное наименование, значение любого атрибута, Альтернативное наименование*.

Для работы с данными используются кнопки, представленные в таблице ниже:

Таблица 28. Кнопки на вкладке «Данные для обучения»

Кнопка		Функция
	**Обновить таблицу**	
	**Загрузить из csv**	Загружает позиции из файла csv. Файл должен содержать четыре колонки со следующими данными: первая колонка (positionName) должна содержать наименование позиции; вторая (propertyName) - наименование предсказываемого атрибута, понятное для пользователя; третья (propertyId) - GUID атрибута в Semantic MDM; четвертая (initialValue) - предсказываемое значение
	**Экспорт в csv**	Выгружает все позиции со значениями атрибутов в csv, с учетом наложенных фильтров
	**Удалить**	Удаляет выделенные позиции из базы данных машинного обучения. Удаление будет выполнено только после подтверждения операции пользователем.
	**Удалить все**	Удаляет все позиции из базы данных машинного обучения, которые есть в таблице с учетом наложенных фильтров. Удаление будет выполнено только после подтверждения операции пользователем.

	Источники данных	Открывает окно **Источники данных** с таблицей имеющихся источников и кнопками создания и удаления источника. Удаление источника доступно только в случае, если позиции для него отсутствуют.

Для загрузки новых или обновления существующих пакетов переобучения следует:

- нажать кнопку Загрузить из csv;
- в открывшемся окне **Выберите источник** (Рисунок 398) установить курсор на наименовании нужного источника данных и нажать кнопку **Ок**;

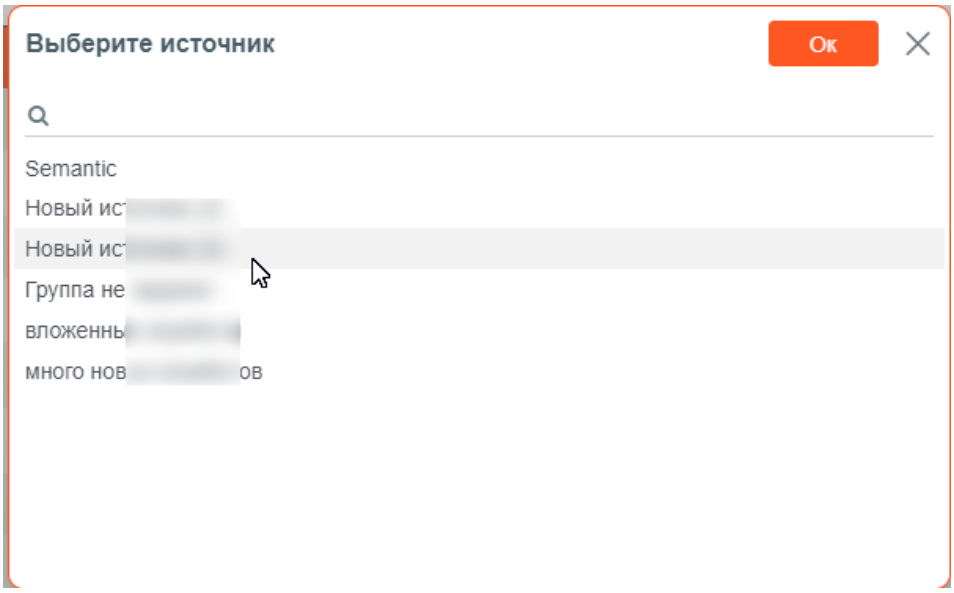


Рисунок 398 - Выбор источника данных при загрузке файла csv

- в открывшемся окне **Подтверждение** установить необходимые опции:
  - **Удалить образцы в источнике перед загрузкой** - перед загрузкой будут удалены данные для выбранного источника. Используется, если загружаются обновленные данные по всему источнику, а старые данные не актуальны;
  - **Удалить образцы в группе источника перед загрузкой** - перед загрузкой будут удалены данные по образцам для группы выбранного источника. Будут удалены данные только по повторно загружаемым группам для выбранного источника;

- **Обучать загружаемые образцы всем атрибутам группы** - для загружаемых образцов будет выполнено машинное обучение по всем атрибутам группы, имеющимся в образцах. Способ учитывает наличие отсутствующих значений для атрибутов.

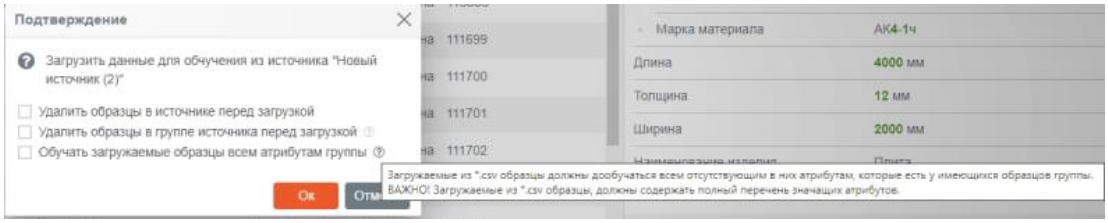


Рисунок 399 - Выбор способа загрузки данных из csv-файла

После нажатия кнопки **ОК** будет выполнена загрузка позиций со значениями атрибутов в соответствие с выбранным источником.

Для добавления нового источника данных нужно выполнить следующие действия:

- нажать кнопку **Источники данных**;
- в открывшемся окне **Источники данных** (Рисунок 400) нажать кнопку ***Создать***;
- отредактировать наименования источника по умолчанию (*Новый источник (номер нового источника)*);
- подтвердить создание нового источника нажав в окне **Подтверждение** кнопку ***Ок*** или отказаться, нажав кнопку ***Отмена***.

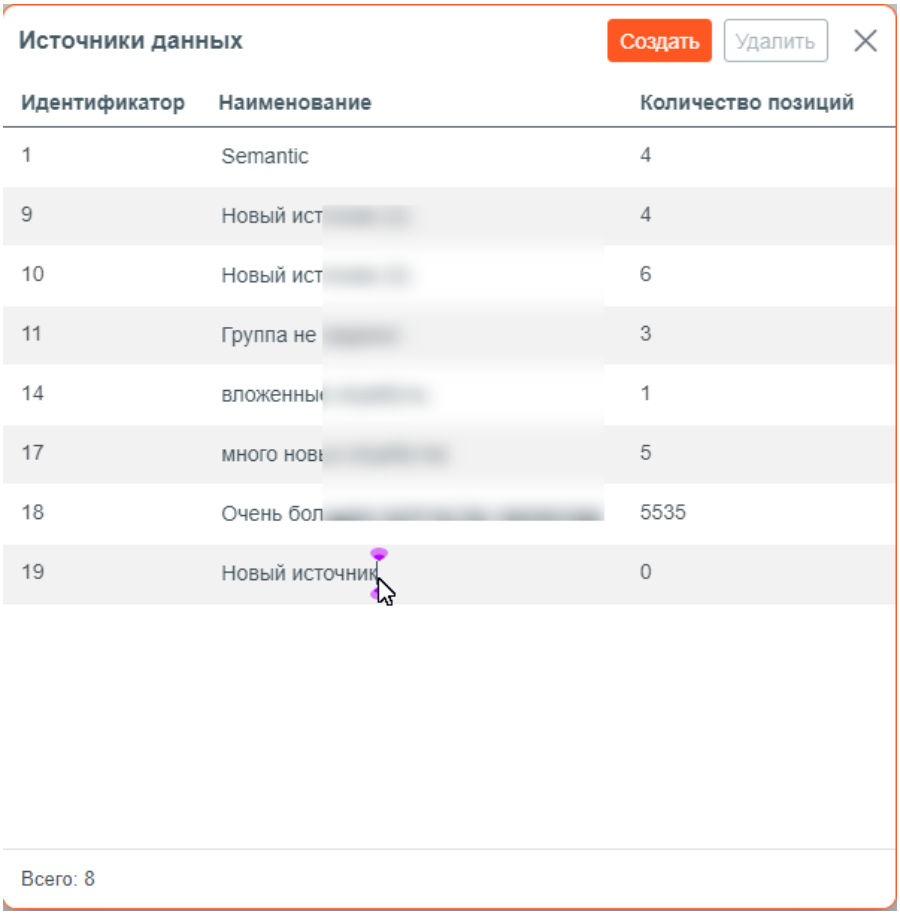


Рисунок 400 - Создание нового источника данных

Таблица с источниками содержит следующие колонки:

- **Идентификатор** - идентификатор источника (уникальный в пределах подсистемы ML) устанавливается автоматически;
- **Наименование** - наименование источника;
- **Количество позиций** - количество позиций, имеющихся для данного источника. Устанавливается автоматически при указании источника загрузки данных.

Справа от таблицы с данными располагается карточка выделенной на сетке данных позиции со значениями атрибутов и альтернативными наименованиями. Атрибуты расположены в том же порядке, как их декларации в справочной группе. Для атрибутов связи, наименование формируется через точку с указанием пути до вложенного атрибута. Сортировка вложенных атрибутов на первом уровне идет по агрегации, а на последующих вложенных атрибутах по наименованию. Альтернативные

наименования автоматически собираются из заявок пользователей на создание новых позиций в *Системе*.

3.27.1.5 Данные для проверки

Основными задачами , решаемыми на вкладке **Данные для проверки** (Рисунок 401) являются:

- формирование проверочных данных, на основе которых будет оцениваться качество машинного обучения;
- поиск групп с недостаточным количеством проверочных позиций;
- автоматическое и ручное назначение проверочных позиций.

На вкладке расположена таблица со справочными группами, для которых в подсистему машинного обучения были загружены данные. В таблице доступны следующие колонки:

- **Группа** – полный путь к группе, начиная от общей родительской группы для всех имеющихся в таблице групп. По щелчку левой кнопкой мыши на группе открывается вкладка **Данные для обучения** с включенным фильтром по выбранной группе;
- **Доступно для проверки** – количество позиций с полной информацией по атрибутам, для которых установлен маркер **Содержит значения всех атрибутов**;
- **Проверочных позиций** – количество позиций, выбранных для проверки качества машинного обучения. Выбор позиций осуществляется только среди тех, которые содержат полные данные по атрибутам;
- **% проверочных** – процентное соотношение проверяемых позиций к общему количеству позиций в группе.

Настройка видимых колонок таблицы выполняется по кнопке ***Настройка таблицы***.

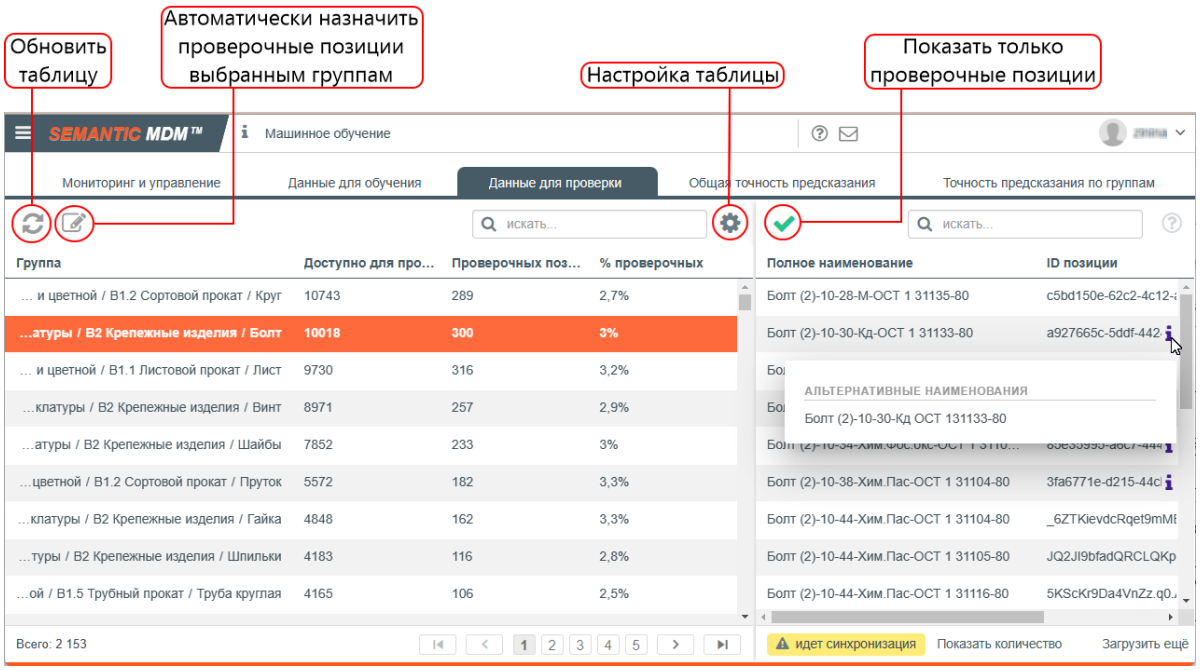


Рисунок 401 - Вкладка данных для проверки качества обучения

В Системе могут быть позиции, для которых не указана группа, но они содержат данные по сквозным атрибутам. Для таких позиций в таблице с группами выведена системная группа *Группа не задана*. Это системное значение также доступно в фасетной фильтрации по группам.

Сортировка доступна по всем колонкам таблицы. По умолчанию включена сортировка по колонке **Группа**. Сортировка по колонке **Группа** выполняется по наименованию полного пути.

Фасетная фильтрация на этой вкладке доступна только по колонке **Группа**. При раскрытии диалога с фасетами отображается «худое» (Рисунок 402) дерево групп с возможностью выбора конечных или промежуточных групп в иерархии.

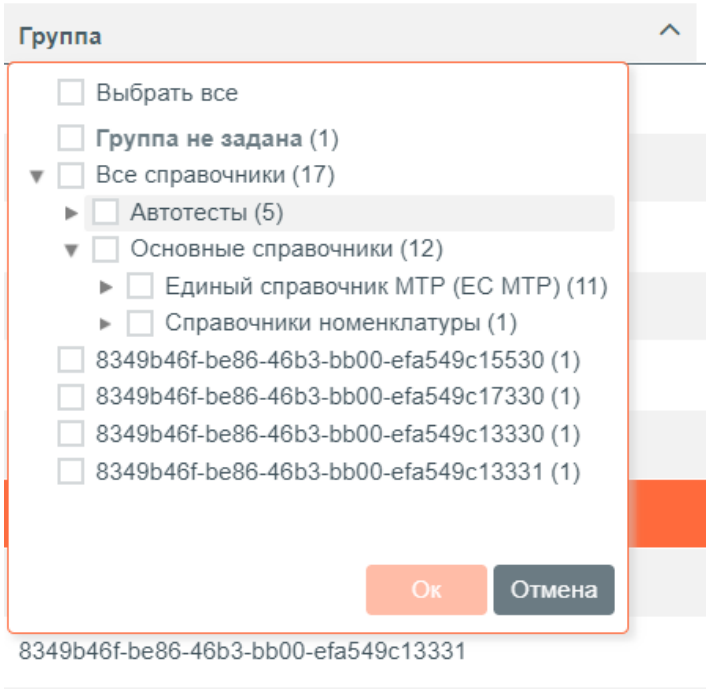


Рисунок 402 - Вид «худого» дерева в фасетном поиске по группам

Над таблицей с группами расположена кнопка **Автоматически назначить проверочные позиции выбранным группам**. По кнопке поднимается окно **Автоназначение проверочных позиций** (Рисунок 403), в котором пользователь может указать процент проверочных позиций. Процент должен находится в диапазоне 0–25%. 0% – означает сброс маркера со всех позиций выбранных групп. 25% – максимально допустимый процент позиций, которые можно выделить для проверки. Количество проверяемых записей по заданному проценту определяется путем округления вниз.

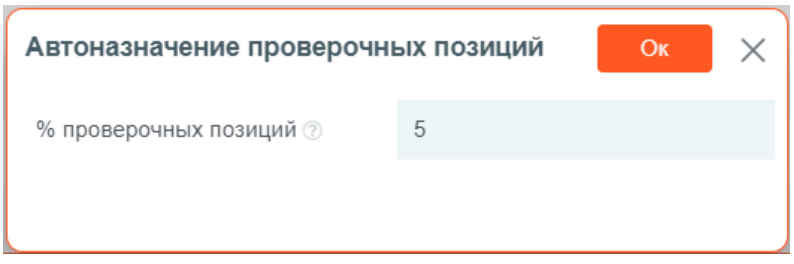
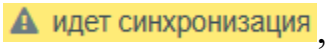


Рисунок 403 - Диалог указания процентного показателя проверочных позиций

Справа от таблицы групп располагается таблица позиций, в которой отображаются доступные для проверки позиции с полными данными по атрибутам со следующими колонками:

- **Полное наименование** – наименование позиции. По этому полю доступна сортировка;
- **ID позиции** – идентификатор позиции (GUID) с возможностью выделить и скопировать данный идентификатор как текст;
- **Источник** – наименование источника, из которого была загружена позиция. По этому полю доступна сортировка и фасетная фильтрация.

Все позиции для выделенной в таблице слева группы по умолчанию отсортированы по наименованию. Над таблицей позиций расположена кнопка **Показать только проверочные позиции**, при нажатии на которую в таблице останутся только проверочные позиции. Проверочные позиции промаркированы «галочкой» **светло–зеленого** цвета справа. Назначение / снятие проверочной позиции осуществляется двойным кликом мыши по позиции. Если для позиции имеется список альтернативных наименований, то справа от наименования позиции отображен значок с пиктограммой  , при наведении на который будет отображена подсказка со списком альтернативных наименований.

Если под таблицей позиций видна строка 

, то это означает, что идет синхронизация данных между сервисами *Системы*. Время синхронизации зависит от показателей системно-технической инфраструктуры.

Для выполнения поиска, над таблицей позиции предусмотрено поле быстрого поиска. Поиск выполняется по ID позиции, полному наименованию, значению любого атрибута и альтернативному наименованию.

3.27.1.6 Общая точность предсказания

Основными задачами данной вкладки являются:

- получение общей оценки точности предсказания подсистемы машинного обучения;
- получение оценки точности предсказания групп для проверочных позиций;

- получение оценки точности предсказания атрибутов для проверочных позиций;
- выполнение сравнения текущих показателей точности с точностью на выбранную из истории дату.

Проверка точности предсказания выполняется на основе наименований позиций, назначенных проверочными, а также на основе их альтернативных наименований. Таким образом, проверяемыми наименованиями следует считать наименования проверочных позиций и их альтернативные наименования.

Основой данного представления является график точности предсказания атрибутов для проверяемых наименований (Рисунок 404). Осями графика являются:

- ось Y – откладывается количество проверяемых наименований в процентах. Начало оси Y начинается с 0%;
- ось X – откладывается количество правильно распознанных атрибутов в процентах для проверяемых наименований.

На графике точности предсказания отображается следующая информация:

- **точность предсказания групп** – пунктирная линия изумрудного цвета, выходящая из точки на оси Y, соответствующей проценту проверяемых наименований, для которых группа была предсказана верно. В середине линии расположена надпись с указанием точного процента правильно предсказанных групп;



Рисунок 404 - График общей точности предсказания

- **точность предсказания с учетом пустых атрибутов** – линия графика светло–фиолетового цвета (цвет может быть изменен разработчиком). При расчете количества правильно предсказанных атрибутов (для оси X) для каждого проверяемого наименования учитываются правильные предсказания в том числе, пустых атрибутов, для которых отсутствует значение в проверяемой позиции;
- **точность предсказания без учета пустых атрибутов** – линия графика светло–оранжевого цвета (цвет может быть изменен разработчиком). При расчете количества правильно предсказанных атрибутов (для оси X) для каждого проверяемого наименования учитываются правильные предсказания только для заполненных в проверяемой позиции атрибутов. Правильно предсказанные пустые значения для атрибутов не учитываются;
- **точность предсказания с учетом пустых атрибутов на выбранную дату** – пунктирная линия графика светло–фиолетового цвета (цвет может быть изменен разработчиком). Показатель отображается на выбранную дату;
- **точность предсказания с учетом пустых атрибутов на выбранную дату** – пунктирная линия графика светло–оранжевого цвета (цвет

может быть изменен разработчиком). Показатель отображается на выбранную дату.

Справа от графика расположен дополнительный параметр, позволяющий выбрать дату, на которую необходимо сравнить показатели точности. По умолчанию ползунок установлен на текущей дате и на графике отображены только данные по актуальной точности. В случае изменения даты с помощью ползунка на графике дополнительно отображаются те же показатели, но уже на выбранную дату. В качестве минимальной даты для сравнения выводится минимальная дата из исторических данных.

3.27.1.7 Точность предсказания по группам

Основными задачами данной вкладки (Рисунок 405) являются:

- получение оценки точности для каждой группы;
- получение оценки точности для каждого атрибута выбранной группы;
- просмотр и анализ ошибок предсказания по каждому из проверяемых наименований;

Все оценки точности предсказания выполняются с учетом пустых значений.

Основным объектом вкладки является таблица с точностью по группам, в которой отображаются следующие столбцы:

- **Группа** – полный путь к группе, начиная от общей родительской группы для всех имеющихся в таблице групп. По клику на группу открывается вкладка «Данные для обучения» с включенным фильтром по выбранной группе;
- **Всего наименований** – количество наименований позиций и альтернативных наименований для данной группы;
- **Проверочных** – количество проверяемых наименований позиций, включая альтернативные наименования;
- **% проверочных** – процентное соотношение проверяемых наименований к общему количеству наименований в группе;

- **50%** – процентное соотношение количества наименований в группе, для которых было верно предсказано более 50% процентов атрибутов, к общему количеству наименований в группе;
- **60%** – процентное соотношение количества наименований в группе, для которых было верно предсказано более 60% процентов атрибутов, к общему количеству наименований в группе;
- **70%** – процентное соотношение количества наименований в группе, для которых было верно предсказано более 70% процентов атрибутов, к общему количеству наименований в группе;
- **80%** – процентное соотношение количества наименований в группе, для которых было верно предсказано более 80% процентов атрибутов, к общему количеству наименований в группе;
- **90%** – процентное соотношение количества наименований в группе, для которых было верно предсказано более 90% процентов атрибутов, к общему количеству наименований в группе;
- **100%** – процентное соотношение количества наименований в группе, для которых было верно предсказано более 100% процентов атрибутов, к общему количеству наименований в группе.

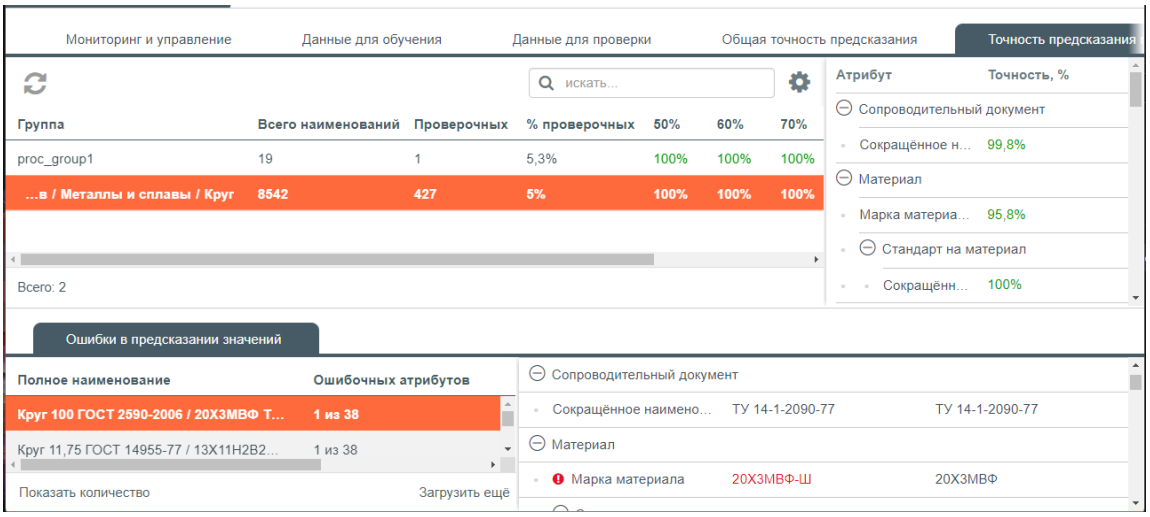


Рисунок 405 - Вкладка с оценками точности предсказания по группам

Сортировка доступна по всем колонкам. По умолчанию сортировка выполняется по колонке **Группа** по наименованию полного пути.

Фасетная фильтрация доступна только по колонке **Группа**. При раскрытии диалога с фасетами отображается «худое» дерево групп с возможностью выбора конечных или промежуточных групп в иерархии.

Справа от таблицы с точностью по группам отображена карточка с точностью по атрибутам группы. Атрибуты отсортированы в таком же порядке, как отсортированы их декларации в группе. Для вложенных атрибутов (например, *Материал.Марка материала*) наименование формируется через точку с указанием пути до вложенного атрибута. Сортировка вложенных атрибутов на первом уровне идет по агрегации, а на последующих вложенных атрибутах по наименованию. Для каждого атрибута посчитана и выведена точность предсказания (в процентах) в рамках группы.

Под таблицей с точностью по группам расположена панель **Ошибки в предсказании значений**, где отображена таблица с проверочными наименованиями, для которых не все значения атрибутов были предсказаны верно. Таблица с проверочными наименованиями отсортирована по наименованиям. Справа от таблицы с проверочными наименованиями отображена карточка атрибутов с предсказанными и эталонными значениями. Атрибуты отсортированы тем же способом, что и для карточки с точностью по атрибутам группы.

3.28 Настройка пользователей и ролей

В *Semantic MDM* принята групповая политика безопасности, которая может дополняться индивидуальными настройками для каждого пользователя на уровне справочных групп и атрибутов.

Согласно принятой политике безопасности, если пользователь выполняет несколько ролей с разными привилегиями, то его привилегии при работе со справочниками суммируются.

Права отдельного пользователя, назначенные на уровне справочной

группы или атрибута (атрибутов) имеют приоритет перед правами роли, которую он выполняет.

3.28.1 Привилегии пользователей

Политика безопасности на уровне ролей пользователей определяется набором привилегий (прав), разрешённых для каждой роли. В *Semantic MDM* принят набор привилегий, представленный в Таблица 29:

Таблица 29. Привилегии пользователей

Привилегия	Описание
Редактирование данных	Пользователю разрешено создавать пакет изменений. Если для пользователя отсутствует данная привилегия, то пользователь может создать пакет изменений только в рамках работы над заявкой. Если пользователь создаст пакет изменений без заявки, то будет автоматически создана системная заявка с требованием выполнить согласование по всем маршрутам согласования, определенным в справочных группах, в которых пользователь вносил изменения в данные. Пользователю с данной привилегией разрешено контролировать качество данных в подсистеме **Качество данных**
Редактирование данных без заявок	Пользователь может создавать пакет изменений без заявки
Управление историчностью данных	Пользователь может выполнять настройку исторических значений в *Системе*
Редактирование метаданных	Пользователь может выполнять настройку метаданных в подсистеме **Конфигурация данных**
Экспорт позиций и шаблонов	Пользователю разрешен экспорт видимых ему позиций в файлы форматов Excel/CSV
Менеджер заявок	Пользователь с данной привилегией может: • видеть все заявки и выполнять над ними любые доступные действия; • видеть вкладку **Отчет SLA**; • настраивать группы пользователей в заявках, которые используются при настройке исполнителей, диспетчеров и маршрутов согласования; • настраивать рабочее время отдела и персональный календарь сотрудников; • настраивать замещающего пользователя; • осуществлять контроль качества данных
Управление безопасностью	Пользователь имеет полный доступ к функциям управления безопасностью. Ему разрешено редактирование пользователей, ролей и матрицы доступа в подсистеме **Администрирование.** Данная привилегия включает в себя привилегию *Чтение безопасности*

Чтение данных	Пользователь может читать данные справочников / классификаторов с учётом настроенных прав доступа
Чтение безопасности	Пользователю разрешен просмотр пользователей, ролей и матрицы доступа в подсистеме **Администрирование** в режиме доступа *Только чтение*. Пользователь может изменять права доступа в тех справочных группах, для которых ему установлен уровень *Полный доступ*
Чтение журнала работы пользователей	Пользователю доступен для чтения журнал работы пользователей в подсистеме **Администрирование.** Пользователь может также просматривать журнал из карточки позиции, атрибута, карточки внешней системы и обменного класса
Просмотр всех справочников	Пользователь имеет доступ к подсистеме **Справочники** и на стартовой странице для него будет доступен системный справочник *Все справочники*. Если привилегия выключена, то: • доступ к справочникам осуществляется только со стартовой страницы, на которой будут отображены только те справочники, для которых включена видимость для пользователя в подсистеме **Конфигурация данных**; • в подсистеме **Качество данных** будет доступен выбора справочника, для которого необходимо выполнить анализа качества данных
Формирование отчетов	Пользователю доступна функция формирования отчетов в подсистеме **Отчеты**
Поиск дубликатов	Пользователю разрешено выполнять поиск дубликатов в справочнике по уже настроенным вариантам поиска.
Системное администрирование	Пользователю разрешен доступ в подсистемы **Машинное обучение**, **Описание компонентов**, **Импорт и репликация**, **Настройки**
Просмотр истории	Пользователю доступен просмотр истории изменения позиции

Если для пользователя не включена привилегия **Редактирование данных**, то пользователь сможет видеть и искать только позиции со статусом *Нормализована*.

При включении для пользователя привилегии **Экспорт позиций и шаблонов** и не имеющего привилегии **Редактирование данных**, будут открыты функции экспорта позиций и шаблона позиции в MS Excel и CSV, а также экспорт описания групп в MS Word.

В Системе существует запрет на вход пользователей, для которых выключены все привилегии.

3.28.2 Создание новой роли

В системе выделяется системная роль **Все пользователи**, которая присваивается автоматически каждому новому пользователю и которая определяет какими минимальными правами доступа и привилегиями должен обладать любой новый пользователь. Если пользователю назначено более одной роли, то пользователь получает максимальные привилегии и права доступа из назначенных ему ролей. Если для пользователя настроены индивидуальные права доступа к объекту, то они перекрывают любые настроенные права доступа к объекту через роли.

Для того, чтобы создать новую роль следует:

- перейти в подсистему **Администрирование** используя кнопку ***Навигация***;
- перейти на вкладку **Роли** (Рисунок 406) нажать кнопку ***Добавить***;
- в карточке роли указать наименование и дать её описание (по необходимости);
- установить маркер **Синхронизация с LDAP**, если предусмотрено получение данных о пользователе из ActiveDirectory (см. раздел 3.29);
- в разделе **Привилегии** щелчком левой кнопки мыши по соответствующей привилегии установить набор прав для созданной роли.

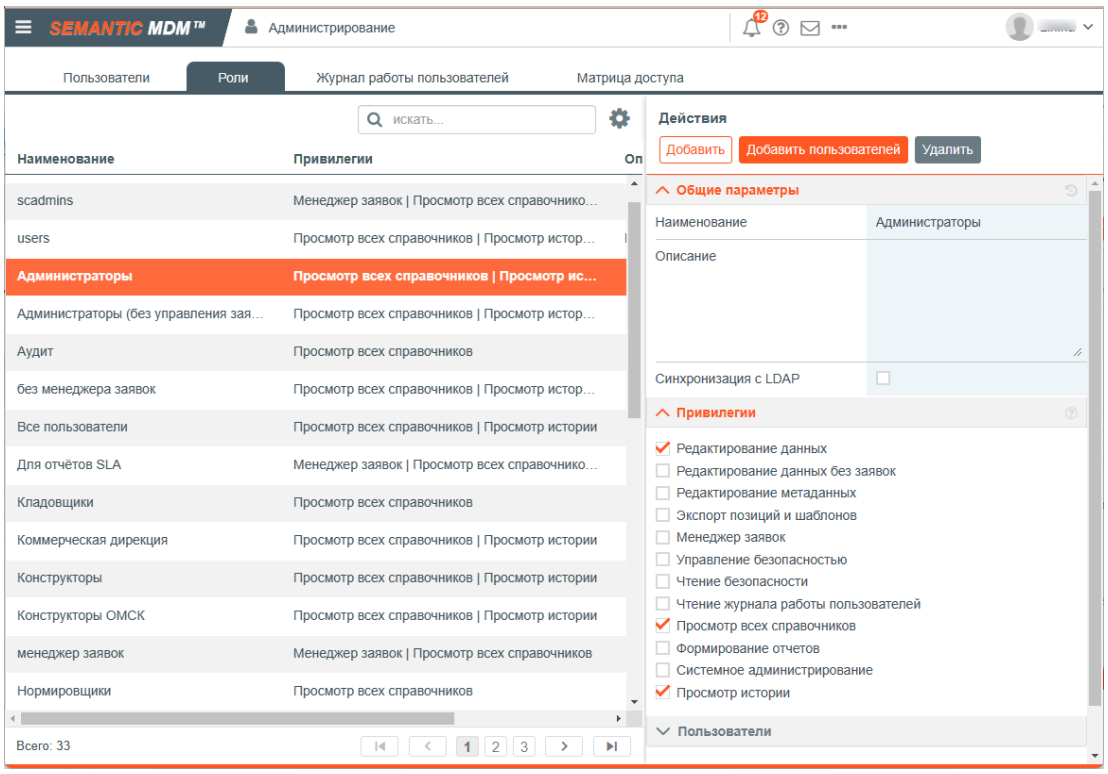


Рисунок 406 - Работа с ролями пользователей

Чтобы удалить роль нужно щёлкнуть по кнопке ***Удалить***, расположенной над карточкой справа.

Кнопка ***Добавить пользователей*** откроет окно **Пользователи**, где, пользуясь мышью, можно выделить в таблице необходимых пользователей (чтобы выделить нескольких нужно держать нажатой клавишу <*Ctrl*>). Для поиска пользователей можно воспользоваться всеми вариантами сортировки и фильтрации. После выбора нужных пользователей нажать внизу справа кнопку ***Выбрать*** и они будут добавлены в соответствующий раздел карточки. Чтобы удалить пользователя из роли нужна нажать кнопку ✖ ***Удалить роль у пользователя*** справа в строке с его логином.

3.28.3 Добавление пользователя. Карточка пользователя

Чтобы добавить нового пользователя необходимо:

- перейти в подсистему **Администрирование** используя кнопку ***Навигация***;
- на вкладке **Пользователи** (Рисунок 407) нажать кнопку ***Добавить***;

- в появившейся карточке пользователя внести следующие сведения:
  - **Учётная запись** – логин пользователя, используемый при входе в *Систему*;
  - **Изображение** – фотография или рисунок, добавленные из графического файла. Для добавления необходимо нажать на кнопку ***Загрузить изображение*** (появляется при наведении курсора на область изображения) слева от области рисунка и выбрать нужный файл из *Проводника*;
  - **Активация** – установленный маркер разрешает пользователю авторизоваться и работать в *Системе*;
  - **Запретить вход в веб-клиент** - опция, обеспечивающая блокировку пользователя. При установленном маркере вход в *Систему* для данного пользователя невозможен;
  - **Электронная почта** – адрес электронной почты текущего пользователя;
  - **Дублировать уведомления МУЗ по email** – опция, при установке которой пользователь будет получать уведомления на электронную почту из подсистемы **Журнал заявок** о наличии заявок требующих реакции пользователя;
  - **Фамилия** – фамилия пользователя;
  - **Имя** – имя пользователя. Отображается под фотографией в профиле пользователя;

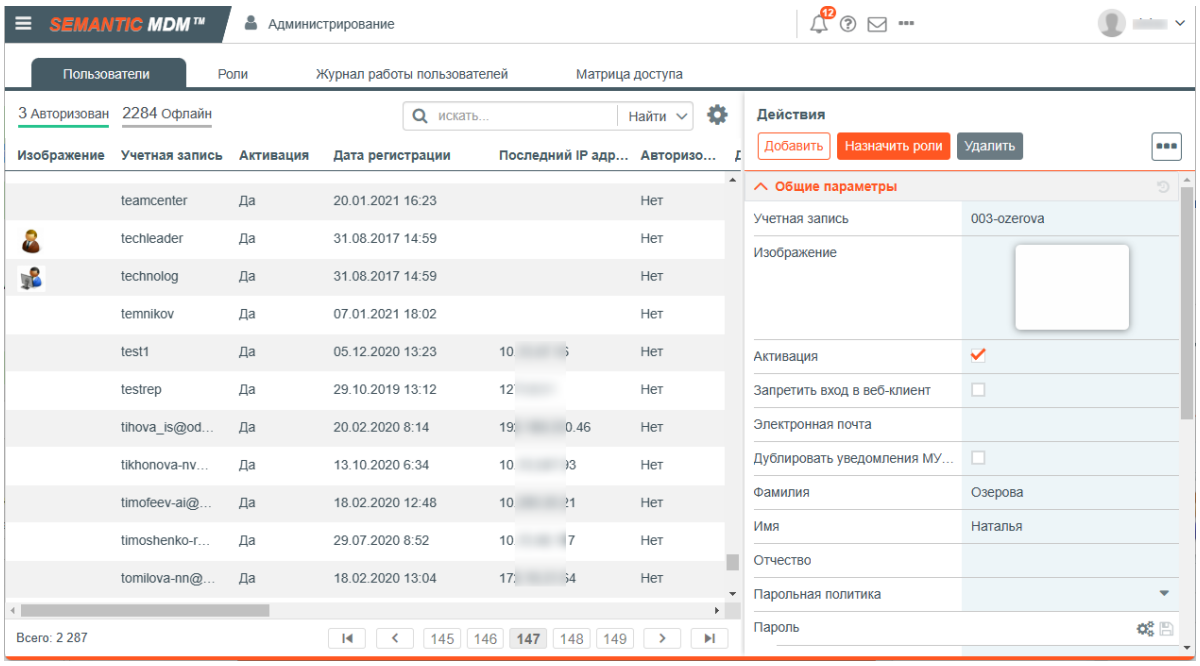


Рисунок 407 - Вид вкладки **Пользователи** и карточка пользователя

- **Отчество** – отчество пользователя;
- **Парольная политика** – выбрать преднастроенную парольную политику (см. п. 3.28.5), применяемую для данного пользователя;

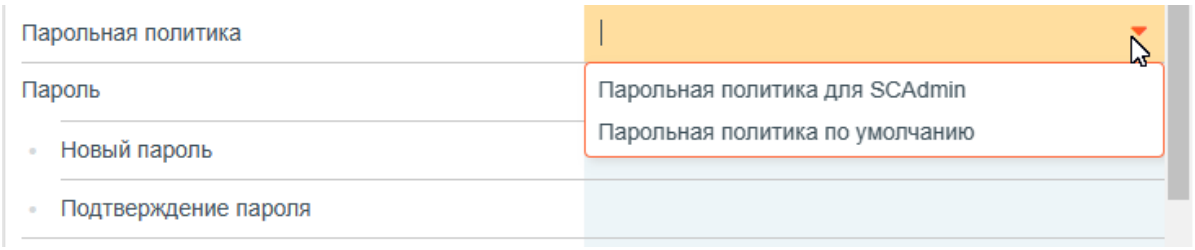


Рисунок 408 - Выбор парольной политики для пользователя

- **Пароль** - ячейка с кнопками **Сгенерировать и отправить новый пароль** и **Сохранить пароль**:
  - **Новый пароль** – пароль входа в *Систему*;
  - **Подтверждение пароля** – повтор нового пароля для входа в *Систему*;

Если по кнопке ***Сгенерировать и отправить новый пароль*** был сгенерирован пароль, не соответствующий парольной политике, то автоматически включается опция **Запросить изменение пароля при следующем входе**.

- **Запросить изменение пароля при следующем входе** - маркер дает возможность для выбранного пользователя запросить смену пароля;
- **Замещающий на время отсутствия** – логин пользователя (выбирается из выпадающего списка или в окне **Заместители**, вызываемом кнопкой ***Выбрать***), который замещает данного при его отсутствии согласно рабочему расписанию. Заместитель может быть указан непосредственно в подсистеме **Журнал заявок** при настройке рабочего расписания;
- **Роли** – отдельный раздел, где маркером отмечаются группы, в которые входит пользователь. Добавление новой группы может быть также осуществлено через всплывающее окно, вызываемое кнопкой ***Назначить группы***;
- **Привилегии** - отдельный раздел, где можно увидеть суммарный набор привилегий пользователя по всем ролям, к которым он относится.

Кроме сведений, которые указываются при регистрации нового пользователя, в карточке присутствуют дополнительные ячейки, где содержится системная (не редактируемая) информация:

- **Дата регистрации** – дата и время (часы и минуты) регистрации пользователя в *Системе*;
- **Последний IP адрес** – интернет-адрес, с которого осуществлялся последний вход;
- **Авторизован** – пользователь работает в Системе в данный момент;
- **Дата и время последнего входа** – дата и время последнего захода пользователя в *Semantic MDM*;
- **Связанная позиция** – для данного пользователя есть связанная с ним позиция в справочнике пользователей (не системном);
- **Тип пользователя** – используется два типа пользователей. *Штатный* означает, что пользователь входит в *Систему* по логину и паролю, указанным в карточке. *LDAP* - пользователь авторизуется в *Системе* как пользователь Windows.

В левой части вкладки **Пользователи** располагается таблица с пользователями, в которой содержаться сведения из карточек. Столбцы таблицы настраиваемые. Кнопка настройки располагается рядом с полем быстрого поиска. Таблица поддерживает расширенный поиск по дате регистрации и дате и времени последнего входа. Параметры могут быть заданы с помощью специальной шкалы или вручную в выпадающей панели с календарем и часами.

Над таблицей расположены ссылки для быстрой фильтрации пользователей по наличию авторизации – *Авторизован* или *Офлайн*. В таблице может быть также произведена сортировка по любому из столбцов, кроме столбца **Изображения**. Фасетный поиск может быть применён во всех столбцах кроме – **Учётная запись**, **Изображение**, **Дата регистрации, Дата и время последнего входа**.

Удаление пользователя выполняется кнопкой ***Удалить***, расположенной над карточкой пользователя, и требует подтверждения во всплывающем окне (Рисунок 409).

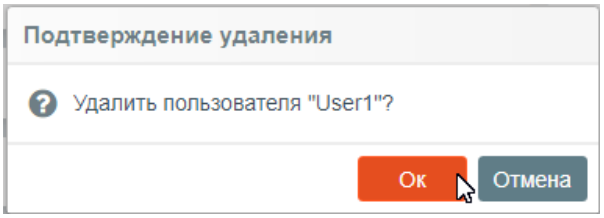


Рисунок 409 - Подтверждение удаления пользователя

Находясь на сетке данных в контекстном меню (Рисунок 410) доступны следующие команды:

- **Открыть в новой вкладке** – *Система* открывает еще одну вкладку браузера с пользователями;
- **Добавить пользователя** – добавление нового штатного пользователя в Систему.;
- **Удалить** – удаление пользователя;

- **Назначить роли** – вызывает всплывающее окно **Назначение ролей** со списком, в котором можно отметить какие роли выполняет текущий пользователь;
- **Сгенерировать и отправить новый пароль** – позволяет автоматически сгенерировать и отправить пользователю новый пароль для входа в систему. Генерация пароля не работает для пользователей, у которых не указана электронная почта;
- **Запросить изменение пароля при следующем входе** – будет сформирован запрос для пользователя на изменение пароля при его следующем входе в *Систему*;
- **Запросить изменение пароля при следующем входе всем** – будет сформирован запрос для всех пользователей на изменение пароля при их следующем входе в *Систему*;
- **Журнал работы пользователя** - описан в п. 3.28.6.

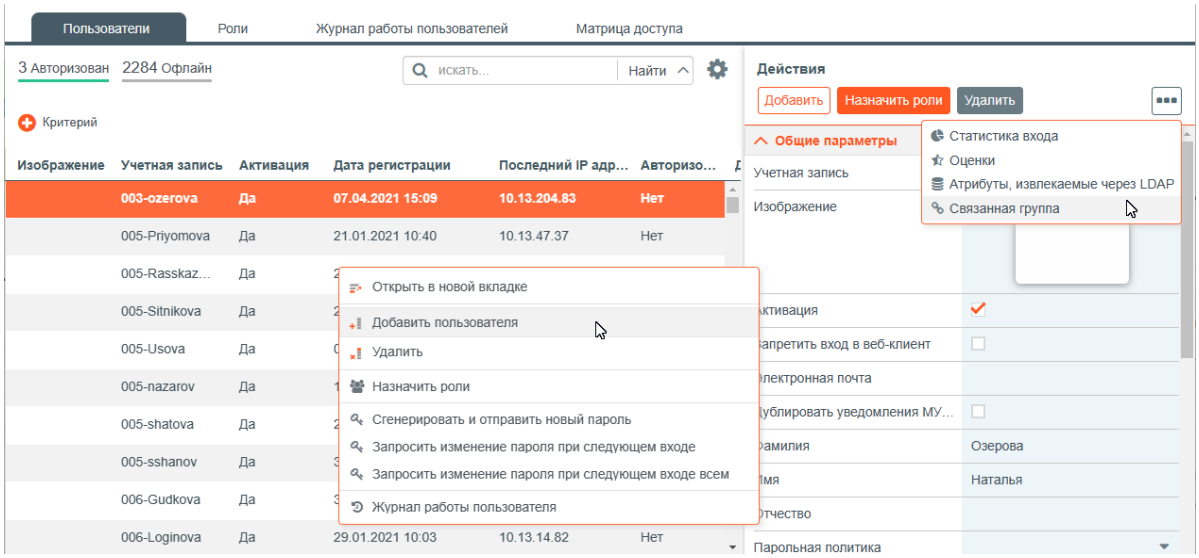


Рисунок 410 - Команды контекстного меню при настройке пользователей

Используя кнопку ***Действия*** над карточкой пользователя можно вызвать следующие команды:

- **Статистика входа** – вызывает отдельное окно, содержащие статистические данные (см.п. 3.28.8)по входам пользователей в Систему;

- **Оценки** – автоматически выгружается файл Excel с оценками, данными пользователями в *Системе* по команде Оцените приложение, расположенной в профиле пользователя;
- **Атрибуты, извлекаемые через LDAP** – список атрибутов, значения которых передаются в карточку пользователя из Active Directory;
- **Связанная группа** - открывает вкладку со справочником пользователей, расположенным в подсистеме **Справочники**. Курсор позиционируется на выбранном пользователе.

3.28.4 Загрузка и выгрузка пользователей в Excel

Экспорт пользователей в Excel позволяет выгрузить по пользователям всю доступную в таблице информацию, за исключением паролей. Для выгрузки следует, находясь в представлении **Администрирование** -> **Пользователи** под кнопкой рядом с профилем пользователя, выбрать команду **Экспорт пользователей в Excel** (Рисунок 411).

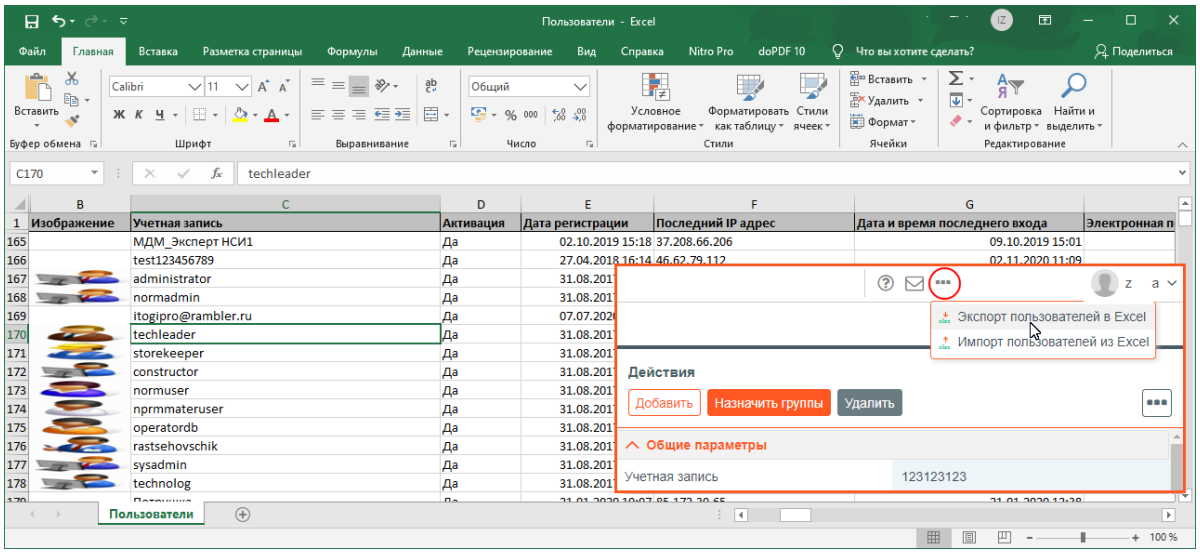


Рисунок 411 - Экспорт пользователей в Excel

Semantic MDM позволяет быстро загрузить пользователей из имеющихся информационных систем через Excel, а также провести массовую модификацию данных по пользователям. Для импорта следует, находясь в представлении **Администрирование** -> **Пользователи** под кнопкой рядом с

профилем пользователя, выбрать команду **Импорт пользователей из Excel**. При выполнении загрузки сопоставление пользователей осуществляется по их GUID, а при его отсутствии по логину. При выполнении импорта новых пользователей, для каждого пользователя автоматически генерируется новый 8-ми значный пароль, который отправляется пользователю по email.

3.28.5 Парольная политика

В *Semantic MDM* присутствует возможность настройки парольных политик для локальных учетных записей. *Система* допускает ведение одной и более парольных политик, одна из которых всегда является политикой по умолчанию и автоматически назначается всем новым пользователям. Доступ к просмотру и настройкам парольных политик выполняется через кнопку на основной панели инструментов подсистемы **Администрирование** (Рисунок 412).

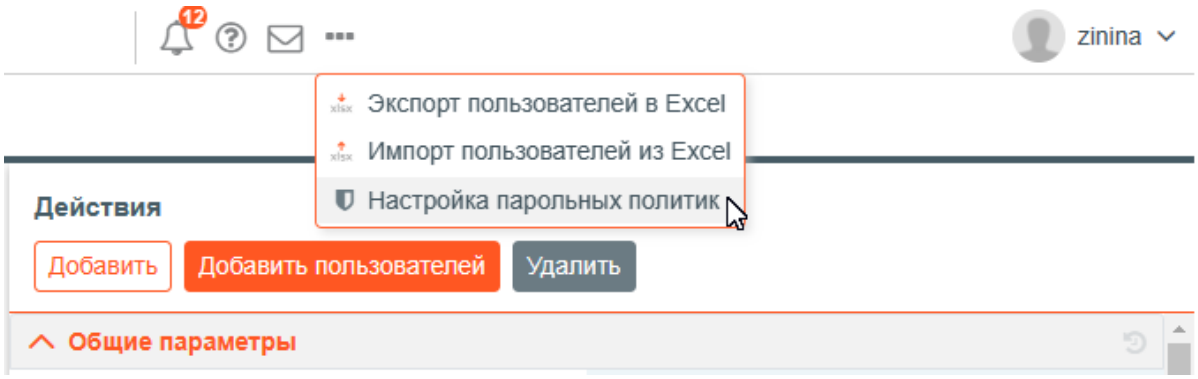


Рисунок 412 - Вызов диалога для просмотра и настройки парольных политик

В диалоге настройки парольной политики (Рисунок 413) доступны следующие параметры:

- **Наименование** – название парольной политики, которое будет отображаться при настройке пользователей;
- **Маска** – регулярное выражение, которому должен удовлетворять вводимый пользователем пароль входа в *Систему*;

- **Подпись с требованиями к паролю** – описание требований к создаваемому паролю, которое отображается пользователю;
- **Срок действия** – срок, в течение которого будет действителен пароль. Назначается в днях от 0 до 3650;
- **Проверять отсутствие совпадения с предыдущими паролями** - количество предыдущих паролей, с которыми не должен совпадать новый (от 0 до 25);
- **Количество неуспешных входов до блокировки пользователя** - количество разрешенных не успешных попыток входа в *Систему*;
- **Парольная политика по умолчанию** - опция, назначающая указанную парольную политику политикой по умолчанию для всех вновь добавляемых пользователей.

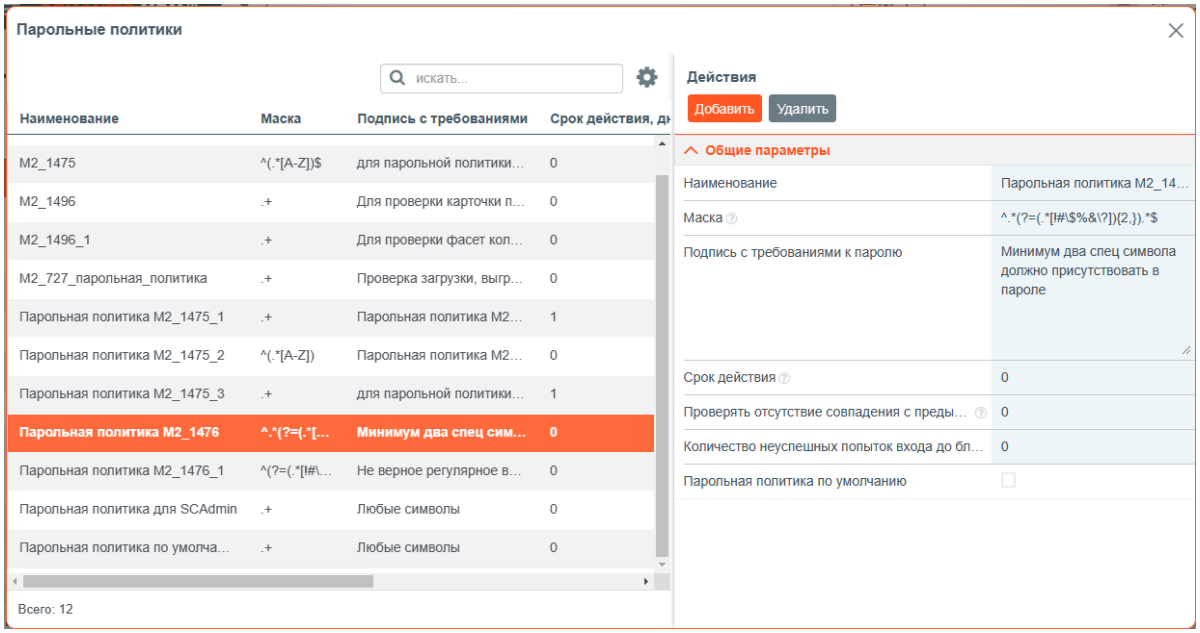


Рисунок 413 - Диалог настройки парольных политик

Для добавления новой парольной политики используется кнопка ***Добавить*** или команда из контекстного меню **Добавить парольную политику**. Для удаления можно использовать кнопку ***Удалить*** и одноименную команду из контекстного меню.

Для поиска парольной политики используется ячейка быстрого поиска, расположенная над таблицей. Рядом с ней расположена кнопка настройки столбцов таблицы.

3.28.6 Журнал работы пользователей

Журнал работы пользователей располагается в представлении **Администрирование** на вкладке **Журнал работы пользователей** (Рисунок 414).

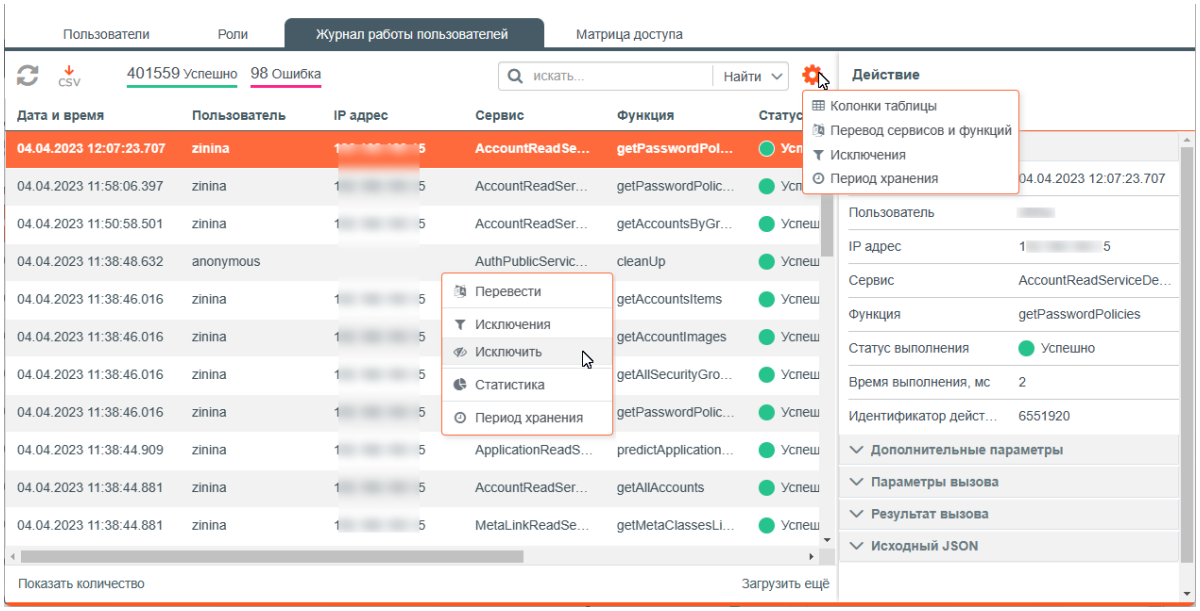


Рисунок 414 - Журнал работы пользователей

Подсистема журналирования отслеживает все функции сервера приложений, которые вызывает веб-клиент, через который работает пользователь. Пользователь, при помощи команды контекстного меню **Исключить**, может исключить из списка функции, которые он не хочет видеть в журнале работы пользователей. Команда **Исключения** позволяет настроить фильтр для исключаемых из просмотра сервисов и функций. Для всех журналируемых сервисов и функций, предусмотрена возможность настройки перевода на русский или английский язык - команда **Перевод сервисов и функций**. Перевод выполняется командой контекстного меню **Перевести**.

Переводы можно выгрузить и загрузить из Excel в одноименном окне настройки.

Все настройки выполняются путём вызова соответствующей команды из меню кнопки ***Настройка***, располагающейся справа от ячейки быстрого поиска или из контекстного меню.

В *Системе* предусмотрено журналирование следующих обязательных параметров:

- Дата и время события;
- IP адрес - с которого было выполнено действие;
- Идентификатор пользователя;
- Логин пользователя;
- Имя сервиса;
- Наименование функции;
- Параметры вызова функции – в формате json;
- Возвращаемый результат функции – в формате json;
- Текст ошибки – если произошла ошибка;
- Статус выполнения - успешно/ошибка;
- Время выполнения действия.

Любой из обязательных параметров с помощью настройки колонок таблицы может быть выведен в таблицу с возможностью поиска через фасетную фильтрацию по всем колонкам, кроме **Дата и время**, **Время выполнения**, **Параметры вызова**, **Результат** и **Идентификатор действия**.

Кроме обязательных параметров, в системе предусмотрено журналирование следующих дополнительных параметров:

- Ссылка на пакет изменений;
- Ссылка на позиции;
- Ссылка на группу;
- Ссылка на атрибут;
- Ссылка на стереотип совместимости;
- Ссылка на справочник;

- Ссылка на классификатор;
- Ссылка на шаблон заявки;
- Ссылка на измеряемую сущность;
- Ссылка на единицу измерения;
- Ссылка на заявку;
- Ссылка на внешнюю систему;
- Ссылка на обменный класс;

По любому из дополнительных параметров можно выполнять расширенный поиск.

Кроме расширенного поиска по журналу, доступен поиск через поле быстрого поиска. Быстрый поиск выполняется по наименованиям функций, их переводам, а также значениям параметров вызова и возвращаемым значениям.

Кроме отображения общего журнала работы пользователей в подсистеме администрирования, в новой версии системы журнал работы можно посмотреть для следующих сущностей в системе:

- Пользователь;
- Группа пользователей;
- Пакет изменений;
- Справочная группа;
- Позиция;
- Атрибут и стереотип агрегации;
- Стереотип совместимости;
- Внешняя система;
- Обменный класс;
- Справочник;
- Классификатор;
- Шаблон заявки;
- Единица измерения;
- Измеряемая сущность.

Доступ к просмотру журнала (Рисунок 415) для любой из сущностей выполняется через специальную кнопку, расположенную в карточке с этой сущностью.

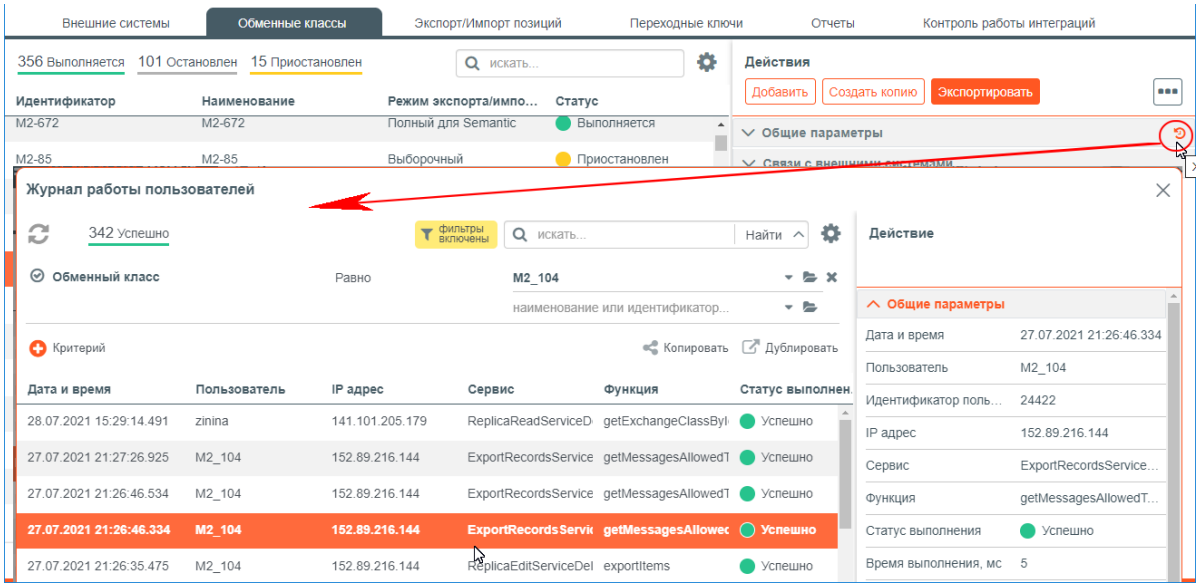


Рисунок 415 - Обращение к журналу работы пользователей из карточки объекта

Для ограничения объёма журнала, в системе предусмотрена настройка срока хранения журнала работы пользователей. Срок настраивается командой **Период хранения**, вызываемой из меню кнопки ***Настройка***. Во всплывающем окне **Период хранения журнала** (Рисунок 416) можно указать:

- *Количество лет* - время хранения в годах. Максимальное количество лет - 10;
- *Количество месяцев* - время хранения в месяцах. Максимальное количество месяцев - 12.

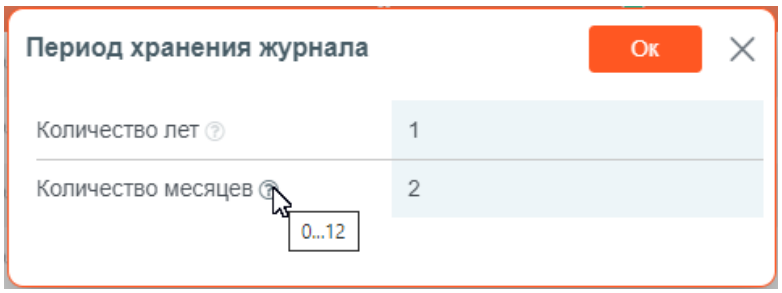


Рисунок 416 - Настройка времени хранения журнала

Журнал работы пользователей может быть выгружен в файл формата CSV. Для этого используется соответствующая кнопка над таблицей журнала. Перед выгрузкой *Система* запрашивает выполнение настроек файла (Рисунок 417), в части используемой кодировки (предлагается выбор UTF-8, cp1251, koi8), символов кавычек (двойные или одинарные), а также вид разделителя.

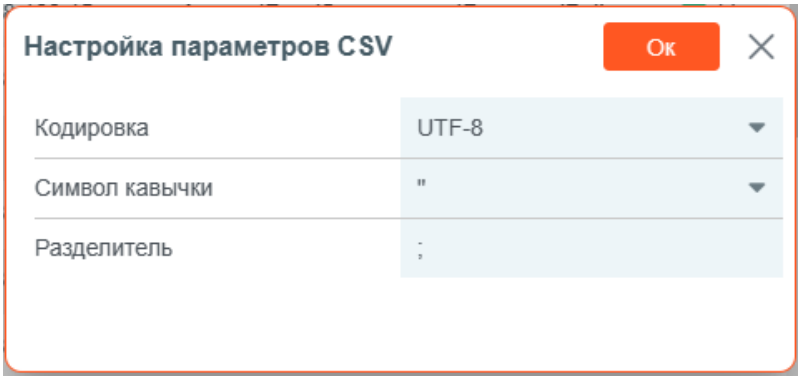


Рисунок 417 - Настройка параметров файла CSV для журнала пользователей

3.28.7 Матрица доступа

Вкладка **Матрица доступа** подсистемы **Администрирование** позволяет выполнять настройку и просмотр настроенных уровней доступа для ролей и пользователей к иерархии справочных групп. Чтение матрицы доступа доступно пользователям, имеющих привилегию *Управление безопасности* или привилегию *Чтение безопасности*. Редактирование матрицы доступа для справочных групп и деклараций атрибутов доступно пользователям с привилегией *Управление безопасностью* в независимости от уровня доступа к справочным группам, а также для пользователей с привилегией *Чтение безопасности* только для справочных групп, где они имеют уровень доступа *Полный доступ*.

В режиме **Роли / Пользователи**, который по умолчанию показывается в матрице доступа, в первой колонке отображается иерархия справочных групп, во второй и последующих колонках отображаются выбранные роли и пользователи, для которых, на пересечении со справочными группами,

отображается текущий уровень доступа. Справа от наименования справочной группы отображаются пиктограммы (Таблица 30), которые показывают наличие в группе настроенной безопасности.

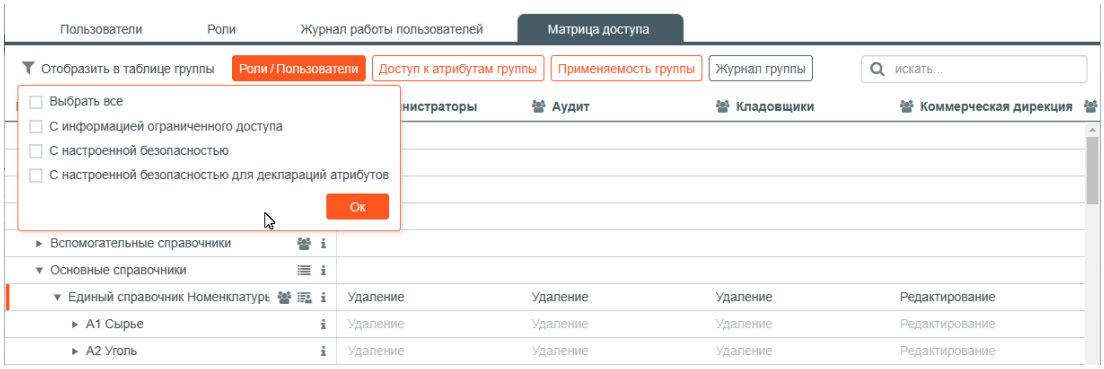
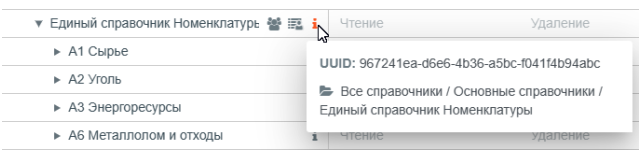


Рисунок 418 - Доступ к просмотру прав доступа в разрезе ролей и пользователей

Таблица 30. Пиктограммы матрицы доступа

Пиктограмма		Описание
	**Есть настройки безопасности для пользователей и (или) ролей**	Справочная группа и ее дочерние группы имеют настройки безопасности (права на группу). Щелчком левой кнопки мыши по пиктограмме можно оставить в таблице столбцы только для тех ролей или пользователей, для которых выполнены настройки безопасности
	**Есть собственные декларации с настроенной безопасностью**	Справочная группа имеет собственные (не унаследованные) декларации атрибутов, для которых настроены права доступа. Щелчком левой кнопки мыши по пиктограмме можно перейти в матрицу доступа по атрибутам данной справочной группы
	**Есть собственные декларации**	Справочная группа имеет собственные (не унаследованные) декларации атрибутов. Щелчком левой кнопки мыши по пиктограмме можно перейти в матрицу доступа по атрибутам данной справочной группы
	**Сведения об UUID и ссылка на справочную группу**	Значение UUID и ссылка появляются во всплывающем окне при наведении курсора на пиктограмму в строке с наименованием справочной группы  Доступ к UUID справочной группы и пути расположения

При нажатии на кнопку ***Роли / Пользователи*** появляется всплывающее окно **Выберите роли и пользователей** (Рисунок 419) для настройки колонок, отображаемых в таблице матрицы доступа. Доступ к настройке также возможен через команду контекстного меню **Настройка видимости колонок**, которое вызывается на заголовке таблицы.

Для настройки достаточно установить маркеры, расположенные перед наименованием роли / пользователя. Дополнительно в окне можно предварительно отфильтровать список для показа только ролей и пользователей с настроенной безопасностью, либо оставить только уже выделенные объекты. После нажатия на кнопку ***Ок*** выбранные роли и пользователи будут отображены в таблице матрицы доступа. Для возврата к состоянию по умолчанию следует выбрать команду контекстно меню **Сбросить настройки таблицы**.

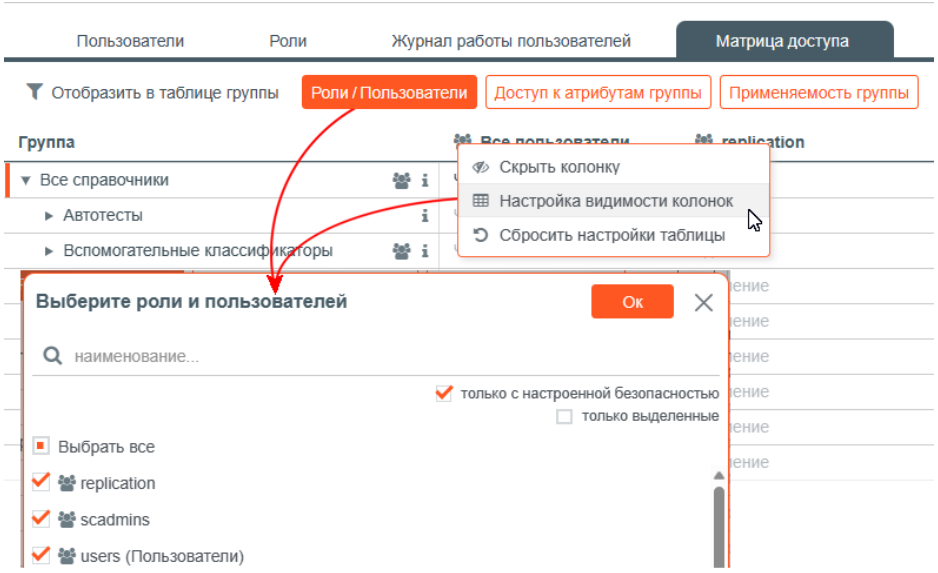


Рисунок 419 - Настройка ролей и пользователей, отображаемых в матрице доступа

Для настройки прав доступа к справочным группам необходимо выполнить двойной щелчок левой кнопкой мыши в ячейке на пересечении столбца с наименованием роли или пользователя и строки с наименованием справочной группы. В открывшемся списке выбрать нужный вариант доступа (Рисунок 420).

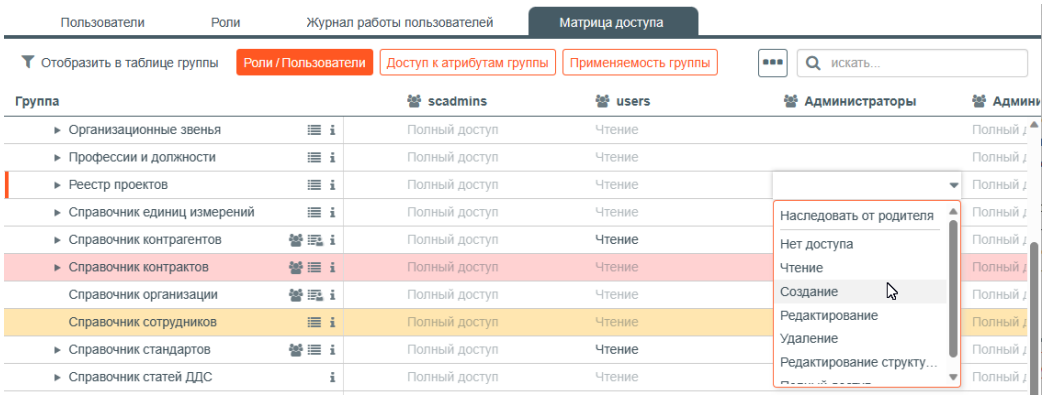


Рисунок 420 - Настройка прав доступа к справочной группе

В режиме **Роли / Пользователи** можно отфильтровать по кнопке ***Отобразить в таблице группы*** представленные в списке справочные группы в соответствии:

- **Выбрать все** - маркер включает все фильтры одновременно;
- **С информацией ограниченного доступа** – в таблице останутся только те справочные группы, в которых настроен признак ИОД (информация ограниченного доступа);
- **С настроенной безопасностью** – в таблице останутся только те справочные группы, у которых имеется настройка прав доступа;
- **С настроенной безопасностью для декларации атрибутов** - в таблице останутся только те справочные группы, у которых имеется настройка прав доступа к декларациям атрибутов.

При включенном фильтре рядом с кнопкой ***Отобразить в таблице группы*** в скобках показано сколько фильтров включено и сколько фильтров всего имеется. Для отключения фильтрации можно снять маркеры напротив соответствующих фильтров либо нажать на кнопку ***Фильтры включены***, рядом с ячейкой быстрого поиска над таблицей справа. Дополнительные команды для работы с матрицей доступа могут быть вызваны из контекстного меню, которое дублирует панель с представлениями матрицы. Если размера окна недостаточно для размещения кнопок перехода к другим представлениям, то они будут свернуты под служебную кнопку, как на рисунке ниже.

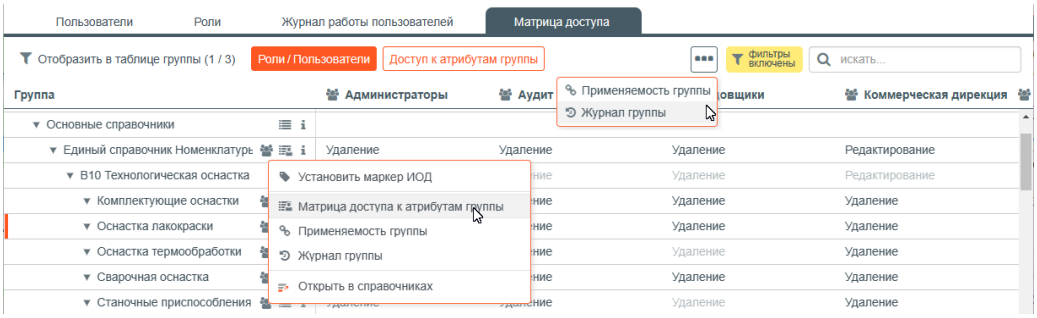


Рисунок 421 – Матрица доступа

В контекстном меню матрицы доступа в режиме **Роли / Пользователи** можно выполнить настройку цветового маркирования справочных групп, содержащих информацию ограниченного доступа. Для этого необходимо установить курсор на наименовании справочной группы в первом столбце таблицы и вызвать из контекстного меню команду **Установить маркер ИОД**, после чего, в открывшемся диалоге (Рисунок 422), выбрать один из предлагаемых вариантов ИОД. В соответствии с выбранным маркером наименование справочной группы получит определенный цветовой фон, благодаря чему справочная группа будет легко определяться по матрице доступа.

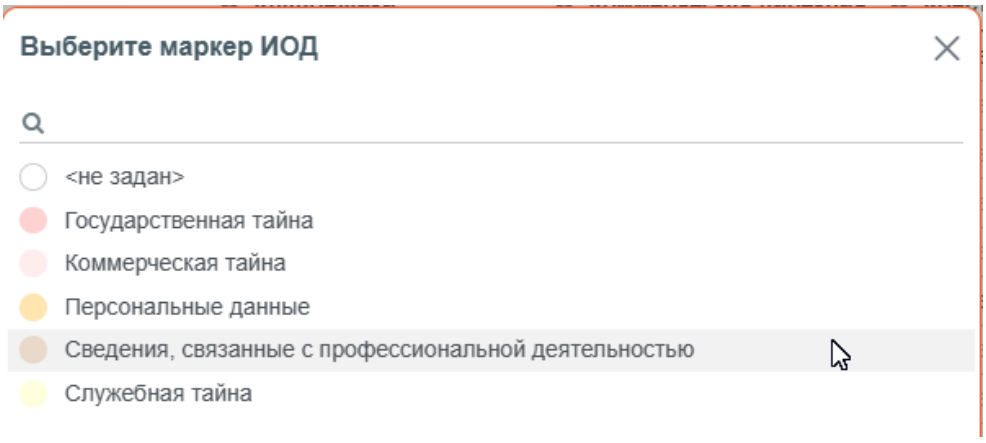


Рисунок 422 - Маркеры для отметки справочных групп с информацией ограниченного доступа

Назначение вида ИОД на справочную группу, позволяет только промаркировать справочную группу соответствующим видом ИОД и не

накладывает каких-либо ограничений на работу с данными в этой группе. Предусмотрены следующие виды ИОД:

- Государственная тайна – бледно-красный цвет;
- Коммерческая тайна – бледно-розовый цвет;
- Персональные данные – бледно-оранжевый цвет;
- Сведения, связанные с профессиональной деятельностью – светло-коричневый цвет;
- Служебная тайна – светло-желтый цвет.

В матрице доступа в режиме **Доступ к атрибутам группы** возможен просмотр и настройка уровня доступа на продекларированные в справочной группе атрибуты. Чтобы перейти в данный режим необходимо установить курсор на одной из справочных групп в таблице, у которой присутствуют декларации атрибутов, и нажать кнопку ***Доступ к атрибутам группы*** либо выбрать в контекстном меню команду **Матрица доступа к атрибутам группы**. В открывшемся окне в заголовке будут указано наименование справочной группы, для которой показаны декларации атрибутов.

Для настройки прав доступа к продекларированным в справочной группе атрибутам необходимо выполнить двойной щелчок левой кнопкой мыши в ячейке на пересечении столбца с наименованием роли или пользователя и строки с наименованием атрибута. В открывшемся списке выбрать нужный вариант доступа (Рисунок 423).

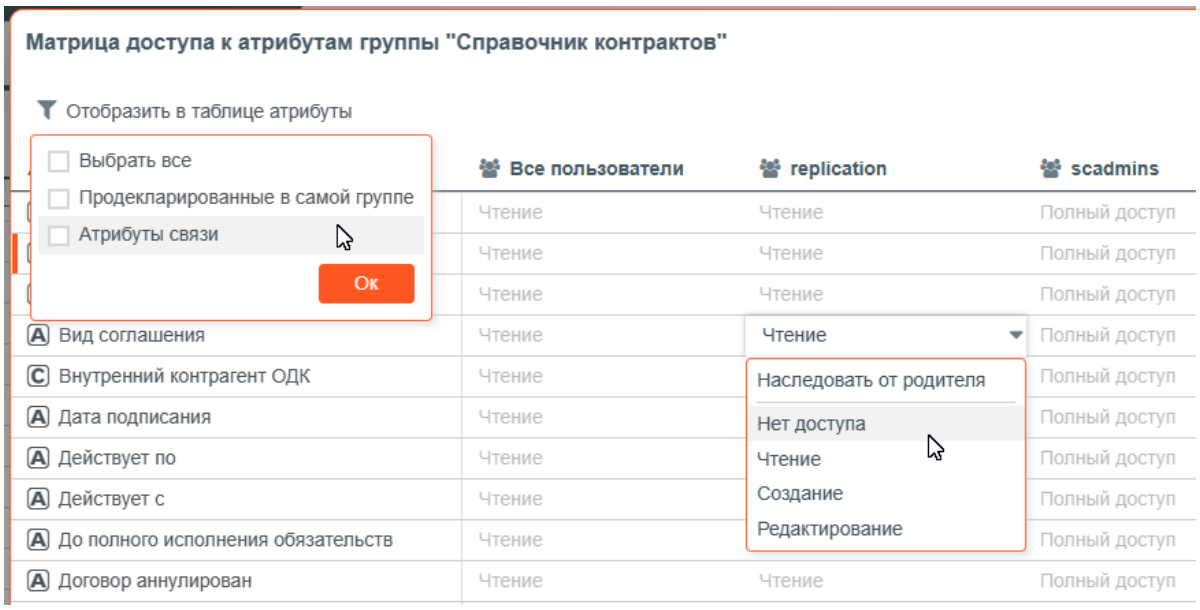


Рисунок 423 - Представление матрицы доступа в разрезе атрибутов справочной группы

В режиме **Доступ к атрибутам группы** доступна фильтрация таблицы по кнопке ***Отобразить в таблице атрибуты***:

- **Выбрать все** – маркер включает все фильтры одновременно;
- **Продекларированные в самой группе** – будут отфильтрованы только те атрибуты, которые продекларированы непосредственно в выбранной справочной группе (без унаследованных);
- **Атрибуты связи** – будут показаны только атрибуты связи, продекларированные в выбранной группе.

При включенном фильтре рядом с кнопкой ***Отобразить в таблице атрибуты*** в скобках показано сколько фильтров включено и сколько фильтров всего имеется. Для отключения фильтрации можно снять маркеры напротив соответствующих фильтров либо нажать на кнопку ***Фильтры включены***, рядом с ячейкой быстрого поиска над таблицей справа.

Для контроля использования информации из справочных групп предназначен режим **Применяемость группы** (Рисунок 424), в котором возможен просмотр применяемости группы и ее атрибутов в: вычисляемых атрибутах, продекларированных атрибутах связи, шаблонах заявки, обменных классах, маршрутах согласования.

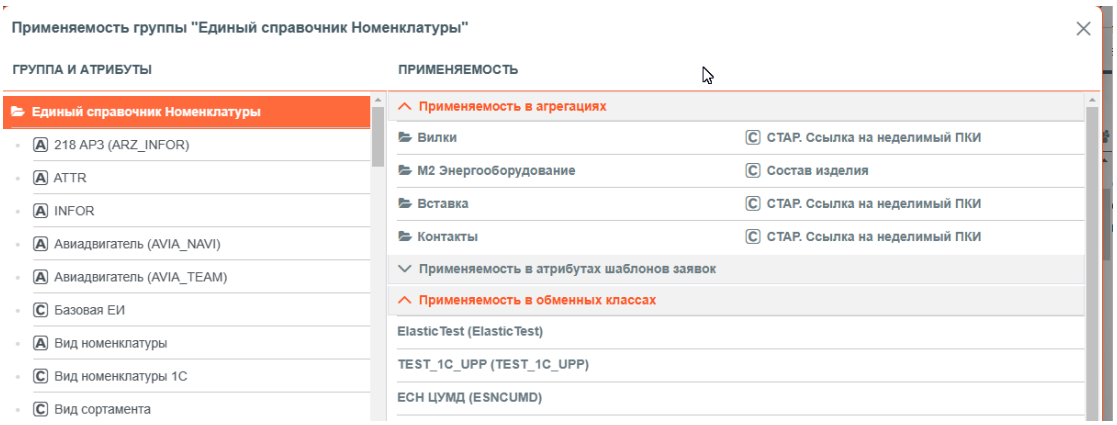


Рисунок 424 - Матрица доступа в разрезе применяемости справочных групп и продекларированных в них атрибутов

В разделе **Применяемость в агрегациях** в матрице доступа в разрезе применяемости справочных групп показано в каких атрибутах связи каких справочных групп используется рассматриваемая справочная группа. В левом столбце дано наименование справочной группы, а в правом - наименование атрибута связи. Оба наименования являются ссылками, по которым можно перейти в соответствующую справочную группу.

В разделе **Применяемость в атрибутах шаблонов заявок** в матрице доступа в разрезе применяемости справочных групп показано в каких шаблонах и атрибутах заявок используется ссылка на текущую справочную группу. В левом столбце дано наименование шаблона заявки, а в правом - наименование атрибута заявки. Оба наименования являются ссылками, по которым можно перейти в соответствующий шаблон.

В разделе **Применяемость в обменных классах** в матрице доступа в разрезе применяемости справочных групп показано в каких обменных классах участвует данная справочная группа. Наименование обменного класса является ссылкой, по которой можно перейти в соответствующий обменный класс.

Также, для справочной группы, доступен просмотр журнала работы пользователей с данной группой. Для доступа используется кнопка ***Журнал группы***, расположенная в верхней части окна матрицы доступа.

3.28.8 Статистика входа

Команда ***Статистика входа***, вызываемая кнопкой ***Действия*** над карточкой пользователя, предназначена для доступа к журналу (Рисунок 425), содержащему статистические данные по количеству входов в *Систему*, выполненных пользователями.

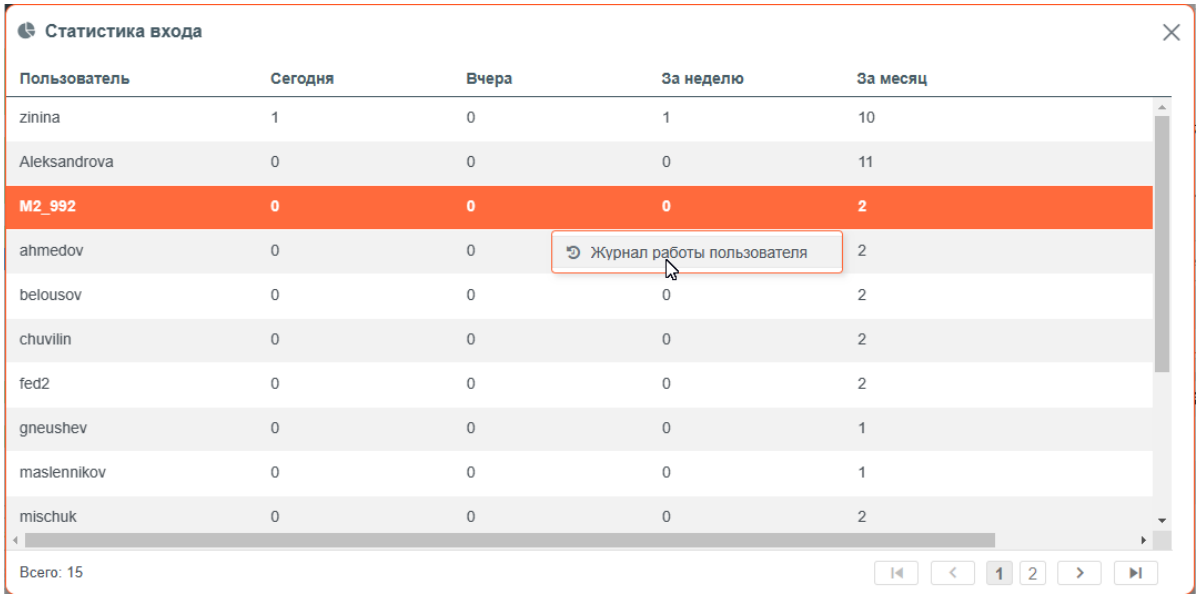


Рисунок 425 - Статистика входа пользователей

Стоя на любой записи в журнале можно вызвать контекстное меню, позволяющее открыть журнал работы конкретного пользователя .

3.28.9 Оценки приложения пользователями

Команда **Оценки** запускает процедуру скачивания журнала (файл в формате Excel), содержащего оценки, данные работе *Системы*, с использованием команды О**цените приложение** в профиле пользователя. Наличие команды **Оцените приложение** зависит от настроек *Системы*.

3.28.10 Атрибуты, извлекаемые через LDAP

Команда **Атрибуты, извлекаемые через LDAP** вызывает окно, в котором можно задать список атрибутов, дополнительно извлекаемых в

карточку пользователя из *Active Directory*. Команда работает в случае настройки доступа к *Системе* пользователей *Windows*.

По умолчанию из *Active Directory* читаются следующие атрибуты:

- givenName - Имя;
- sn - Фамилия;
- mail - электронная почта;
- memberOf - список групп в которые входит пользователь, чтобы включить его в соответствующие группы Semantic.

Для того чтобы увидеть атрибуты пользователя, нужно включить флаг Advanced features.

Чтобы посмотреть атрибуты пользователя, выбрать в Active Directory пользователя, открыть свойства и на вкладке Atribute editor посмотреть доступные атрибуты.

3.28.11 Связанная группа

Команда **Связанные группы** позволяет перейти в справочную группу, в которой находится позиция с записью о текущем пользователе.

3.29 Active Directory. Авторизация пользователей Windows

3.29.1 Настройка авторизации пользователей через Active Directory на сервере

- Включить в домен хост с сервером приложений Semantic MDM.
- Обеспечить доступ к контроллеру домена по имени с хоста с сервером приложений *Semantic MDM*.
- В представлении **Настройки**, включить авторизацию через LDAP.
- Для включения поддержки Kerberos-аутентификации в сервере приложений *Semantic MDM*, требуется сформировать cli-файл через специальный диалог в веб-приложении. После формирования файла с расширением cli, требуется выполнить его на хосте с сервером

приложений *Semantic MDM* с помощью команды:

- /opt/wildfly/bin/jboss-cli.sh --file=**<путь к файлу cli>**
- На контроллере домена выполнить команду:
- setspn -A HTTP/**<имя домена СП> <имя участника-службы>**
  - **<доменное имя СП> –** доменное имя хоста с сервером приложений Semantic MDM, например, app.sdi.ru;
  - **<имя участника-службы>** - имя пользователя домена, по которому клиент однозначно определяет экземпляр службы.
- На контроллере домена выполнить команду:

  ktpass -princ HTTP/**<имя домена СП>@<имя домена>** -mapuser **<имя участника-службы>** -pass **<пароль участника-службы>**-ptype KRB5_NT_PRINCIPAL -out **<путь к файлу>**
  - **<имя домена>** – имя домена, например, sdi.ru;
  - **<доменное имя СП>** – доменное имя хоста с сервером приложений Semantic MDM, например, app.sdi.ru;
  - **<имя участника-службы>** – имя пользователя домена, по которому клиент однозначно определяет экземпляр службы**;**
  - **<пароль участника-службы>** – пароль;
  - **<путь к файлу>** – полный путь к файлу, который будет сформирован.
- Сформированный на контроллере домена файл требуется скопировать на хост с сервером приложений *Semantic MDM*, в директорию **/opt/wildfly/standalone/configuration**, после чего перезапустить сервер-приложений Wildfly командой: sudo service wildfly restart

3.29.2 Настройка авторизации через Active Directory на клиенте

Данный процесс описан для браузеров (например, *Google Chrome*), использующих настройки *Internet Explorer*.

- Открыть свойства браузера;

- На вкладке **Дополнительно** (Рисунок 426) установить опцию **Разрешить встроенную проверку подлинности Windows**;

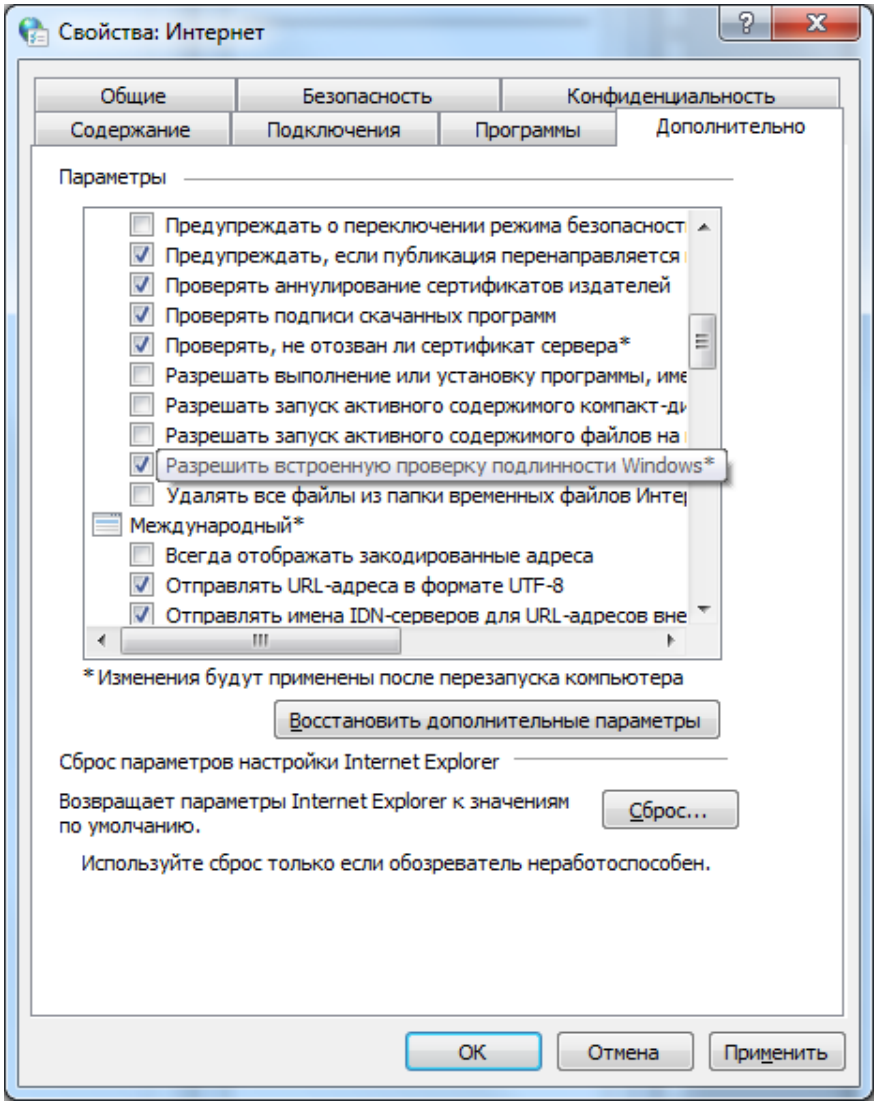


Рисунок 426 - Дополнительные настройки браузера

- На вкладке **Конфиденциальность** по кнопке ***Сайты*** открыть диалог конфиденциальности (Рисунок 427) веб-сайтов и добавить доменное имя хоста с сервером приложений *Semantic MDM*;
- Открыть веб-приложение *Semantic MDM* через браузер и проверить вход в систему от имени текущего доменного пользователя.

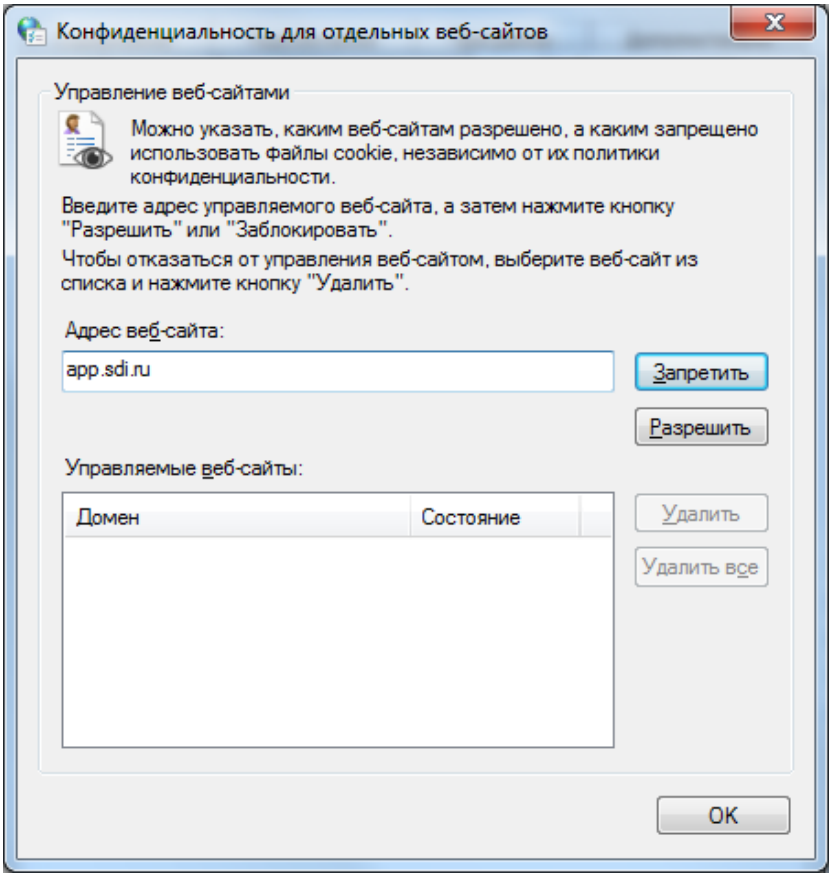


Рисунок 427 - Настройка конфиденциальности подключения

Для настройки работы с браузером *Internet Explorer* (*Microsoft*) следует:

- Перейти на вкладку **Безопасность** (Рисунок 428);
- Выбрать **Местная интрасеть -> Сайты**;

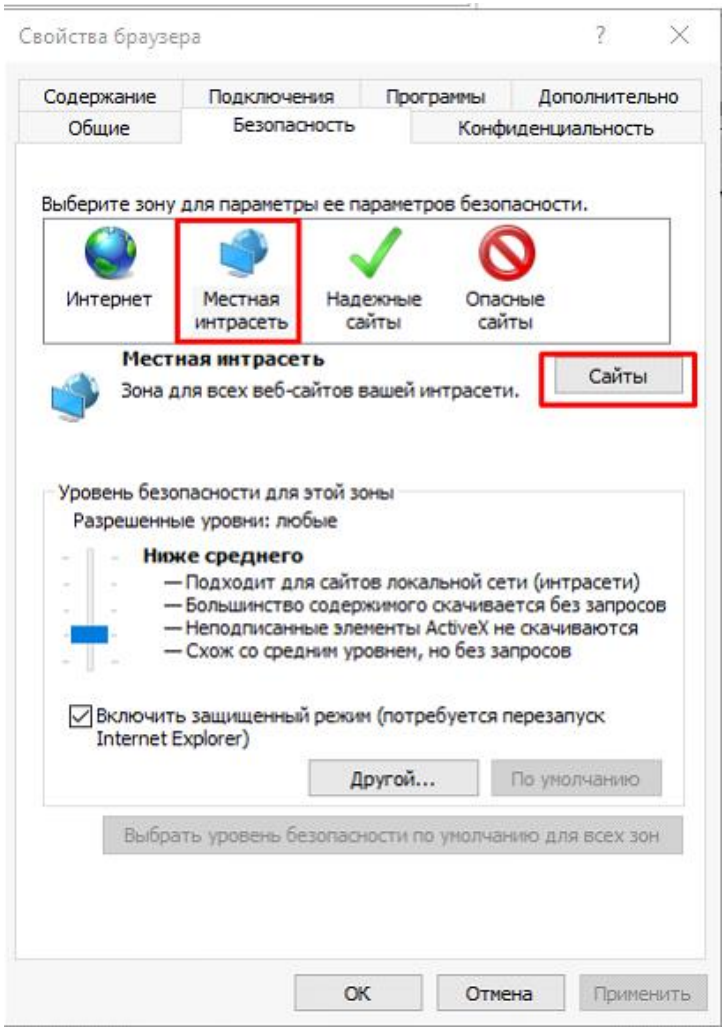


Рисунок 428 – Вкладка Безопасность

- Добавить в зону следующий узел (Рисунок 429) (Например: http://*mdm.test.ru)

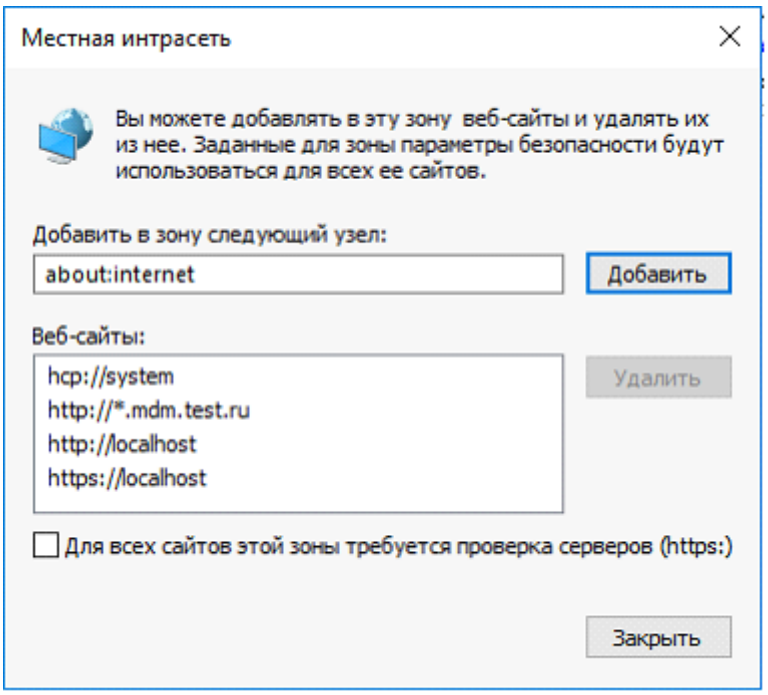


Рисунок 429 – Местная интрасеть

3.30 Импорт и репликация

Подсистема **Импорт и репликация** предназначена для интеграции с внешними информационными системами. В подсистеме можно зарегистрировать перечень внешних информационных систем (ИС), с которыми выполняется обмен данными, а также настроить обменные классы[21], с возможностью:

- настроить структуру обменного класса вручную или использовать структуру имеющейся модели данных, как структуру обменного класса. Передача атрибутов в структуру обменного класса, выполняется автоматически, так как настройка структуры обменного класса выполняется на основе атрибутов имеющейся модели данных;
- разработать с помощью скрипта Groovy различные динамические преобразования данных и структуры объекта обменного класса.

[21] *Обменный класс - класс, описывающий настройки репликации, то есть соответствие атрибутов, направления обмена информацией, группы справочников, данные которых обрабатываются в соответствии с заданными настройками.*

Преобразования можно выполнять как на импорт, так и экспорт;

- разработать с помощью скрипта Groovy функции проверки и отклонения данных, пришедших из внешней ИС;
- настроить список систем, с которыми через данный обменный класс будет выполняться обмен данными.

Можно отслеживать состояние сообщений на импорт и экспорт и выполнять повторную выгрузку всех или выбранных сообщений.

На вкладке **Внешние системы** располагается таблица с зарегистрированными внешними ИС. Установив курсор на строке с наименованием ИС в карточке можно ознакомится с настроенными параметрами подключения. Таблица поддерживает возможность настройки видимых полей, сортировку по столбцам, а также возможность фасетного поиска. Фасетный поиск разрешён для столбцов – **Учётная запись**, **Способ репликации**, **Формировать сообщения**.

В системе *Semantic MDM* возможно использование двух вариантов интеграции с внешними информационными системами. Первый – это интеграция через через API, который наиболее востребован при интеграции с различными САПР. Основным способом интеграции с системами управления производством, является второй способ – интеграция через репликацию. Подсистема **Импорт и репликация** позволяет настроить выгрузку и загрузку данных для организации интеграции.

3.30.1 Подключение внешней информационной системы

Для подключения любой внешней информационной системы необходимо зарегистрировать её. Для этого нужно перейти в представление **Импорт и репликация** и, находясь на вкладке **Внешние системы**, нажать кнопку ***Добавить***, которая расположена справа над карточкой внешней ИС либо использовать команду **Добавить систему** из контекстного меню под правой кнопкой мыши.

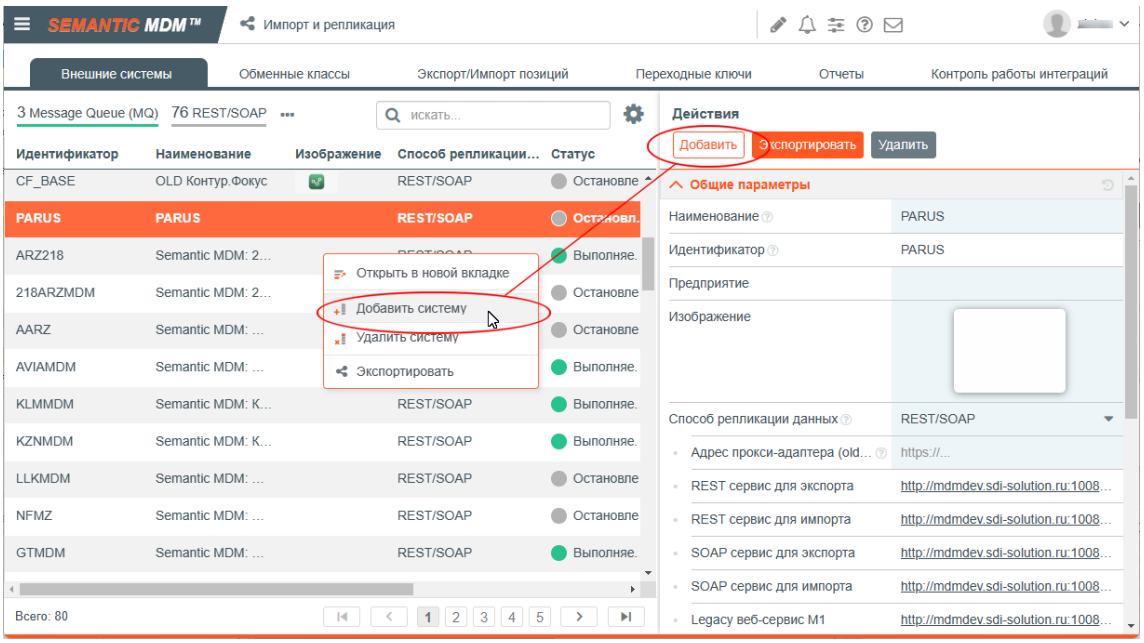


Рисунок 430 - Добавление новой внешней системы

В появившейся пустой карточке внешней ИС необходимо заполнить следующие данные (общие параметры) о ней:

- **Наименование** – название внешней информационной системы;
- **Идентификатор** – условное или сокращённое обозначение внешней ИС. Длина строки 255 символов. Запись регистрозависима. Может содержат любые буквы, цифры и символы Unicode;
- **Предприятие** – значение может быть введено вручную или через кнопку выбора. В табличном представлении для данного столбца разрешена фасетная фильтрация;
- **Изображение** – область для добавления рисунка, обозначающего систему. Не является обязательным;
- **Способ репликации данных** – четыре варианта взаимодействия *Semantic MDM* с внешней ИС (Рисунок 431), которые выбираются из списка – Message Queue (MQ), REST/SOAP (подробнее см. п. 3.30.2) или *Репликация в дочерний Semantic MDM, Репликация в центральный Semantic MDM* ;

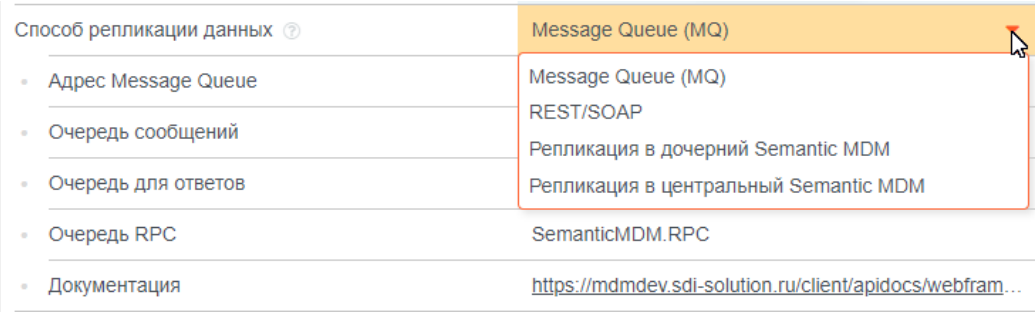


Рисунок 431 - Выбор способа репликации данных

- **Интервал проверки PING-сообщений** – интервал времени в минутах для контроля статуса работы адаптера к внешней информационной системе. Контроль осуществляется путём отсылки интеграционным адаптером сообщения «ping» в подсистему импорта репликации через заданный интервал времени, а в подсистеме импорта и репликации организована проверка получения сообщения «ping» через указанный интервал времени. При отсутствии «ping» со стороны интеграционного адаптера в течение заданного времени *Система* оповестит администратора через браузер и электронную почту (e-mail) о наличии проблем в интеграции;
- **Формировать сообщения на экспорт:**
  - **для каждой позиции** – для каждой позиции вне зависимости от числа связанных переходных ключей будет формироваться одно сообщение на экспорт;
  - **для каждого переходного ключа** – если с одной эталонной позицией в Системе связано несколько переходных ключей одной внешней ИС, то для каждого переходного ключа будет сформировано сообщение на экспорт с указание оригинала;
- **Адрес сервиса внешней применяемости** – указывается интернет-адрес сервиса, обеспечивающий обращение к внешней ИС из справочников;
- **Последняя активность** – дата и время последнего выполнения репликации данных.

В разделе **Преобразование строковых значений** (Рисунок 432) задаются правила маскирования символов в строковых и текстовых атрибутах при выполнении экспорта во внешнюю систему. Для каждой внешней системы можно настроить таблицу замен, где в столбце «*Что заменяем*» задается регулярное выражение, а в столбце «*На что заменяем*» задается строковое значение, на которое будет выполнена замена найденных значений. Например, регулярное выражение «[^\w\s]+?» уберет из записи символы «~ # % » и т.п. Если столбец «*На что заменяем*» будет не заполнен, то вместо символов в записи будут пробелы. В приведенном регулярном выражении используются следующие значения: \w [0-9a-z_] любая цифра, буква или знак подчеркивания; \s [\t\v\r\n\f] пробельный символ; ^ - это отрицание того, что указано после.

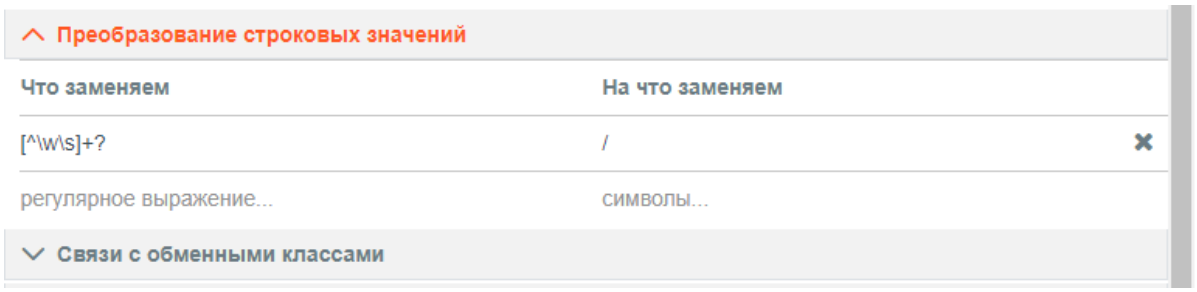


Рисунок 432 - Настройка преобразования строковых значений

Если один из обязательных атрибутов внешней ИС не настроен, то в таблице и соответствующей ячейке карточки появляется предупреждающий знак – .

В разделе **Права доступа** настраивается уровень доступа к функциям внешней системы В подсистеме **Импорт и репликация** выделяются следующие уровни доступа к функциям:

- *Нет доступа* – пользователи не могут видеть информацию по данной системе;
- *Чтение* – пользователи могут видеть информацию о ИС, а также информацию о переходных ключах, сообщениях экспорта и импорта;

- *Экспорт* – пользователи имеют право *Чтение* и могут выполнять экспорт позиций в данную систему;
- *Импорт* – пользователи имеют право *Экспорт* и могут выполнять повторный импорт сообщения импорта, полученного из внешней ИС.

На этом настройка подключения внешней ИС завершена и необходимо установить связь с обменными классами, то есть указать из каких справочных групп будут экспортироваться данные.

3.30.2 Способ передачи данных

Доступ к данным, обмен которыми настроен в Semantic MDM возможен через несколько каналов взаимодействия (транспорт – формат передачи данных):

- SemanticMQ – Json (предпочтительный)
- REST – Json
- сервис SOAP Document – Literal
- сервис SOAP RPC – Literal

Способ обмена данными указывается в параметрах системы выбором одного из двух вариантов – Message Queue(MQ) или REST/SOAP (Рисунок 433).

В настройках для способа Message Queue (MQ) доступны следующие опции:

- **Адрес Message Queue** – интернет-адрес сервиса подключения;
- **Очередь сообщений** – создаётся автоматически из сообщений от *Semantic MDM* (одно сообщение – одна позиция);
- **Очередь для ответов** – в данной очереди внешняя ИС уведомляет о результате обработки сообщений (одно сообщение – одна позиция);
- **Очередь RPC** – используется внешней ИС для сообщения Semantic MDM, что система с идентификатором <systemId> функционирует и готова к приёму сообщений. Функция вызывается один раз в минуту.

А также для приёма файлов большого размера (получения тела файла с известным хэшем);

- **Документация** – интернет-адрес, где расположена документация по работе с сервисом импорта и репликации.

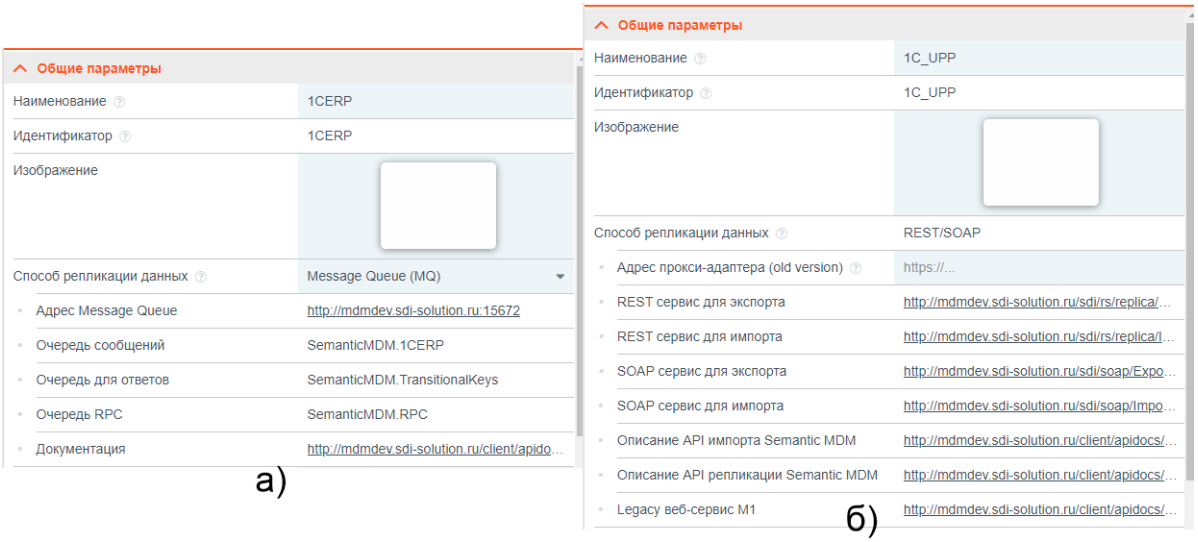


Рисунок 433 - Настройки способа репликации данных: а) Message Oueue (MQ); б) REST/SOAP

Для варианта REST/SOAP настраиваются:

- **Адрес прокси-адаптера (old version)** – интернет-адрес, где размещён прокси-адаптер, обеспечивающий работу с репликационными таблицами предыдущей версии *Semantic MDM*;
- **REST сервис для экспорта** – интернет-адрес, где размещён соответствующий сервис;
- **REST сервис для импорта** – интернет-адрес, где размещён соответствующий сервис;
- **SOAP сервис для экспорта** – интернет-адрес, где размещён соответствующий сервис;
- **SOAP сервис для импорта** – интернет-адрес, где размещён соответствующий сервис;
- **Описание API импорта Semantic MDM** – интернет-адрес, где расположена документация по работе с сервисом импорта;

- **Описание API репликации Semantic MDM** – интернет-адрес, где расположена документация по работе с сервисом репликации;
- **Legacy веб-сервис M1** – интернет-адрес, где размещён SOAP веб-сервис, обеспечивающий работу *Модуля импорта и репликации* предыдущей версии *Semantic MDM*.

Обмен данными через SemanticMQ предпочтителен, поскольку выполняется в реальном времени, сразу после того, как изменённые данные были опубликованы и подготовлены для экспорта. При использовании этого варианта выполнения репликации следует учитывать, что система – подписчик обязан создать очередь с нужными параметрами и правильным наименованием очереди, которое указано для внешней системы. Задачей Semantic MDM является отправка сообщения с правильным ключом маршрутизации, по которому сообщение попадет в нужную очередь сообщений. Если очередь на момент отправки сообщения не будет найдена, то сообщение будет отправлено в системную очередь, из которой оно будет удалено в течении 15 минут.

Если пользователь сам будет решать, когда инициировать передачу данных, то следует использовать веб-сервис REST / SOAP.

3.30.3 Приоритизация экспорта

В *Semantic MDM* в подсистеме **Импорт и репликация** возможна настройка приоритизации экспорта . В *Системе* выделено три приоритета:

- **Высокий приоритет** – присваивается сообщениям экспорта, сформированным на основе пакета изменений, содержащего до 200 изменений в пакете изменений;
- **Средний приоритет** – присваивается сообщениям экспорта, сформированным на основе пакета изменений, содержащего от 200 до 1000 изменений в пакете изменений;

- **Низкий приоритет** – присваивается сообщениям экспорта, сформированным на основе пакета изменений, содержащего более 1000 изменений в пакете изменений.

При формировании сообщения экспорта *Система* присваивает ему приоритет исходя из количества изменений выполненных в пакете изменений. *Система* позволяет настроить пороги для каждого из приоритетов, а также для сформированных сообщений экспорта выполнить изменение приоритета вручную. Если для внешней системы, выбран способ репликации Message queue (MQ), то для каждого из приоритетов в карточке внешней системы можно настроить ключи маршрутизации. Если для внешней системы, выбран способ репликации REST/SOAP, то для получения сообщения экспорта с требуемым приоритетом необходимо использовать метод getMessagesAllowedToDispatchByPriority.

Для настройки приоритетов экспорта необходимо на главной панели нажать кнопку ***Настройка приоритетов экспорта*** (Рисунок 434) и, в открывшемся диалоге, при помощи перемещения «ручек», настроить количество разрешенных для каждого приоритета изменений в пакете изменений и подтвердить выполненные изменения кнопкой ***Ок***.

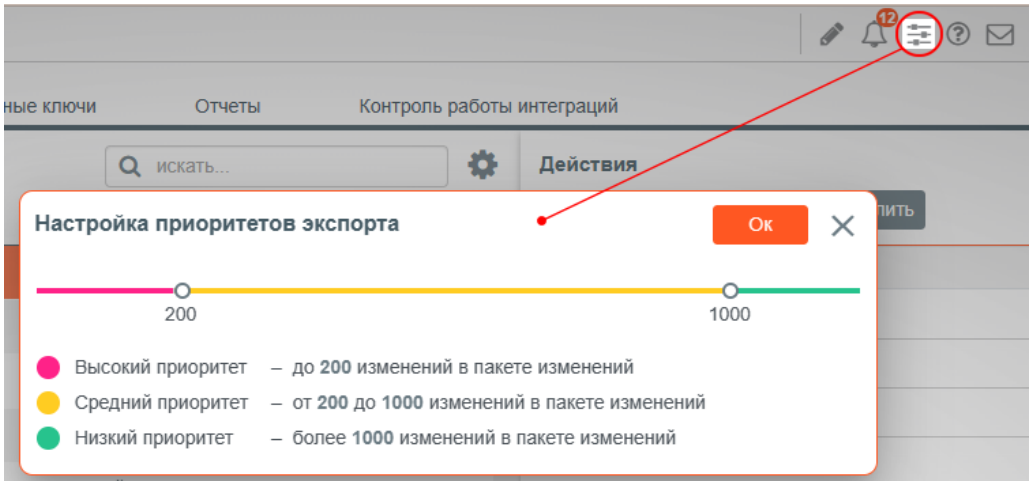


Рисунок 434 - Настройка приоритетов экспорта сообщений

3.30.4 Настройка обменных классов

Вкладка **Обменные классы** (Рисунок 435) содержит список справочных групп, позиции которых будут передаваться во внешние или из внешних ИС. Один и тот же обменный класс может участвовать в процессе репликации с несколькими ИС. В таблице обменных классов разрешена сортировка по столбцам, а также фасетный поиск везде, кроме колонок **Наименование** и **Идентификатор**. Для фильтрации по статусу может быть использована строка с интерактивными статусами.

Для добавления нового обменного класса необходимо:

- перейти на вкладку **Обменные классы** представления **Импорт и репликация**;
- нажать кнопку ***Добавить***, расположенную справа или вызвать аналогичную команду из контекстного меню;

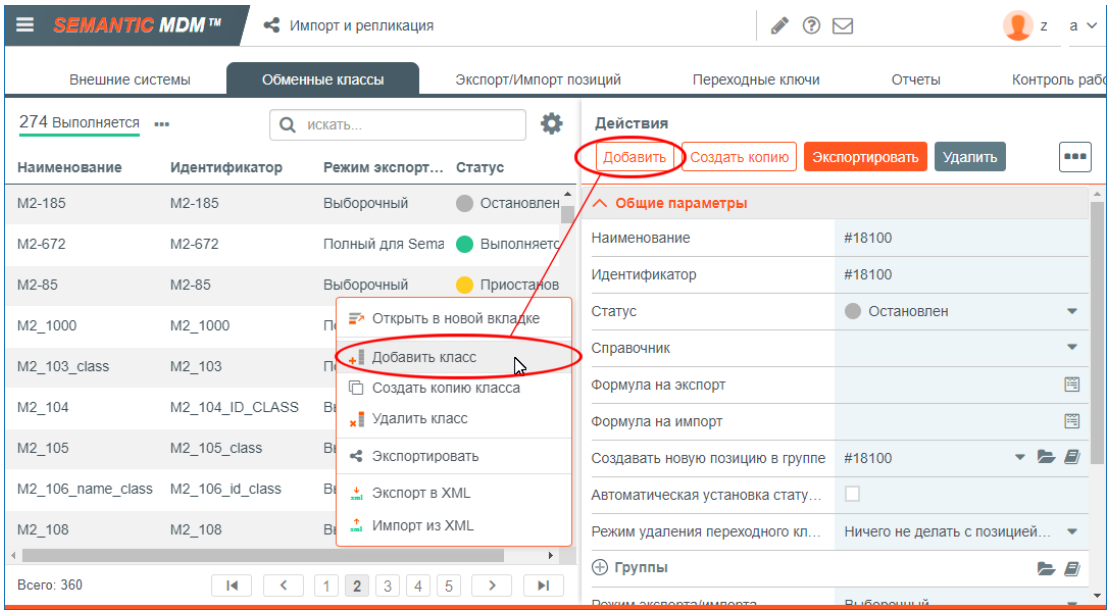


Рисунок 435 - Вид вкладки обменных классов

- перейти в карточку обменного класса и заполнить **Общие параметры**:
  - **Наименование** – название обменного класса;
  - **Идентификатор** – условное или сокращённое обозначение обменного класса. Длина строки 255 символов. Запись

регистрозависима. Может содержат любые буквы, цифры и символы Unicode;

- **Статус экспорта** – принимает три возможных значения, выбирающихся из выпадающего списка. *Остановлен* – изменения, выполняемые в Semantic MDM, игнорируются. *Приостановлен* – изменения, выполняемые в *Semantic MDM*, накапливаются и будут обработаны, как только будет включён статус *Выполняется*. *Выполняется* – обработка изменений выполняется в онлайн режиме. Статус может быть определён для каждого класса и / или для внешней ИС, связанной с ним;
- **Справочник** – выбирается справочник или классификатор из списка имеющихся в конфигураторе справочников и классификаторов;
- **Формула на экспорт** – содержит формулу, представляющую собой скрипт Groovy, написанный в специальном редакторе. Формула определяет алгоритм экспорта данного класса во внешнюю ИС;

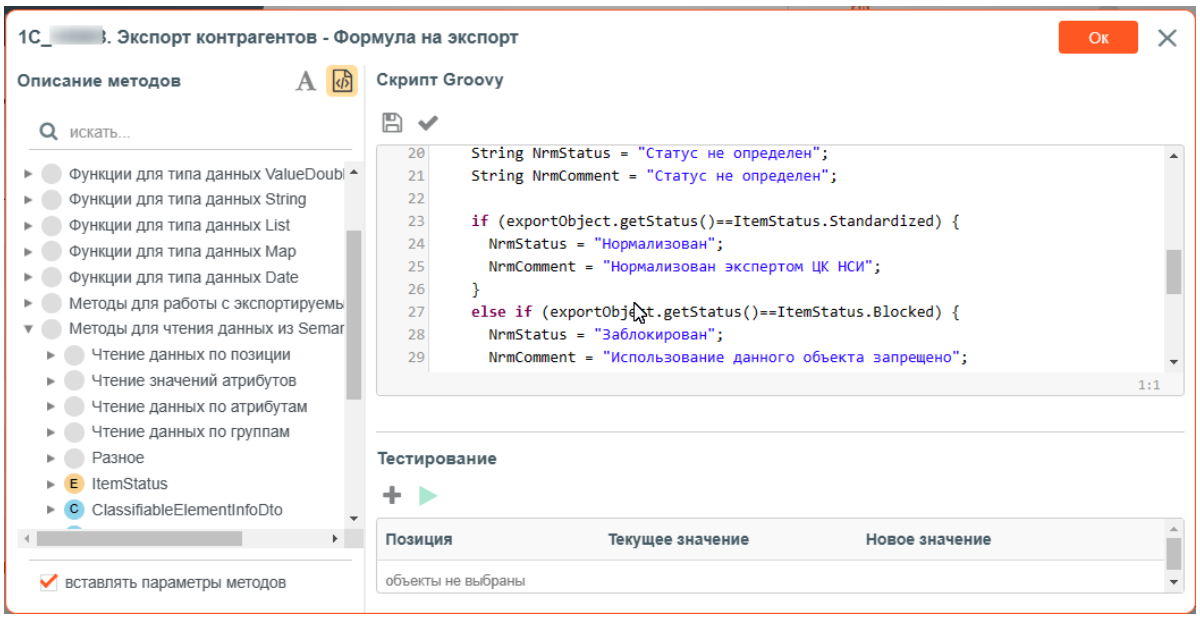


Рисунок 436 - Формула на экспорт позиции в виде скрипта Groovy

- **Формула на импорт** – содержит формулу, представляющую собой скрипт Groovy, написанный в специальном редакторе. Формула определяет порядок импорта данного класса в Semantic MDM;
- **Создавать новую позицию в группе** – из выпадающего списка или в дереве справочников выбирается справочник или ветка справочника, куда будут записываться новые позиции при импорте;
- **Автоматическая установка статуса «Нормализован»** – установка данного маркера позволяет присваивать всем импортируемым позициям данного класса статус *Нормализован*а в справочнике *Semantic MDM*. Если маркер не установлен все записи получают статус *Не нормализована*;
- **Режим удаления переходного ключа** – определяет порядок удаления позиции: *Удалить позицию, если переходные ключи отсутствуют* – позицию разрешено удалить, если у неё нет ни одного переходного ключа с внешними ИС; *Удалять позицию, если у позиции установлен статус «Не нормализована» и отсутствуют переходные ключи; Удалять позицию, если отсутствуют переходные ключи, в противном случае установить статус «Заблокировано»; Ничего не делать с позицией при удалении переходного ключа;*
- **Группы** – список справочных групп, входящих в данный обменный класс, которые выбираются в окне **Группы** (Рисунок 437), вызываемом кнопкой ***Выбрать***, расположенной в правой части ячейки;

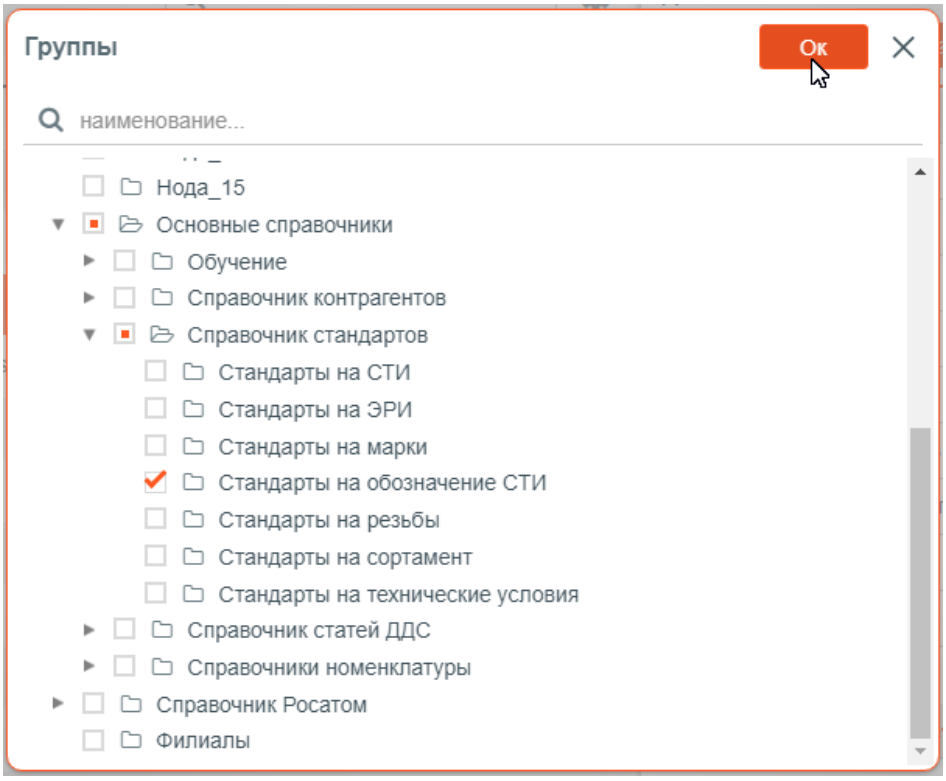


Рисунок 437 - Выбор справочных групп для обменного класса

- **Режим экспорта/импорта** – настройка передаваемых атрибутов. Существует три режима. **Полный** – будут реплицированы все атрибуты, описывающие позиции группы, кроме атрибутов, указанных в списке **Неэкспортируемые атрибуты. Выборочный** – будут реплицированы только те атрибуты, которые указаны в списке **Сопоставление атрибутов**. Полный Semantic - Подробнее о режимах в п. 3.30.7;
- **Сопоставление атрибутов** – сопоставление атрибутов обменного класса и *Semantic MDM* и установление порядка их выгрузки;
- **Переходные ключи с системами** – выбор внешних ИС, для которых необходимо выполнят выгрузку переходных ключей в обменный класс;

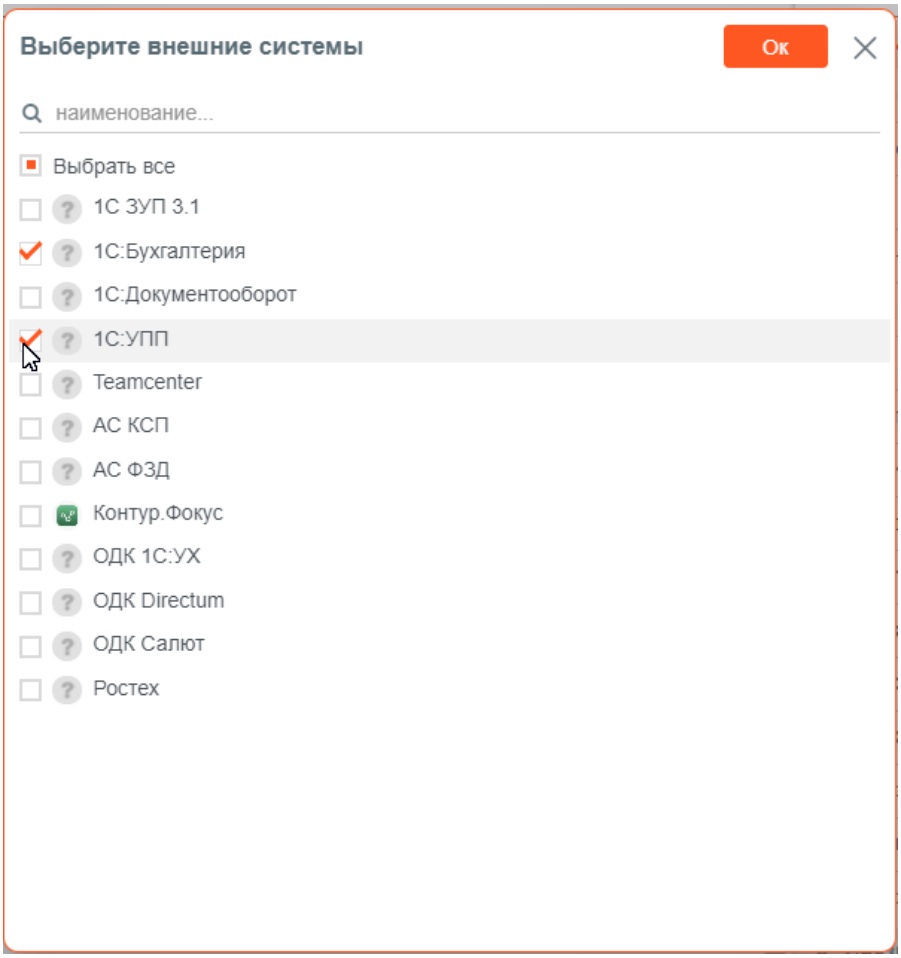


Рисунок 438 - Выбор внешних ИС, для которых будут выгружаться переходные ключи

- **Совместимости** - перечень типов совместимостей объектов НСИ (Рисунок 439), которые следует передавать во / из внешних ИС. Возможна выгрузка настроенных на справочных группах совместимостей по умолчанию (маркер - **настроенные в группах обменного класса**);

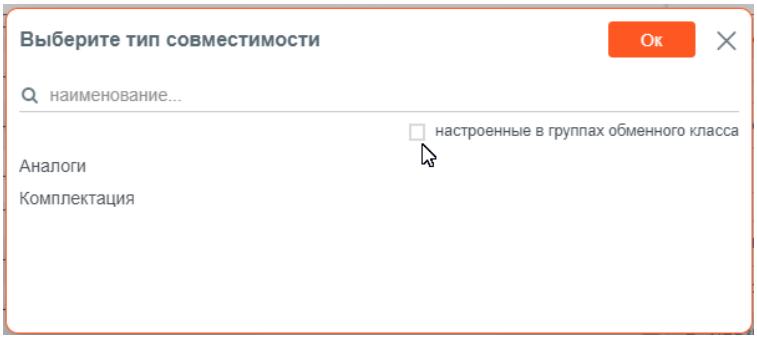


Рисунок 439 - Выбор типов совместимостей, которые будут выгружаться в обменный класс

- **Управление заявками** (Рисунок 440)**:**
  - **Создать заявку на «Нормализацию»**– опция, которая позволяет автоматически создавать заявки (см.п. 3.30.5) на нормализацию при получении (импорте) записей из внешней ИС;
  - **Создать заявку на «Изменение»** – опция, которая позволяет автоматически создавать заявки на изменение позиции при получении (импорте) записей из внешней ИС.
  - **Создать заявку на «Консолидацию»** – опция, которая позволяет автоматически создавать заявки на консолидацию позиций справочника с позициями, полученными (импортированными) из внешней ИС

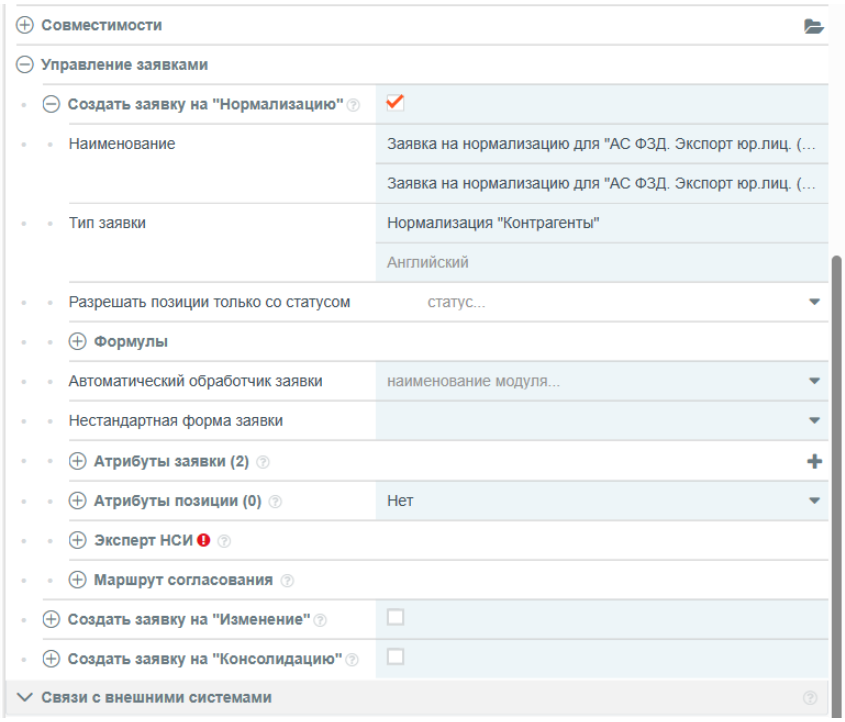


Рисунок 440 - Доступ к настройке заявок на «Нормализацию», «Изменение» и «Консолидацию»

Далее следует установить связь созданного обменного класса с требуемыми внешними ИС. Эта процедура может быть проведена как со стороны обменного класса, так и со стороны ИС на вкладке **Внешние системы**.

3.30.5 Заявки на импортируемые позиции

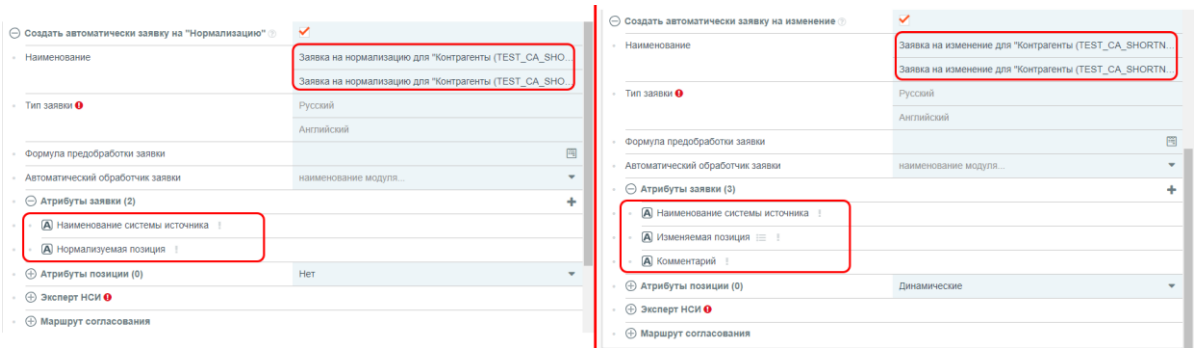
В Semantic MDM предусмотрено автоматическое создание заявок на нормализацию, изменение и консолидацию позиций из сообщений импорта от внешних информационных систем, с возможностью отклонения и возврата причины отклонения в исходную систему.

При обработке сообщения импорта из внешней ИС, если у позиции отсутствует переходной ключ с *Системой* и она не была найдена, при сопоставлении с имеющимися позициями, то будет создана новая позиция со статусом «Не нормализована». Для неё будет создана **заявка на нормализацию** с указанием в ней ссылки на созданную позицию. При отклонении заявки экспертом НСИ, будет проведено отклонение сообщения импорта и передача информации во внешнюю систему с причинами отклонения. Аналогично организована работа с **заявками на изменение**, при этом позиция НСИ уже должна существовать (поиск осуществляется по GUID и переходному ключу) со статусом «Нормализована». Через заявки на изменение можно также выполнять изменение композируемых позиций. Все исключаемые атрибуты редактируются автоматически, без заявки.

Заявка на консолидацию создается, если для импортируемой позиции отсутствует опубликованный ранее переходной ключ и *Система* выполнила сопоставления с имеющейся позицией через GUID или поиск дубликатов. В этом случае, будет создана новая позиция со статусом «Не нормализована», а также заявка на консолидацию, в которой импортированная позиция будет указана как дубликат, а найденный эталон будет указан как оригинал. По данной заявке пользователь может выполнить полноценную дедубликацию или сделать из созданной позиции новый эталон.

Данный функционал позволяет организовать точку первичного ввода и корректировки информации во внешней ИС, с последующим преобразованием этой информации в заявки *Semantic MDM*, их обработки экспертами НСИ и возвратом во внешнюю систему либо нормализованного / изменённого объекта, либо причины отклонения объекта.

Для получения заявок из сообщений импорта необходимо настроить соответствующую опцию в карточке обменного класса (см. п. 3.30.4) в категории **Управление заявками**. Настройка заявок ведется по алгоритму изложенному в разделе Шаблоны заявок. При установке опций **Создать заявку на «Нормализацию»**, **Создать заявку на «Изменение»** или **Создать заявку на «Консолидацию»**, часть настроек - наименование и атрибуты заявки (показаны на рисунке ниже), уже выполнены. При необходимости пользователь может внести дополнительные атрибуты заявки и указать атрибуты позиции (Рисунок 441).



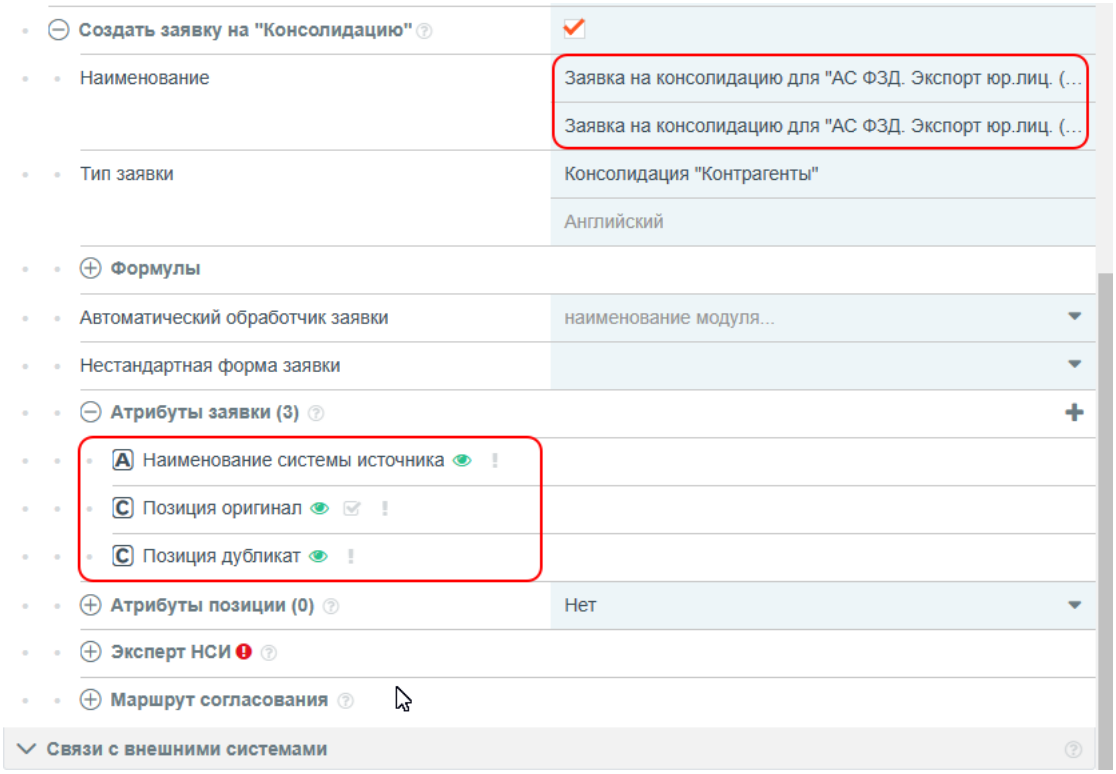


Рисунок 441 - Автоматические настройки заявок обменного класса

3.30.6 Копирование обменного класса. Работа через XML

Один и тот же обменный класс может использоваться с несколькими внешними ИС как для экспорта, так и для импорта данных. С целью уменьшения объёма работ по настройке обменного класса, он может быть скопирован с уже настроенного. Для этого достаточно установить курсор на копируемом классе и нажать кнопку ***Создать копию*** либо вызвать из контекстного меню команду ***Создать копию класса***. В окне **Создание копии обменного класса** (Рисунок 442) указать новое наименование и нажать кнопку ***Ок***. Копия будет создана и размещена в конце списка. При использовании скопированного класса в него могут быть внесены любые дополнительные изменения.

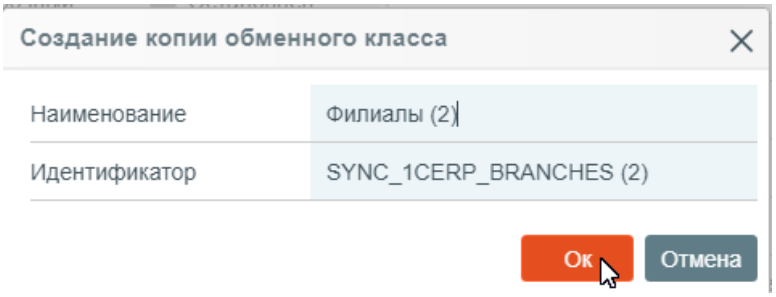


Рисунок 442 - Создание копии обменного класса

Кроме процедуры копирования обменный класс может быть передан или получен в / из другой базы данных. Например, при разработке интеграции с новой системой, может потребоваться создать новый обменный класс и настроить его. Эти работы можно выполнить в начале на тестовой БД, сделав все необходимые тесты для подтверждения работоспособности, после чего перенести через XML обменный класс на рабочую базу данных. При экспорте обменного класса в XML также выполняется экспорт и шаблонов, определенных для заявок на нормализацию, изменение, консолидацию.

Для этой процедуры необходимо установить курсор на нужный обменный класс и вызвать из дополнительного или контекстного меню соответствующую команду. Никаких дополнительных действий при экспорте / импорте обменного класса в XML не требуется.

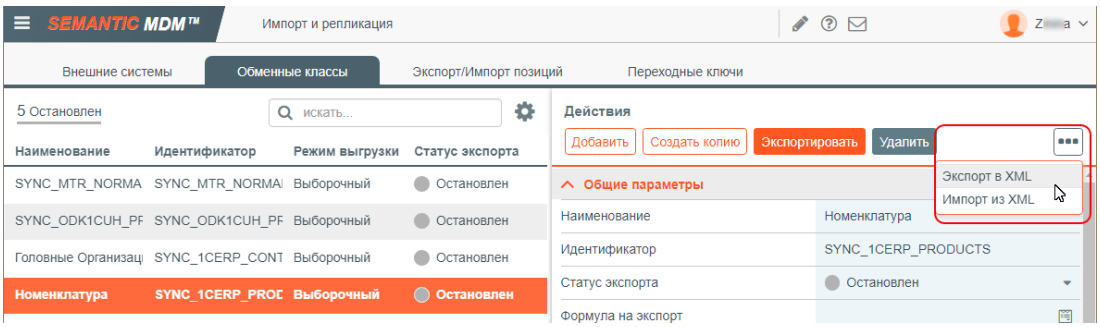


Рисунок 443 - Вызов команд **Экспорт в XML / Импорт в XML**

Помимо атрибутов обменного класса в выгрузку также попадают стереотипы совместимости, если они настроены для текущего класса.

3.30.7 Режим экспорта/импорта

Режим экспорта/импорта отвечает за количество и условия выгрузки атрибутов позиций, принадлежащих выбранным справочным группам.

Полный – будут реплицированы все атрибуты, имеющие значения и описывающие позиции группы, кроме атрибутов, указанных в списке **Неэкспортируемые атрибуты** (Рисунок 444). Для добавления неэкспортируемых атрибутов следует нажать кнопку ***Выбрать атрибуты*** и, в открывшемся списке **Атрибуты**, щелчком левой кнопки мыши на названии, выбрать атрибут (Рисунок 445).

Полный для Semantic - аналог режима **Полный**, применяемый только для репликации данных из одной *Semantic MDM* в другую (подробнее см. пункт 3.30.19).

Все атрибуты, оказавшиеся в списке неэкспортируемых, не будут реплицироваться во внешние ИС.

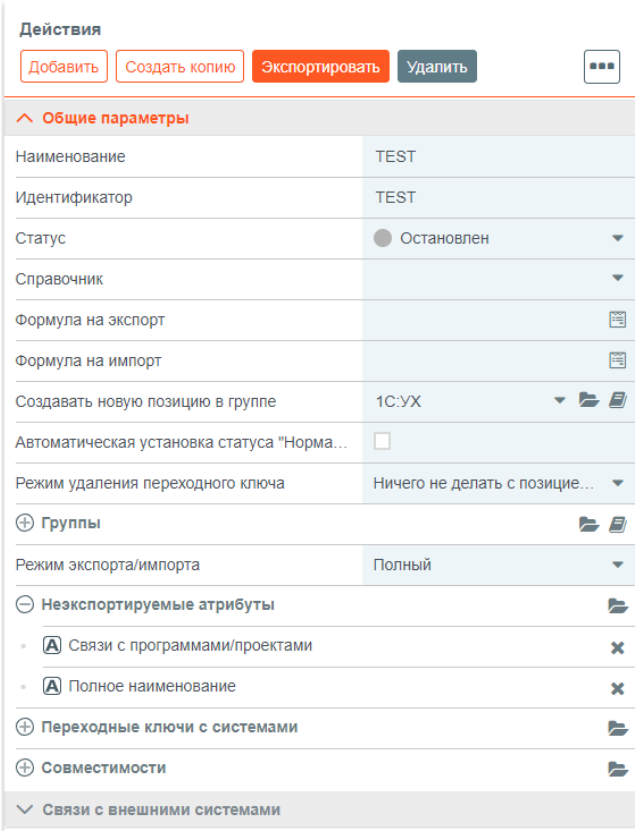


Рисунок 444 - Полный режим выгрузки позиций

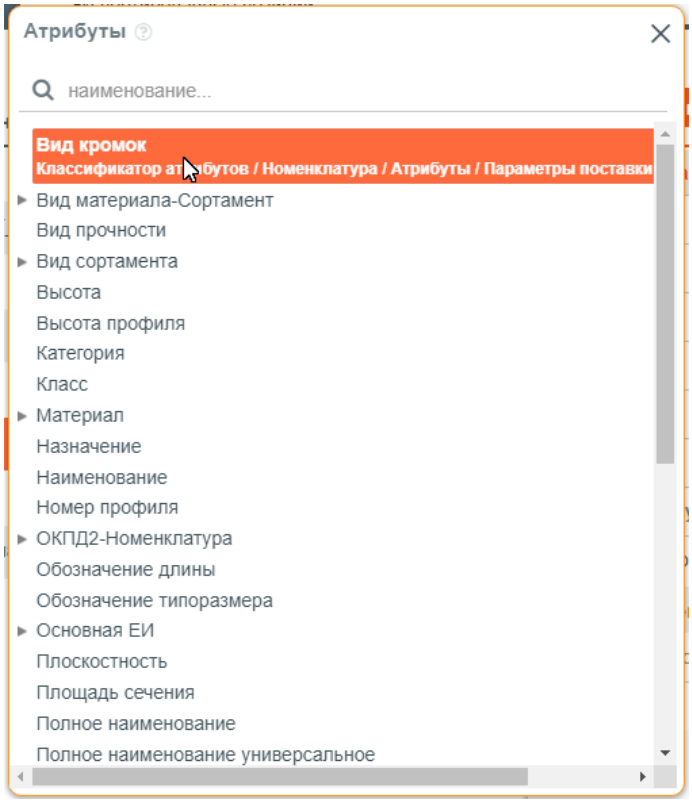


Рисунок 445 - Выбор атрибутов

Выборочный – режим, при котором будут реплицированы только те атрибуты, которые указаны в списке **Сопоставление атрибутов** (Рисунок 446). Атрибуты в список добавляются кнопками ***Атрибуты для сопоставления*** или ***Добавить атрибут***.

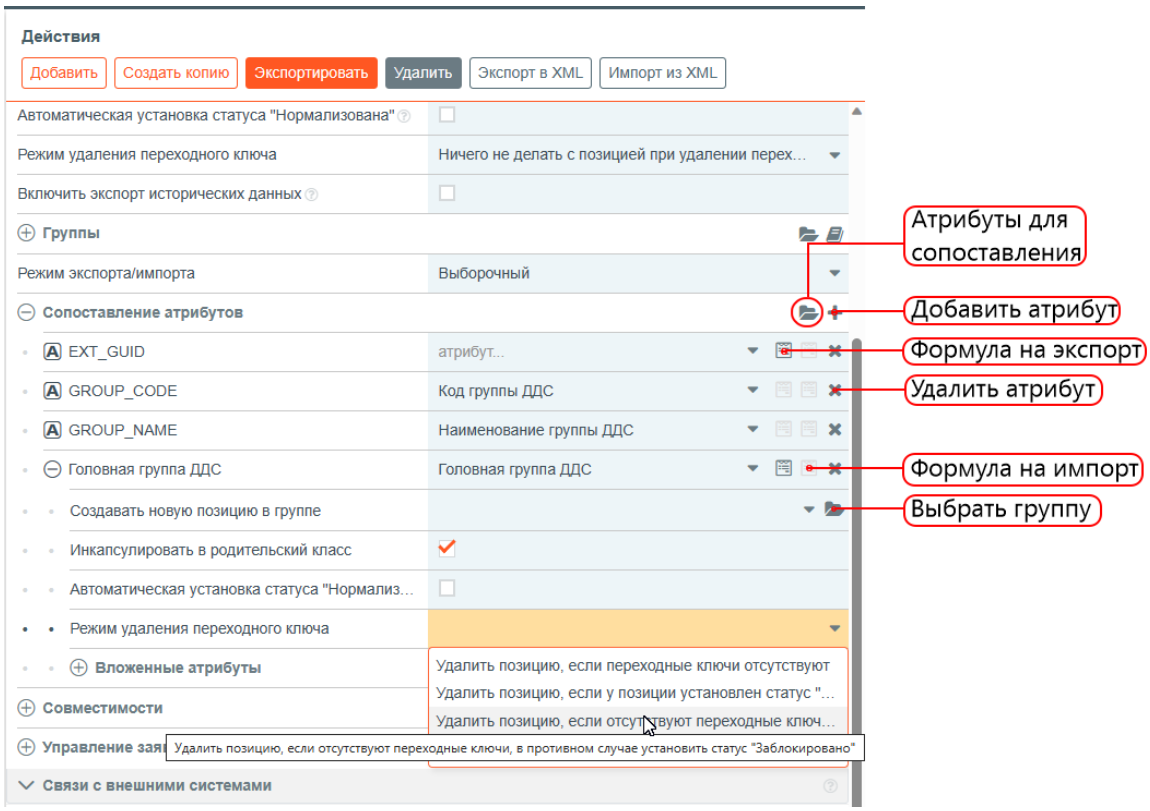


Рисунок 446 - Настройки атрибутов обменного класса в выборочном режиме

Атрибуты связи типа *Агрегация* или *Композиция*, имеют дополнительные свойства:

- **Создать новую позицию в группе** – в ячейке указывается справочная группа, в которую будет производиться запись новой позиции при импорте данных или где будет осуществляться поиск. Группа выбирается из списка;
- **Инкапсулировать в родительский класс** – при установке данного маркера все вложенные свойства (атрибуты) будут подняты на один уровень иерархии. Таким образом, атрибуты агрегируемого / композируемого объекта станут собственными атрибутами родительского обменного класса;

- **Автоматическая установка статуса «Нормализована»** – если для атрибута связи установлен данный маркер, то позиции, формирующиеся в связанном справочнике в результате импорта данных, будут автоматически получать статус «*Нормализована»*;
- **Режим удаления переходного ключа** – свойство присутствует только у атрибутов связи типа *Композиция* с признаком *Множественный*. Свойство отвечает за удаление множественной композиции. При удалении композируемой позиции, переходные ключи для текущей системы удаляются на всю глубину вложенности и удаление выполняется в соответствии с выбранным режимом удаления переходного ключа:
  - Удалить позицию, если переходные ключи отсутствуют;
  - Удалить позицию, если у позиции установлен статус «Не нормализована» и отсутствуют переходные ключи;
  - Удалить позицию, если переходные ключи отсутствуют, в противном случае установить статус «Заблокирована»;
  - Ничего не делать с позицией при удалении переходного ключа;
- **Вложенные атрибуты** – указываются те атрибуты агрегируемой / композируемой позиции, значения которых должны передаваться совместно с атрибутами основной позиции. Раздел может быть пустым, в этом случае осуществляется передача значения атрибута, имеющего свойство *Наименование позиции*.

Если для сопоставляемых атрибутов используются формулы на экспорт и/или импорт**,** то их добавление осуществляется при помощи скрипта Groovy или шаблона в специальном **Редакторе формул** (Рисунок 447). Порядок работы с редактором формул изложен в разделе 3.17. Формулы на экспорт и / или импорт используются в тех случаях, когда при проведении репликации необходимо: изменение структуры данных (например, из пяти типов атрибутов необходимо получить три); преобразование данных (например,

изменение типа данных из числа с плавающей точкой в строку или наоборот); добавление свойств (например, единиц измерения).

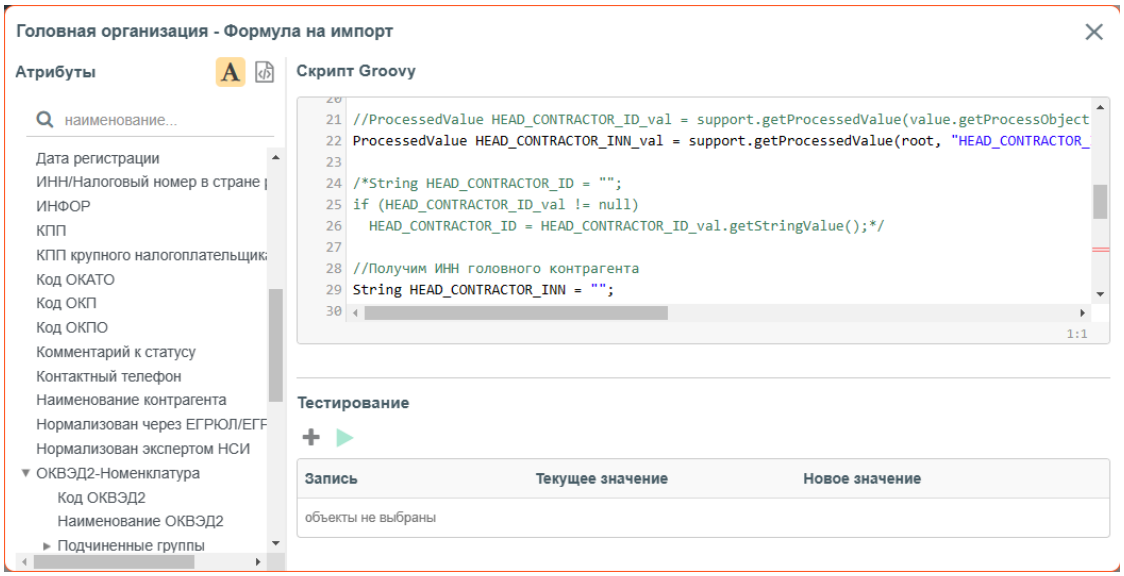


Рисунок 447 - Формула на импорт атрибута в виде скрипта Groovy

Результатом сопоставления атрибутов через формулу или шаблон может быть только простой тип данных, то есть нельзя вычислить ссылку на агрегацию, поскольку она будет восприниматься как число.

3.30.8 Настройка связи внешней ИС с обменными классами

Настройка связи между обменным классом и внешней ИС может быть осуществлена как при настройке обменного класса, так и при настройке ИС.

При настройке связи со стороны обменного класса необходимо выполнить следующие действия:

- перейти в карточке обменного класса в раздел **Связи** с внешними системами;
- в ячейке **Добавить внешнюю систему** развернуть список и щелчком левой кнопки мыши выбрать одну из зарегистрированных ИС;
- повторить выбор нужное число раз по количеству систем, в которых используется данный обменный класс;
- в ячейке с внешней ИС настроить условия экспорта для чего нажать

кнопку ***Условия экспорта*** и перейти в **Редактор условий экспорта**;

- в **Редакторе условий экспорта** (Рисунок 448) настроить условия при которых выполняется экспорт. Для этого используется создание групп условий, каждая из которых может содержать ссылки на атрибуты, скрипты и условия на работу с внешней системой. Все созданные условия могут быть протестированы в соответствующем разделе **Редактора**;

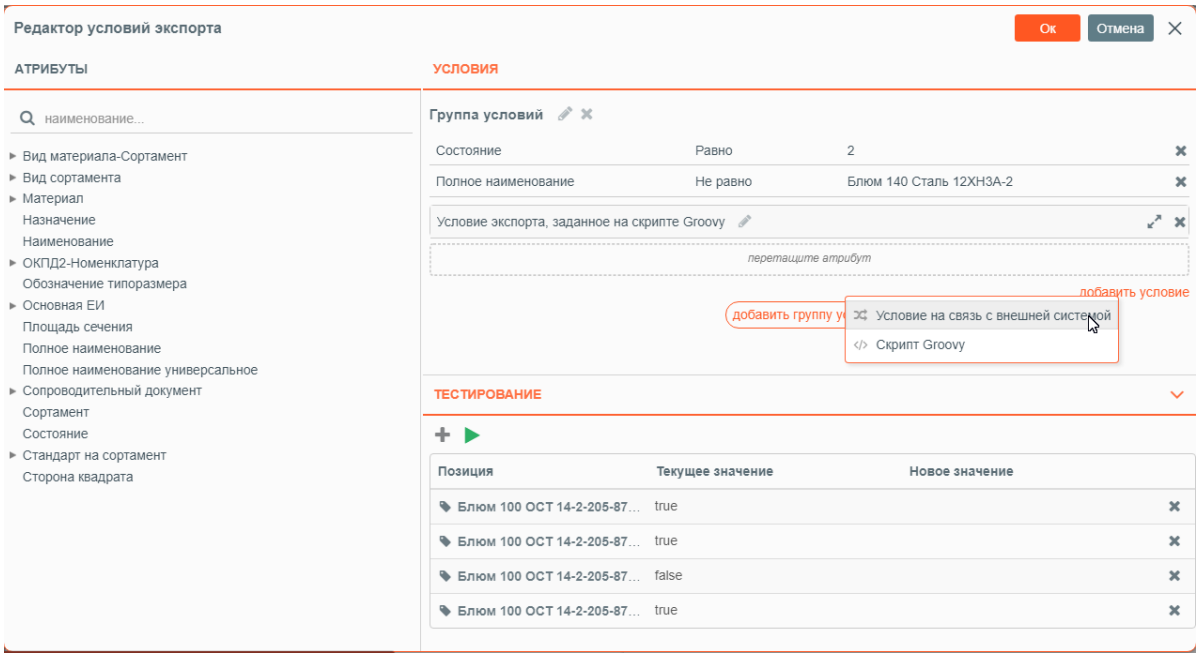


Рисунок 448 - Редактор условий экспорта

- после настройки условий указывается статус экспорта класса во внешнюю систему. Причём окончательное состояние определяется статусом экспорта самого обменного класса, указываемого в разделе *Общие параметры* (Рисунок 449):
  - *Остановлен* – изменения, выполняемые в *Semantic MDM*, игнорируются;
  - *Приостановлен* – изменения, выполняемые в *Semantic MDM*, накапливаются и будут обработаны, как только будет включён статус *Выполняется*;

- *Выполняется* – обработка изменений выполняется в онлайн режиме;

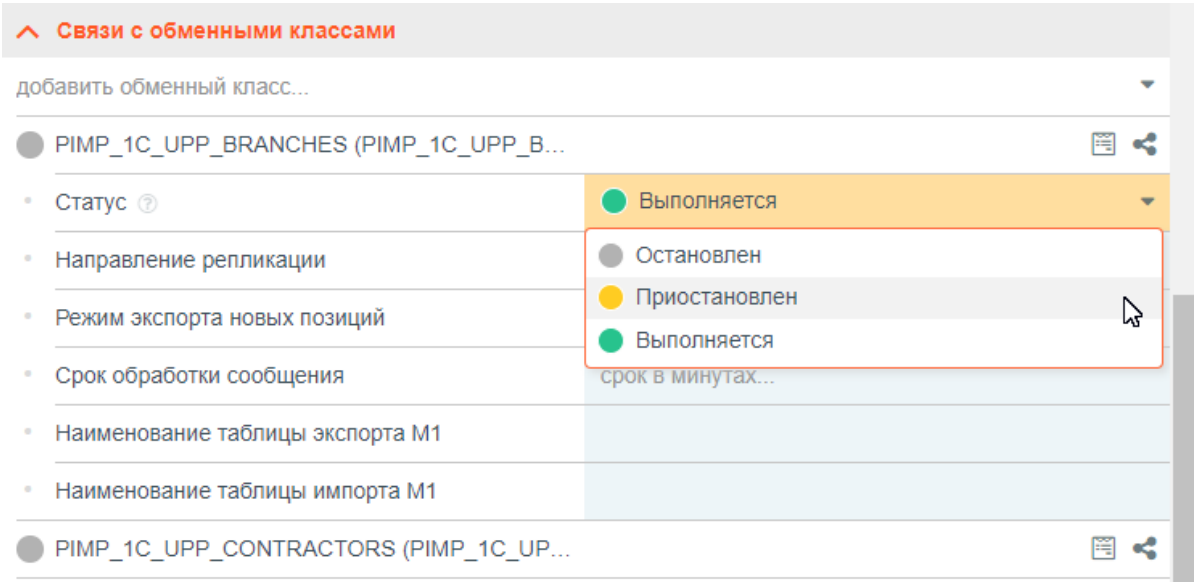


Рисунок 449 - Определение статуса экспорта обменного класса в ИС

- в ячейке **Направление репликации** (Рисунок 450) из выпадающего списка выбирается направление репликации текущего обменного класса с данной внешней ИС:
  - Импорт и экспорт;
  - Импорт;
  - Экспорт;

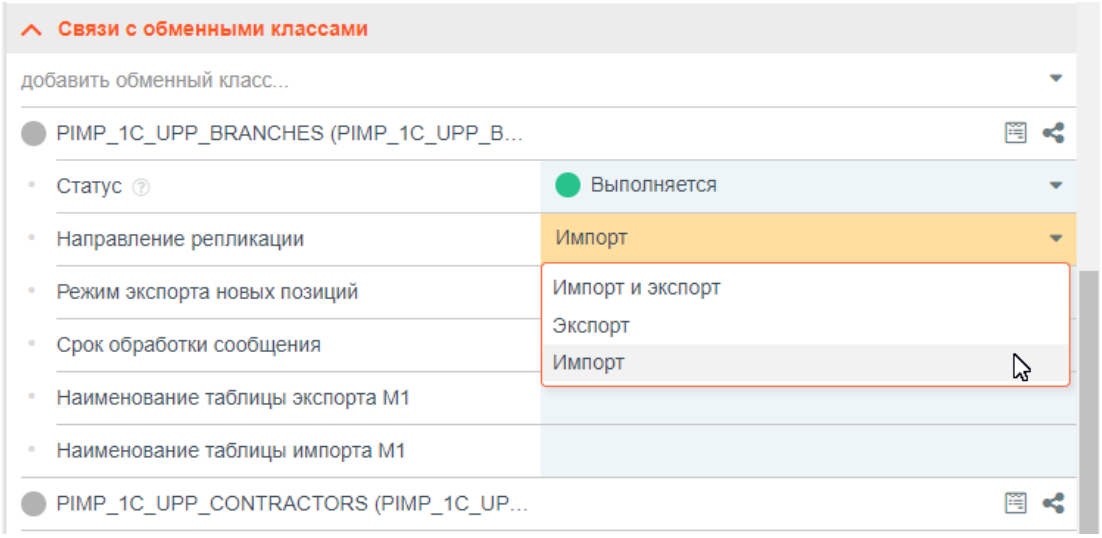


Рисунок 450 - Определение направления репликации

- **Режим экспорта новых позиций** (Рисунок 451) отвечает за способ экспорта:
  - *Выполнять автоматически* – стандартный режим работы, при котором система автоматически экспортирует все новые позиции;
  - *По запросу пользователя* – в этом режиме новая позиция экспортируется во внешнюю систему только при выполнении пользователем команды **Экспортировать** из контекстного меню на вкладке **Экспорт/Импорт позиций**. После получения переходного ключа, все изменения передаются в автоматическом режиме.

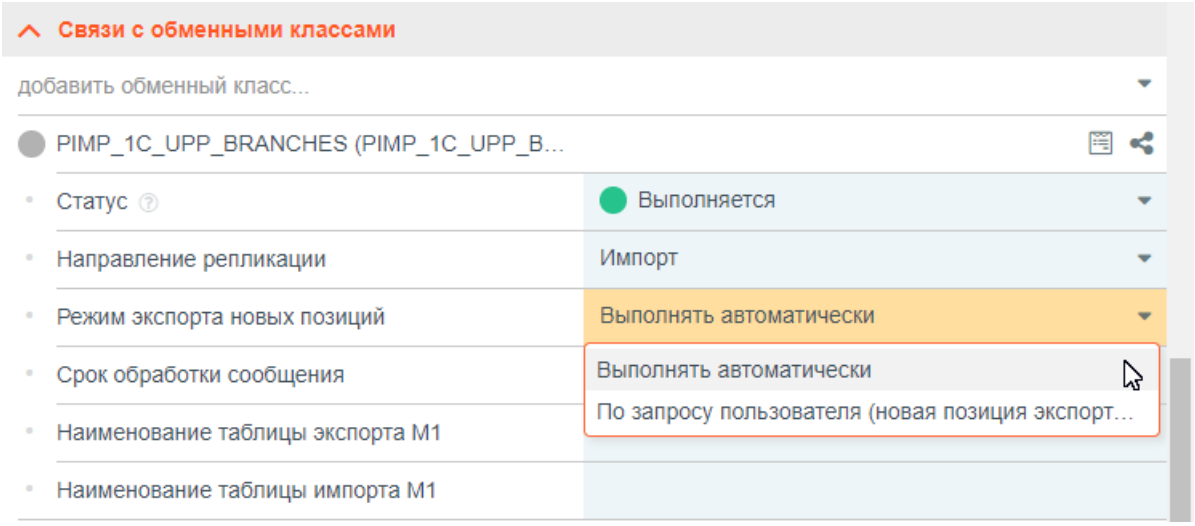


Рисунок 451 - Указание режима экспорта для новых позиций

- **Срок обработки сообщения** – в ячейке указывается время (минимум 30 мин) отведенное на обработку сообщения. Если значение не задано или равно 0, то срок обработки не отслеживается. Если задано ненулевое значение для срока обработки, то система отслеживает сообщения на экспорт со статусом *Экспорт разрешен* и, при превышении срока обработки, устанавливает для них признак превышения срока обработки. Также система отслеживает сообщения на импорт и время прошедшее от момента загрузки до момента обработки. Если время с момента загрузки сообщения на импорт

превысило заданный срок обработки и сообщение не было обработано (не получило статус *Обработана*, *Отклонена*, *Ошибка*), то система устанавливает признак превышения срока обработки для сообщения. На вкладке **Экспорт/Импорт позиций**, система выводит справа от наименования системы и обменного класса пиктограмму наличия просроченных сообщений на экспорт/импорт, если такие сообщения находятся в обменных классах с установленным статусом *Выполняется* или *Приостановлен*.

- **Наименование таблицы экспорта M1** – в ячейке указывается наименование таблицы экспорта, используемой для данного обменного класса в *Модуле импорта и репликации* предыдущей версии *Semantic MDM*;
- **Наименование таблицы импорта M1** – в ячейке указывается наименование таблицы импорта, используемой для данного обменного класса в *Модуле импорта и репликации* предыдущей версии *Semantic MDM*.

3.30.9 Настройка безопасности

Разграничение прав доступа пользователей к работе с репликацией из / в различные внешние системы производится путем настройки прав доступа на соответствующей панели карточки внешней информационной системы (Рисунок 452).

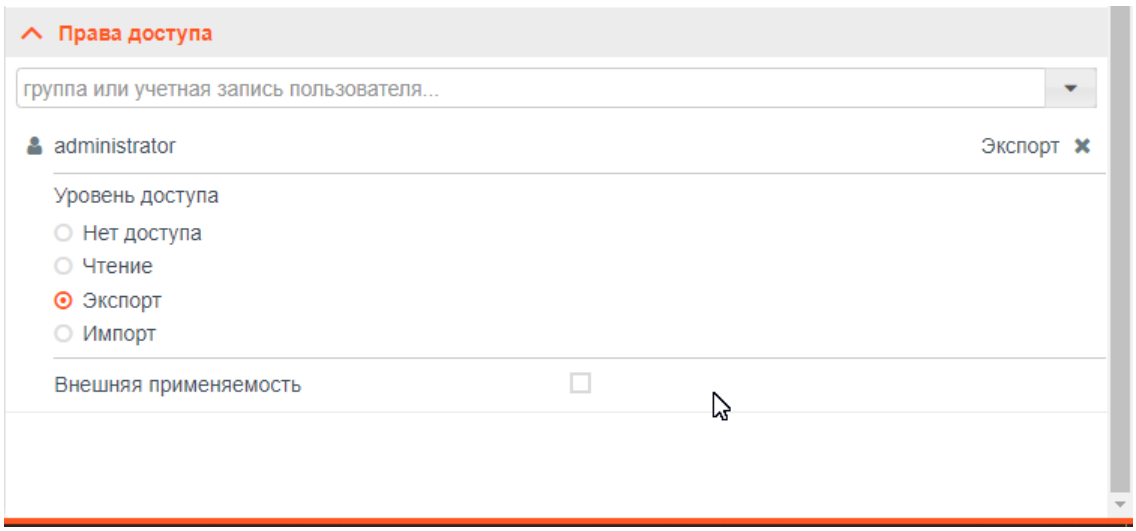


Рисунок 452 - Настройка прав доступа к работе с внешней ИС

Наименование группы пользователей или конкретного пользователя выбирается из выпадающего списка. Для каждого выбранной группы или пользователя настраивается уровень доступа:

- **Нет доступа** - указанные пользователи и группы не могут выполнять никаких действий по репликации с данной внешней ИС;
- **Чтение** - указанным пользователям и группам разрешен только просмотр настроек внешней ИС, а также результатов репликации;
- **Экспорт** - указанные пользователи и группы могут читать и запускать процедуру экспорта данных во внешнюю информационную систему из подсистемы **Справочники**;
- **Импорт** - указанным пользователям и группам доступно чтение, а также запуск процедур экспорта и импорта данных из подсистемы **Справочники**.

Маркер **Внешняя применяемость** устанавливается если пользователь может видеть применяемость данной внешней системы в карточке позиции и при дедубликации. Если маркер не установлен, то для просмотра доступны только переходные ключи.

3.30.10 Обработка агрегаций / композиций

При передаче данных через *Модуль импорта и репликации* производится обработка следующих типов связи:

- *Единичная композиция* – никогда не сравнивается через поиск дубликатов, все изменения выполняются над единственной имеющейся композированной позицией;
- *Инверсная (обратная) композиция* – например, если филиалы контрагента композированы в головную организацию и для одного из филиалов требуется указать нового головного контрагента или если для композированной позиции получен новый идентификатор родителя, то выполняется перекомпозирование имеющейся позиции в нового родителя. При выполнении экспорта проверяется наличие связи с системой для обратной композиции. Логика для обратной композиции, такая же как и для одиночной агрегации. Например, экспорт композированных филиалов будет выполняться только после того, как головной контрагент (обратная композиция) был реплицирован в указанные ИС;
- *Множественная композиция* – предполагается, что всегда импортируется полный список имеющихся композиций. Сопоставление присланных композированных позиций выполняется через поиск дубликатов с имеющимися в системе. Все отсутствующие в системе объекты, из присланных в сообщении, добавляются, а отсутствующие удаляются в соответствии с настроенным режимом удаления.
- *Множественная агрегация* – предполагается, что всегда импортируется полный список имеющихся агрегаций. Все отсутствующие агрегируемые позиции в системе, из присланных в сообщении, добавляются, а все отсутствующие в присланном сообщении, для которых в системе имеется переходной ключ с текущей внешней системой, удаляются.

3.30.11 Просмотр реплицированных объектов

Вкладка **Экспорт/Импорт позиций** позволяет просматривать, повторно реплицировать и удалять позиции и таблицы в целом. Экспортированные и импортированные позиции располагают на отдельных дополнительных вкладках.

В левой части вкладки располагается список зарегистрированных внешних ИС, а также обменных классов, связанных с данными системами. Справа от таблицы позиций находится карточка текущего обменного класса или внешней ИС (зависит от того, на чем установлен курсор).

Устанавливая курсор на наименовании ИС или обменных классов можно просматривать таблицы реплицированных позиций. Над таблицей располагаются сведения об общем числе отправленных и обработанных записей, ячейка быстрого поиска, кнопка расширенного поиска, а также кнопка доступа к настройкам **видимости колонок таблицы** (Рисунок 453). Колонки для таблиц с экспортированными и импортированными позициями имеют разные настройки.

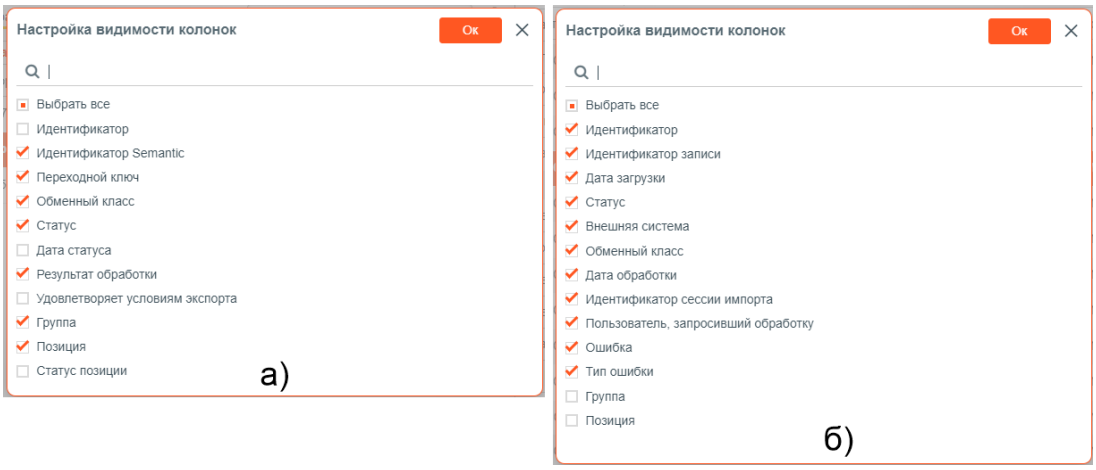


Рисунок 453 - Настройка видимости колонок таблиц:
а) для экспортированных позиций; б) для импортированных позиций

Таблицы реплицированных позиций поддерживают быстрый поиск по наименованию или идентификатору системы, по наименованию или идентификатору обменного класса, а также фасетную фильтрацию. Над

таблицей располагается указание количества записей с определенным статусом, которые также могут использоваться для фильтрации. Расширенный поиск доступен по критерию *Дата статуса*. Дата устанавливается с помощью временной шкалы, путем перемещения бегунка или вручную с помощью выпадающего календаря.

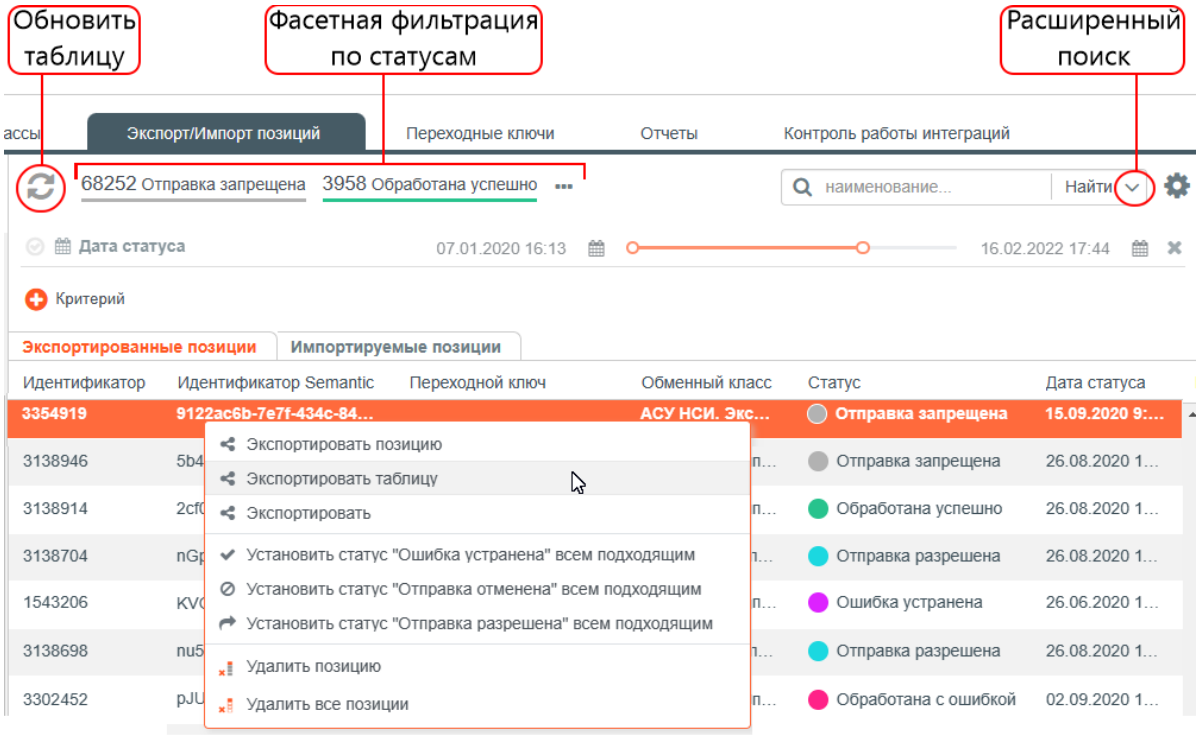


Рисунок 454 - Вкладка **Экспортированные позиции** и её контекстное меню

При просмотре позиций над таблицей пользователю доступна кнопка ***Обновить таблицу***, которая позволяет обновить записи, в соответствии с их текущим статусом. Кнопка доступна как для вкладки **Экспортированные позиции** (Рисунок 454), так и для вкладки **Импортируемые позиции**.

В *Системе* возможен экспорт в *Excel* таблиц с сообщениями экспорта и импорта. Экспортируются только те колонки, которые выбраны для просмотра в табличном представлении. Для выполнения экспорта требуется установить курсор в дереве на наименовании внешней системы и затем нажать кнопку ***Экспорт в Excel*** (Рисунок 455) на панели инструментов таблицы над вкладкой **Экспортированные позиции** или **Импортированные позиции**.

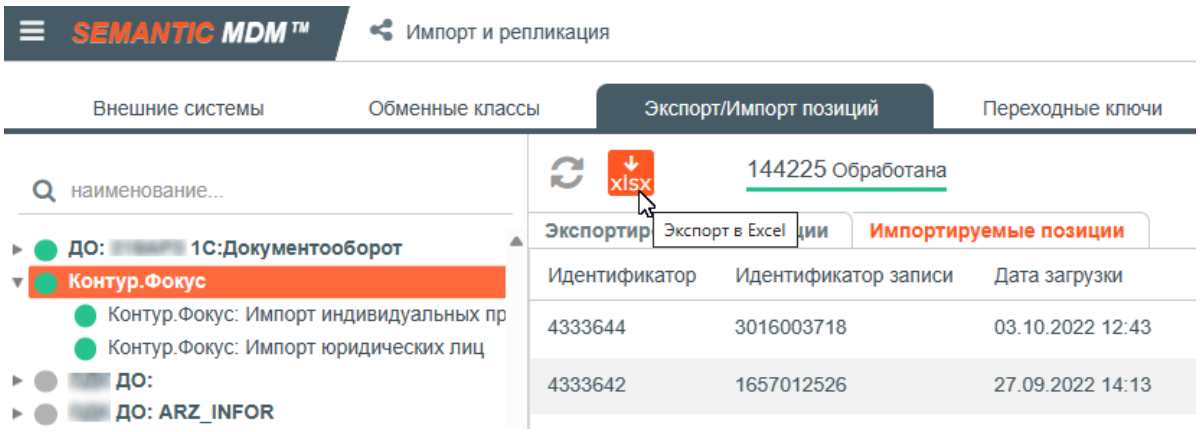


Рисунок 455 - Выполнение экспорта в Excel данных из таблицы сообщений экспорта и/или импорта

Находясь на сетке данных вкладки **Экспортированные позиции** можно повторно экспортировать текущую позицию или группу, а также удалить текущую позицию или все позиции в таблице. Для этого используются команды контекстного меню.

- **Отфильтровать выбранные** – в таблице будут отображены только ранее выбранные пользователем позиции;
- **Экспортировать позицию** – будет произведен экспорт позиции, на которой установлен курсор;
- **Экспортировать таблицу** – будет выполнен повторный экспорт позиций, присутствующих в текущей таблице, с учетом примененных фильтров, например, только позиции с ошибками;
- **Экспортировать** - команда применяется только, если были внесены изменения в состав обменного класса или в условия репликации. В этом случае, выполняется экспорт всех позиций обменного класса, а не только тех, которые были в таблице;
- **Экспортировать выбранные** – будет выполнен повторный экспорт выбранных в таблице позиций;
- **Установить статус «Ошибка устранена» всем подходящим** – команда применяется к предварительно отобранным с помощью быстрого, расширенного поиска или фильтрации позициям со статусом «Обработана с ошибкой»;

- **Установить статус «Ошибка устранена»** – команда доступна только для позиций со статусом «Обработана с ошибкой». Такое изменение статуса разрешено, если эксперт НСИ считает, что возникшая ошибка является, например, результатом некорректной работы адаптера;
- **Установить статус «Отправка отменена» всем подходящим** – команда применяется к предварительно отобранным с помощью быстрого, расширенного поиска или фильтрации позициям;
- **Установить статус «Отправка разрешена» всем подходящим** – команда применяется к предварительно отобранным с помощью быстрого, расширенного поиска или фильтрации позициям;

При повторном экспорте позиции или таблицы необходимо выбрать режим выгрузки (Рисунок 456):

- **Все позиции** – будут повторно выгружены все позиции (выбранная или все содержащиеся в таблице). применяется при первичной репликации данных;
- **Изменённые позиции** – повторно будут экспортированы только те позиции, которые присутствуют в репликационной таблице и в которых произошли изменения по сравнению с состоянием при последней выгрузке.

Установить курсор на наименовании требуемой внешней ИС и нажать кнопку ***Экспортировать***. Если необходимо выполнить экспорт во все настроенные ИС, то курсор следует оставить на наименовании обменного класса.

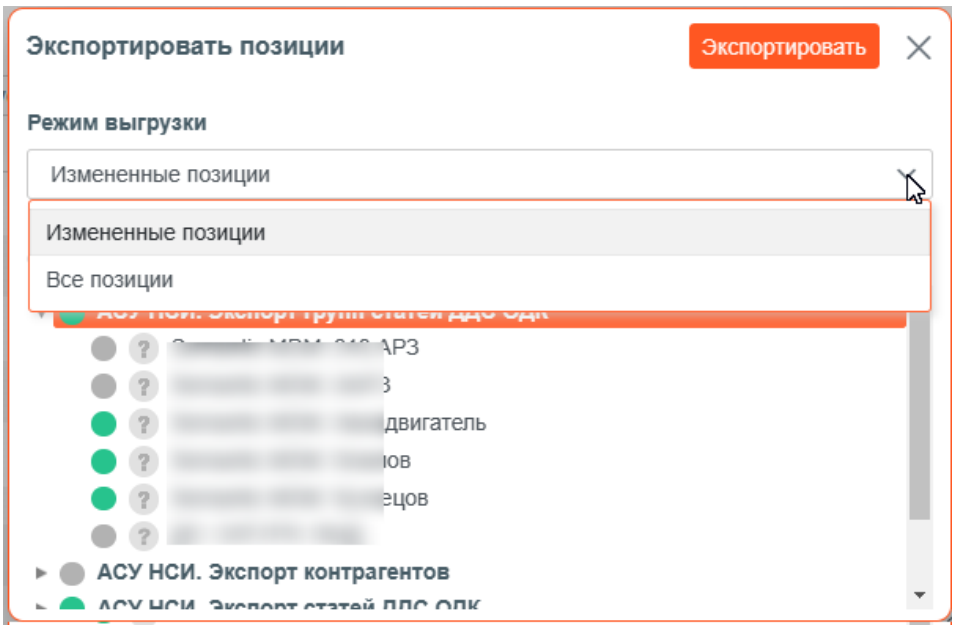


Рисунок 456 - Настройки при повторном экспорте

При удалении позиции или таблицы необходимо подтвердить выполнение данного действия (Рисунок 457).

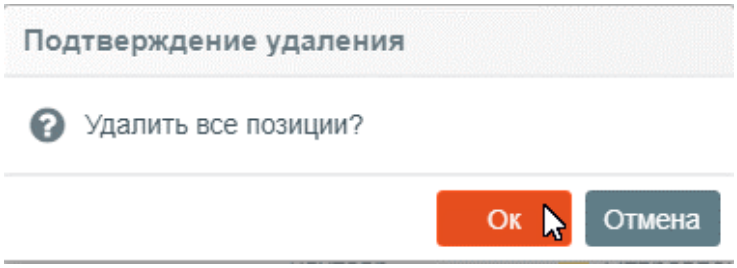


Рисунок 457 - Подтверждение удаления

Если в результате экспорта запись получила статус **Ошибка экспорта** (Рисунок 458), то это означает, что для данной позиции были обнаружены следующие проблемы:

- Было сформировано более одного сообщения, удовлетворяющего условиям экспорта, т.е. позиция экспортируется во внешнюю ИС более чем через один обменный класс. В тексте ошибки в карточке будет следующее описание ошибки: «Ошибка в настройках условий экспорта для обменных классов <имя обменного класса> позволяет выполнить экспорт позиции с идентификатором <GUID позиции> более чем для одного обменного класса». Для устранения ошибки

необходимо изменить условиях экспорта таким образом, чтобы позиция могла быть выгружена только через один обменный класс;

- Наличие ошибки в скрипте условий экспорта. В этом случае в тексте ошибки будет написано: «Ошибка в условиях репликации»;
- Наличие ошибки в формуле экспорта свойства. В этом случае в тексте ошибки будет написано: «Ошибка в формуле на экспорт для свойства»;
- Наличие ошибки в формуле экспорта обменного класса. В этом случае в тексте ошибки будет написано: «Ошибка в формуле на экспорт для обменного класса».

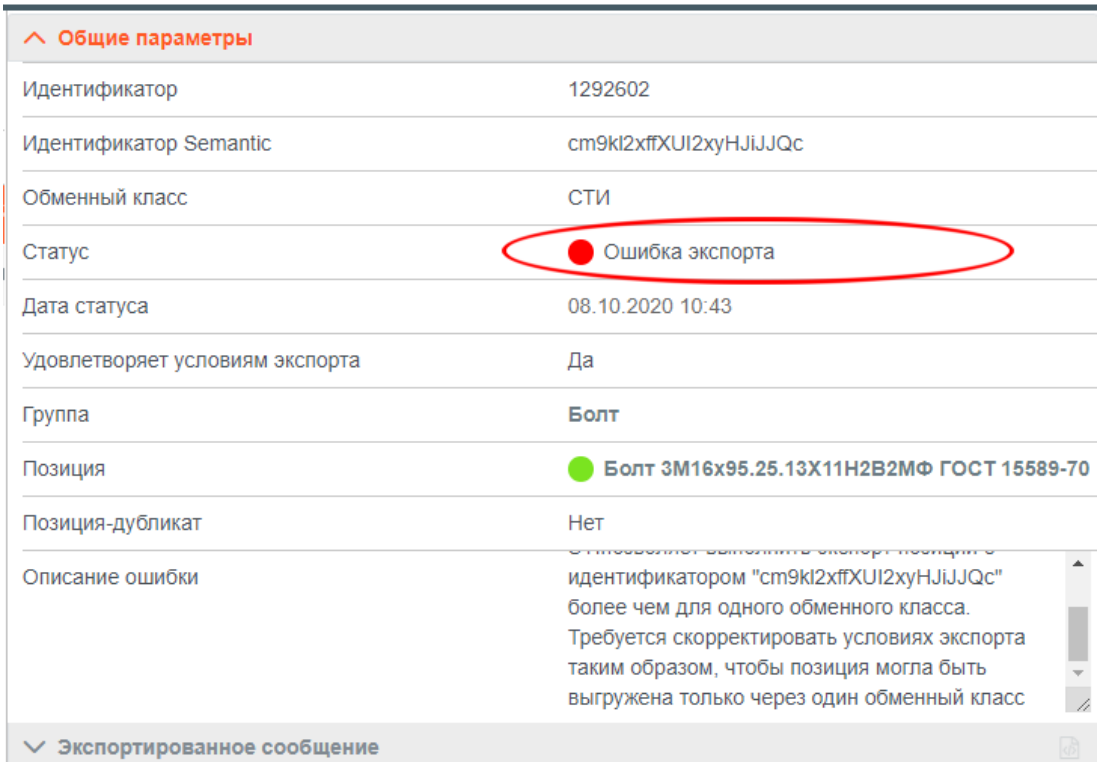


Рисунок 458 - Запись обменного класса со статусом **Ошибка экспорта**

В карточке экспортированной позиции в разделе **Значения** (Рисунок 459) можно просматривать все экспортированные значения атрибутов позиции или только измененные. Для этого в разделе присутствует соответствующая кнопка.

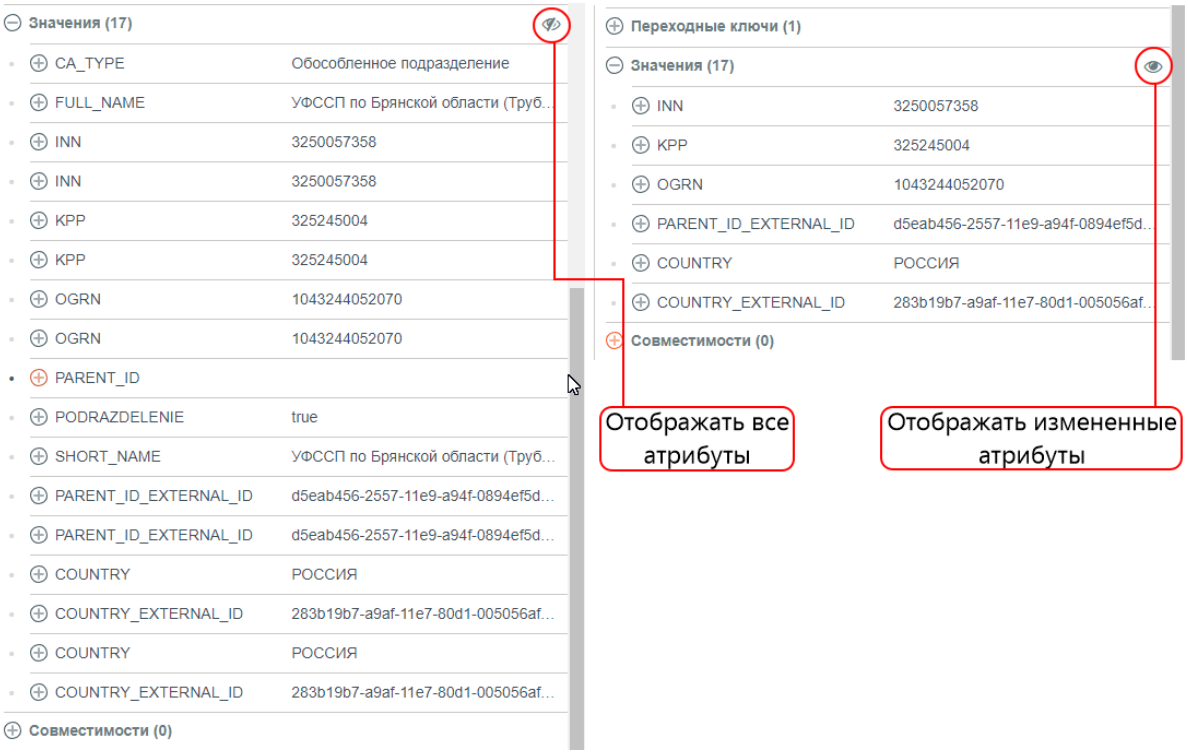


Рисунок 459 - Просмотр значений в сообщении экспорта - все атрибуты / измененные атрибуты

Находясь на сетке данных вкладки **Импортируемые позиции** (Рисунок 460) можно повторно обработать текущую позицию или группу, отклонить импорт указанной позиции, а также удалить текущую позицию или все позиции в таблице. Для этого используются команды контекстного меню.

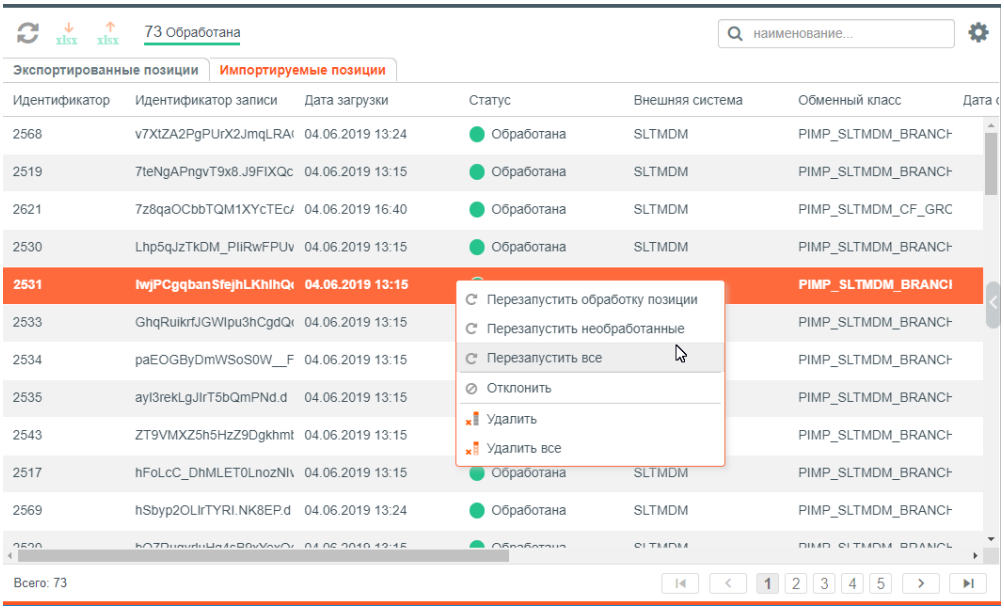


Рисунок 460 - Действия из контекстного меню на вкладке **Импортируемые позиции**

Выполнение перезапуска импорта не требует никаких подтверждений.

Если для записи выбрана опция **Отклонить**, то необходимо указать причину отклонения в окне **Подтверждение** (Рисунок 461). Такая запись получает статус **Отклонена**. Она может быть повторно импортирована командой **Перезапустить обработку позиции**, после устранения ошибок.

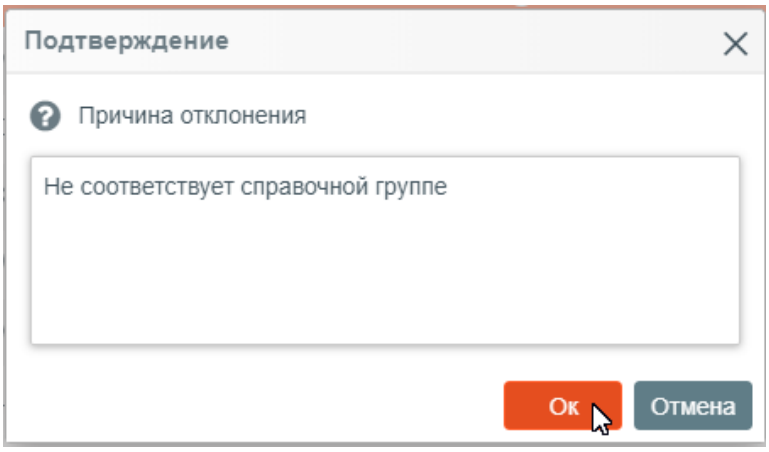


Рисунок 461 - Подтверждение отклонения импортируемой позиции

3.30.12 Статусы реплицированных позиций

В процессе репликации, как экспорта так и импорта, сообщения изменяют свой статус. Каждый статус соответствует определенному этапу репликации и состоянию сообщений. Общее количество сообщений того или иного статуса показывается над таблицей на вкладке **Экспорт / Импорт позиций** (Рисунок 462).

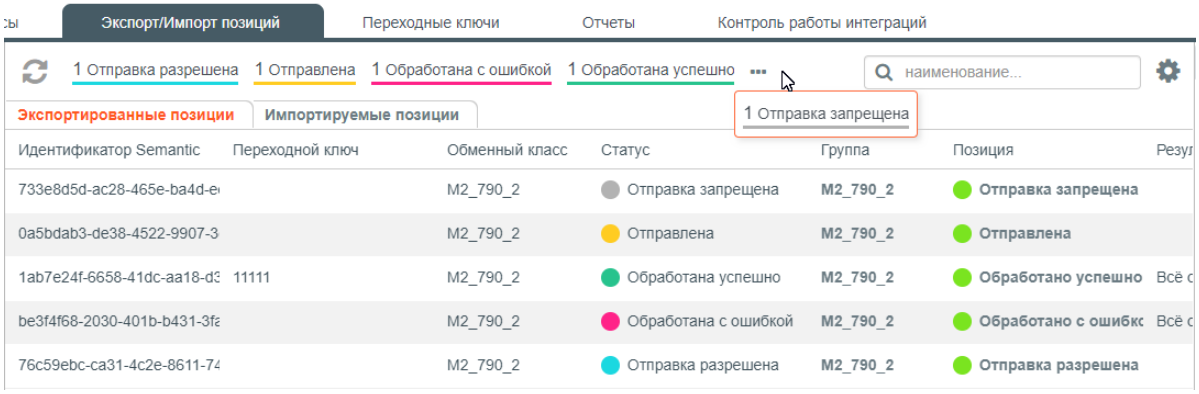


Рисунок 462 - Экспортированные сообщения со статусами состояния

Статусы, которые могут принимать экспортированные сообщения перечислены в таблице Таблица 31. Каждый статус сопровождается цветной пиктограммой, цвет которой обозначает тот или иной статус. Схема изменения статусов экспортированного сообщения приведена на Рисунок 463

Таблица 31 – Статусная модель экспортированных сообщений

Пиктограмма	Статус	Комментарий
	Отправка запрещена	Позиция передана в обменный класс, но не прошла проверку на выполнение условий экспорта или наличие переходного ключа с внешней ИС
	Отправка разрешена	Позиция прошла проверку на выполнение условий экспорта и наличие переходного ключа и может быть передана во внешнюю ИС
	Отправка отменена	Позволяет отменить отправку сообщений внешней системе и запустить её позже, путём возврата этим сообщениям статуса «Отправка разрешена». Смена статуса выполняется системным администратором для выбранных сообщений экспорта через контекстное меню
	Отправлена	Позиция отправлена, но ответ от ИС с результатами обработки пока не получен. Статус возникает при раздельном выполнении приема и обработки записей внешним интеграционным адаптером
	Обработана успешно	Позиция создана, изменена, блокирована или удалена во внешней ИС, получено сообщение об успешной репликации
	Ошибка экспорта	Произошла ошибка на этапе формирования сообщения (подробнее см. 3.30.11). Ошибка может возникнуть из-за того, что позиция экспортируется во внешнюю ИС более чем через один обменный класс или из за ошибок в скриптах на условия экспорта. Причина указана в описании ошибки
	Обработана с ошибкой	Произошла ошибка обработки записи на стороне интеграционного адаптера ИС. Причина ошибки содержится в сообщении json.

	Ошибка устранена	Статус устанавливается вручную *Экспертом НСИ* для записей со статусом **Ошибка экспорта**, если ошибка была устранена или ее причина является несущественной

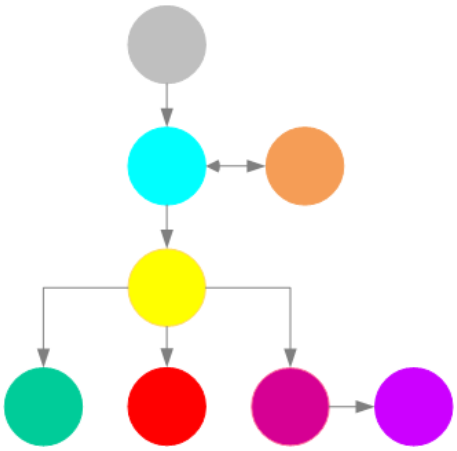


Рисунок 463 - Схема изменения статусов экспортированного сообщения

Статусы, которые могут принимать импортированные сообщения перечислены в таблице ниже. Каждый статус сопровождается цветной пиктограммой, цвет которой обозначает тот или иной статус. Схема изменения статусов импортированного сообщения приведена на рисунке Рисунок 464.

Таблица 32 – Статусная модель импортированных сообщений

Пиктограмма	Статус	Комментарий
	Сессия активна	После старта интеграционным адаптером сессии импорта, все загруженные этим адаптером сообщения будут иметь данный статус
	Ожидает обработки	После закрытия сессии импорта все сообщения, которые были загружены в рамках сессии, получат этот статус
	Обрабатывается	Запущена процедура импорта из обменного класса в *Semantic MDM*
	Обработана	Импортированная позиция создана, изменена или удалена в *Semantic MDM*
	Отклонена	Импорт позиции отклонен. Отклонение позиции может быть выполнено в автоматическом режиме, если позиция не прошла проверку по условиям импорта, либо она может быть отклонена *Экспертом НСИ*

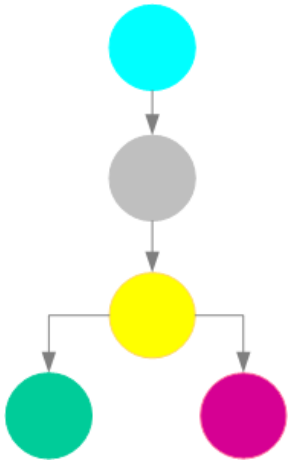


Рисунок 464 - Схема изменения статусов импортированного сообщения

3.30.13 Загрузка данных в таблицу импорта из Excel

Одним из наиболее распространённых способов первичной загрузки данных из внешней системы, является загрузка из файла Excel с согласованным набором полей. Эта процедура применима для первичной загрузки данных из внешней системы и также может быть использована для периодической загрузки изменений из внешней системы вручную.

Для выполнения процедуры импорта / экспорта данных таблицы в формат Excel следует перейти на дополнительную вкладку **Импортируемые позиции,** установить курсор на нужном обменном классе под наименованием внешней ИС и нажать кнопку ***Импорт из Excel*** / ***Экспорт в Excel***, расположенную над таблицей с реплицируемыми позициями (Рисунок 465).

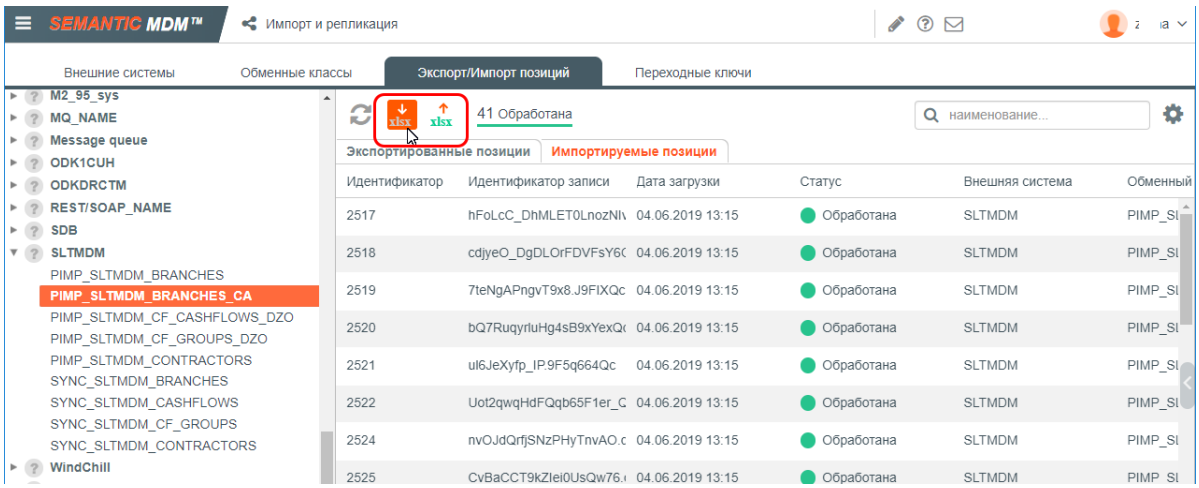


Рисунок 465 - Загрузка / выгрузка данных таблицы импорта в *Excel*

Если в обменном классе присутствует атрибут связи (агрегированный атрибут), то обязательным условием импорта из Excel является установка у этого атрибута свойства **Инкапсулировать в родительский класс** (Рисунок 466). В случае отсутствия этой опции, загрузка из Excel не возможна (функция не активна).

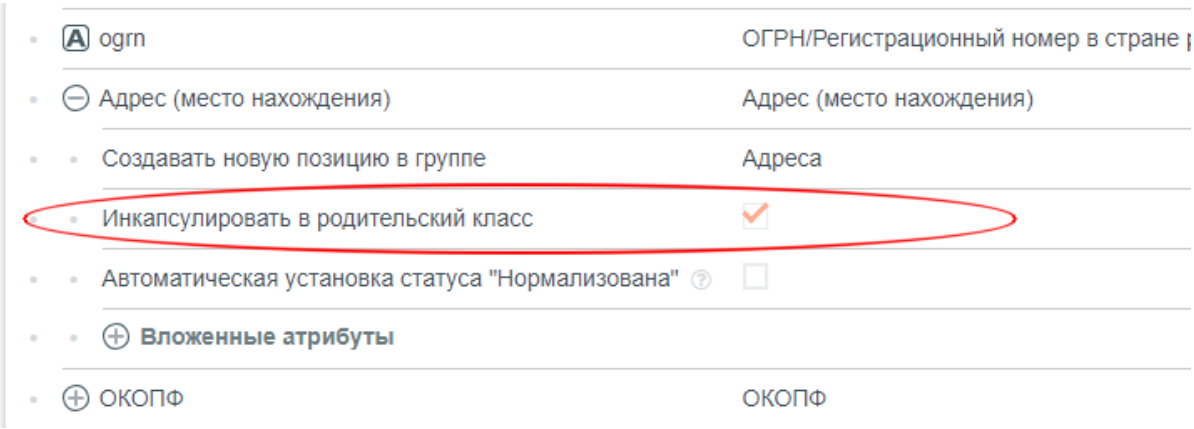


Рисунок 466 - Настройка атрибута связи при импорте данных

3.30.14 Переходные ключи

Вкладка **Переходные ключи** (Рисунок 467) предназначена для просмотра и работы с переходными ключами. Переходные ключи возникают у каждой позиции при выполнении её импорта / экспорта во внешнюю ИС. Каждая справочная позиция имеет столько переходных ключей, сколько

внешних ИС работают с ней.

При создании переходного ключа в процессе репликации данных Система в автоматическом режиме создает и активизирует пакет изменений, что позволяет Эксперту НСИ отследить моменты возникновения, изменения и удаления переходного ключа.

В обычном режиме на вкладке может быть просмотрена таблица переходных ключей по каждой зарегистрированной внешней ИС. При установке курсора на любой позиции в таблице справа от таблицы будет видна карточка реплицированной позиции, дублирующая поля таблицы с указанием:

- *Внешний идентификатор* – идентификатор (ID) позиции во внешней информационной системе;
- *Идентификатор Semantic* – идентификатор (ID) позиции в справочнике;

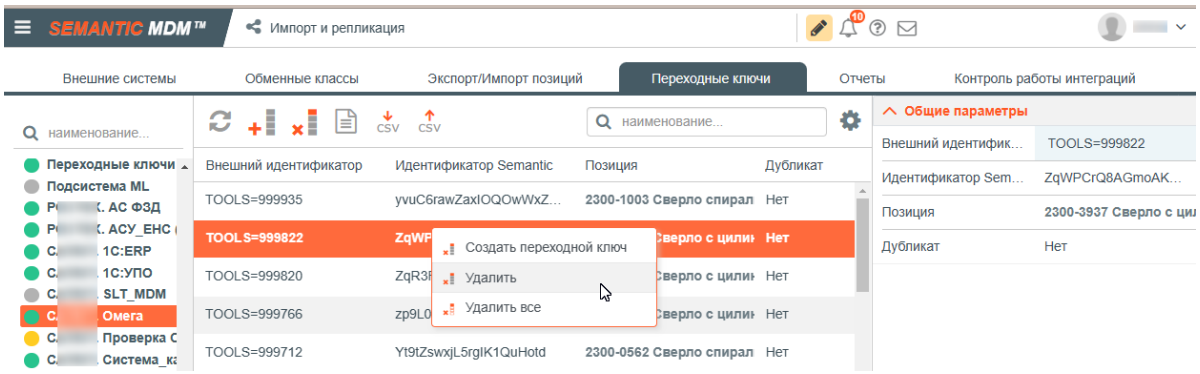


Рисунок 467 - Вкладка **Переходные ключи**

- *Позиция* – наименование позиции в справочнике, щёлкнув по которому можно перейти в сам справочник и спозиционироваться на текущей записи;
- *Дубликат* – в столбце указывается наличие дубликата у реплицированной записи.

Кроме просмотра реплицированных позиций в режиме просмотра можно сформировать **шаблон для импорта ключей** в формат CSV для чего используется кнопка , расположенная над таблицей . Шаблон может быть

использован для создания и последующей загрузки таблицы позиций с переходными ключами.

После нажатия на кнопку ***Шаблон для импорта ключей*** необходимо выполнить настройку параметров CSV в одноименном окне (Рисунок 468). В настройке выбирается кодировка, отображение кавычек и знак разделителя для значений. После настройки необходимо нажать кнопу ***Ок*** и шаблон будет сформирован и выгружен на ПК пользователя.

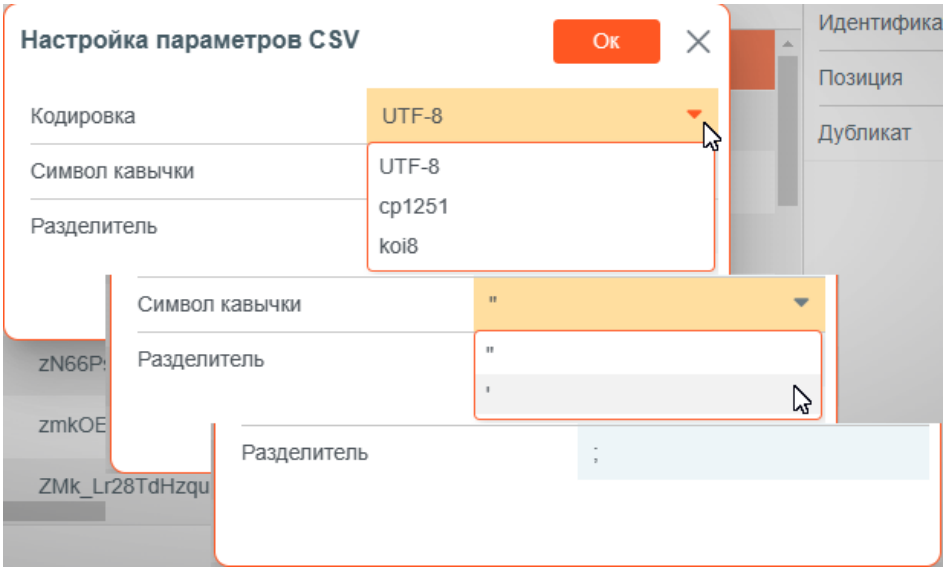


Рисунок 468 - Настройка параметров для формирования шаблона файла CSV

Переходные ключи могут быть выгружены (кнопка ***Экспорт в CSV***) или загружены (кнопка ***Импорт из CSV***) из файлов формата CSV. Для экспорта также, как и при выгрузки шаблона необходимо выбирать кодировку, отображение кавычек и знак разделителя для значений. При экспорте переходных ключей в файл выгружаются:

- Идентификатор внешней системы;
- Внешний идентификатор объекта;
- Признак дубликата;
- Идентификатор Semantic MDM;
- Наименование позиции в Semantic MDM.

Вкладка **Переходные ключи** поддерживает режим редактирования, включая импорт из файлов формата CSV, который активизируется кнопкой ***Управление пакетами изменений***.

В режиме редактирования в контекстном меню пользователю доступны следующие действия:

- **Создать переходный ключ** – команда доступна с одноименной кнопки и из контекстного меню. Позволяет создать новую позицию репликации. Для этого используется специальное всплывающее окно **Создание переходного ключа** (Рисунок 469), где можно выбрать (из справочных групп) требуемую позицию и указать внешний идентификатор (ID внешней ИС) для неё. После нажатия на кнопку ***Ок*** новая позиция будет размещена в таблице;

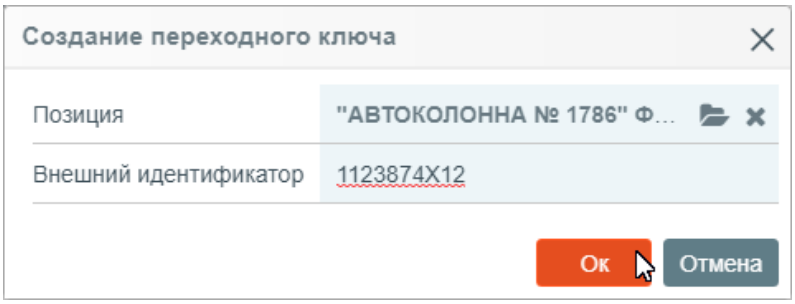


Рисунок 469 - Окно **Создание переходного ключа**

- **Удалить** – команда (доступна с одноименной кнопки и из контекстного меню) удаляет переходный ключ выбранной позиции с текущей информационной системой;
- **Удалить все** – позволяет удалить все переходные ключи в данной таблице.

3.30.15 Отчёты

Вкладка **Отчеты** представления **Импорт и репликация** предназначена для формирования, просмотра и выгрузки различных отчетов, связанных с работой модуля.

3.30.16 Импорт и репликация в карточке позиции

Карточка позиции в секции **Импорт и репликация** (Рисунок 470) содержит:

- кнопку ***Экспортировать*** для выполнения передачи позиции в выбранные внешние ИС;
- кнопку ***Обновить*** для выполнения процедуры обновления текущих данных по позиции;
- перечень внешних информационных систем, с которыми у неё есть переходные ключи, в которые она может быть экспортирована и из которых может быть импортирована;
- для каждой из систем можно посмотреть имеющиеся переходные ключи, сообщения JSON, которые были отправлены во внешнюю систему или были получены из внешней системы;

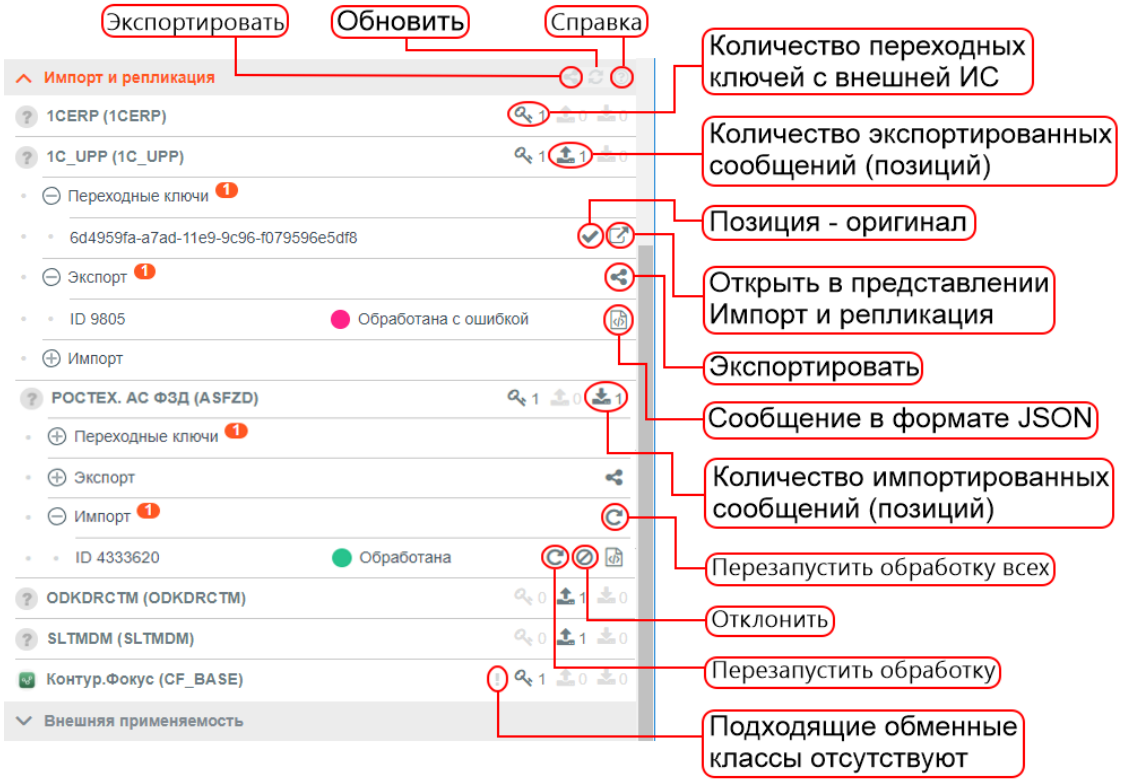


Рисунок 470 - Раздел **Импорт и репликация** в карточке позиции

- для импортированных данных можно в ручном режиме отклонить сообщение, на основе которого была сформирована данная позиция. Например, эксперт НСИ может отклонить информацию, пришедшую из внешней системы, так как она содержит противоречивую информацию и позиция не может быть нормализована на основе данной информации;
- импортированные сообщения могут быть перезапущены непосредственно из карточки позиции;
- в разделе *Экспорт* при наведении курсора на статус позиции всплывает информационное сообщение (хинт), в котором указывается результат обработки;
- позиция может быть быстро экспортирована в выбранную внешнюю систему без дополнительных диалогов через команду контекстного меню на сетке данных **Экспортировать позицию.** Эта команда применима в случаях, когда возвращена ошибка на ранее отправленное сообщение или если настроен режим экспорта по запросу пользователя (см. п. 3.30.8).

3.30.17 Экспорт таблиц и позиций из справочников

Для позиций, расположенных в справочных группах, для которых настроен экспорт во внешние информационные системы, возможно проведение принудительной процедуры экспорта как одной (нескольких) позиции, так и всей таблицы.

Процедура экспорта вызывается из контекстного меню командами **Экспортировать позицию** или **Экспортировать таблицу** (Рисунок 471).

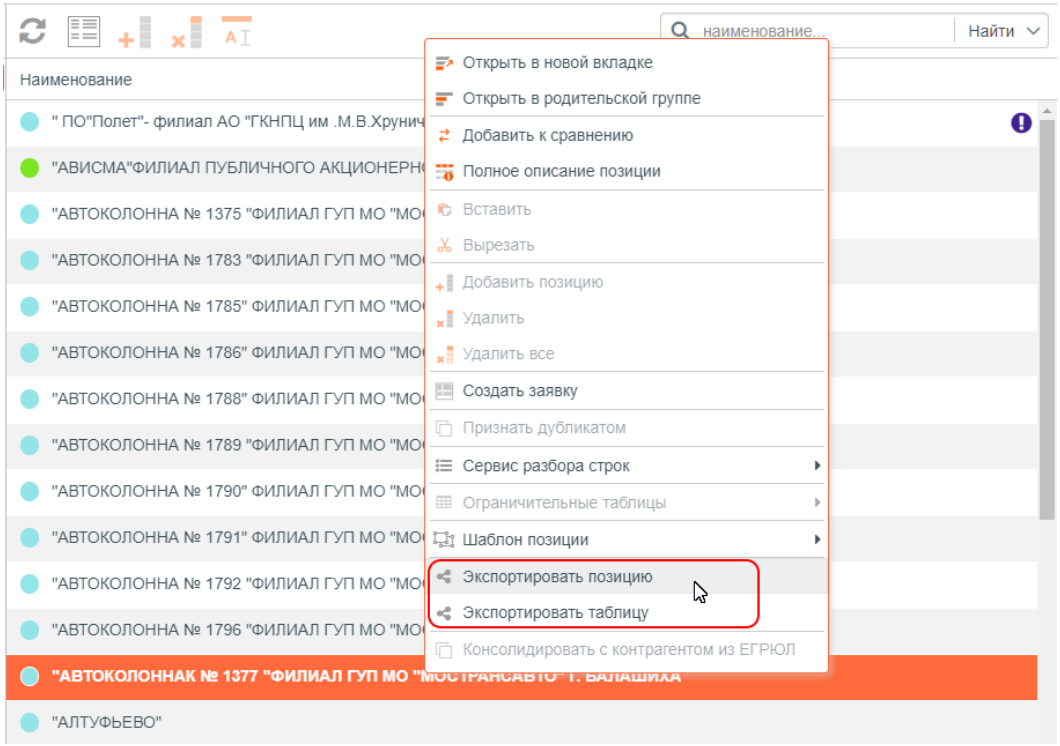


Рисунок 471 - Вызов команд экспорта из контекстного меню на сетке данных

При выполнении экспорта таблиц учитываются все фильтры, примененные на момент запуск процедуры. Например, если данные были предварительно отфильтрованы по какому-либо признаку, то в сообщение на экспорт попадут только позиции, оказавшиеся в выборке.

После вызова команды **Экспортировать позицию** (**таблицу**) будет открыто окно для выбора внешней ИС и обменного класса, куда будет произведен экспорт данных (Рисунок 472). Также здесь выбирается режим выгрузки:

Измененные позиции – выгружаются только те позиции, которые изменились с момента последнего экспорта;

Все позиции - выгружаются все выбранные позиции вне зависимости от наличия в них изменений.

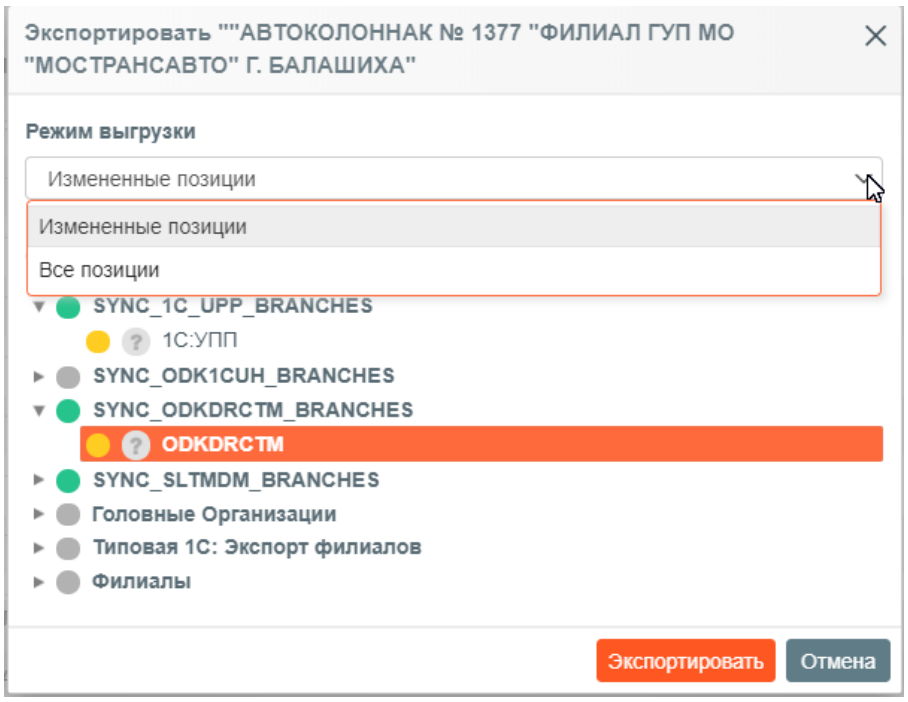


Рисунок 472 - Выбор внешней ИС и режима выгрузки данных из таблицы данных

После настройки следует нажать кнопку ***Экспортировать***, после чего *Система* отправит выбранные данные во внешнюю ИС. Экспорт выполняется в соответствии с настройками выбранного обменного класса.

Детальный вывод информации о количестве позиций и системах, в которые не удалось экспортировать выбранные позиции из табличного представления или карточки позиции, появляется во всплывающем окне (Рисунок 473). При этом указывается количество позиций, которые не были экспортированы во внешние ИС.

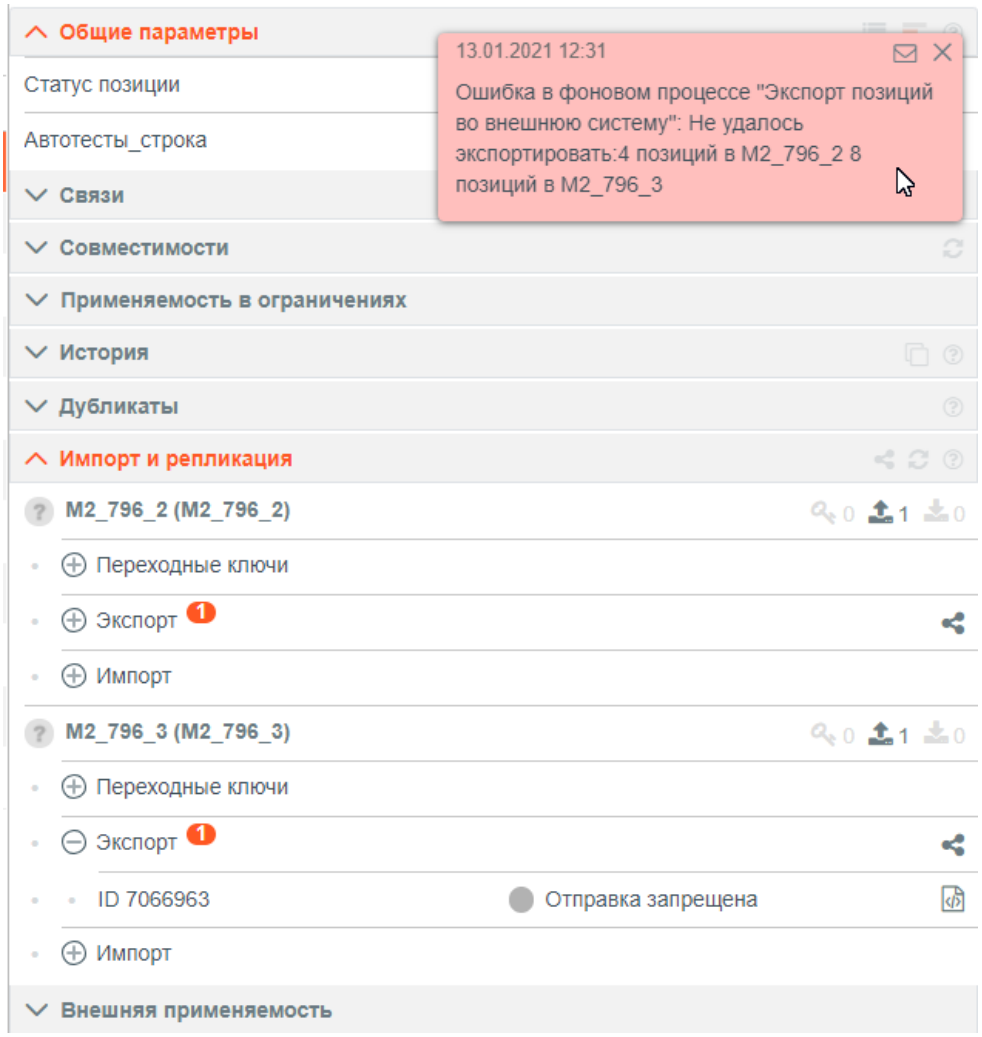


Рисунок 473 - Детализация ошибок экспорта во внешние ИС

3.30.18 Контроль работы интеграций

В представлении **Импорт и репликация** присутствует вкладка **Контроль работы интеграций** (Рисунок 474), предназначенная для анализа состояния модуля импорта и репликации.

В верхней части над таблицами указывается текущая дата, когда выполняется просмотр данных. Внизу справа указано время последнего обновления информации по интеграции.

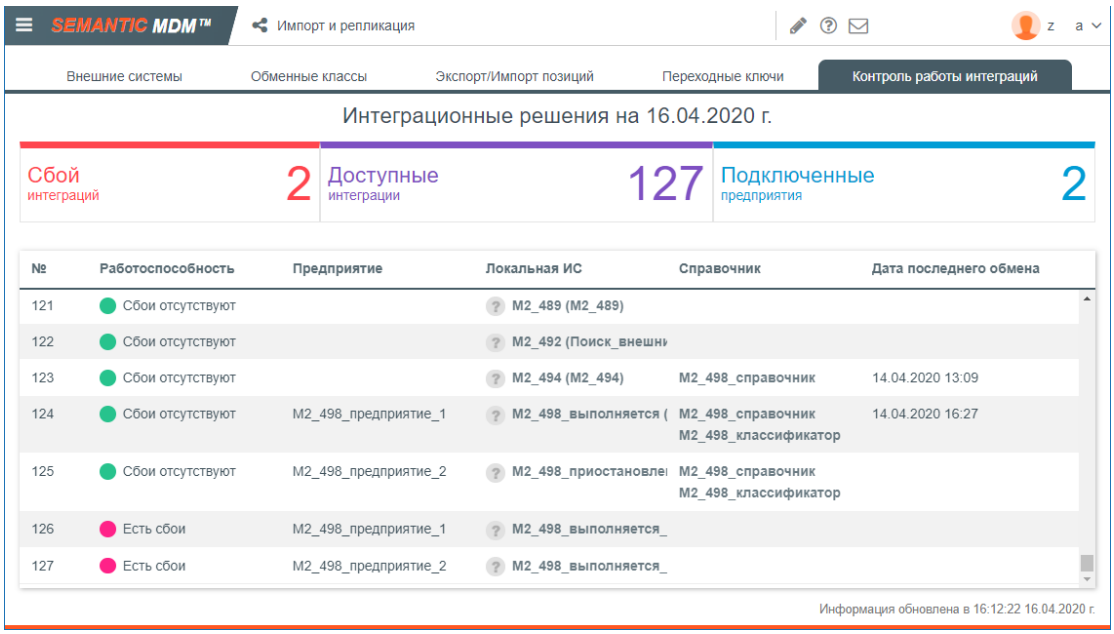


Рисунок 474 - Вкладка **Контроль работы интеграций**

На вкладке отражаются следующие показатели:

- **Сбой интеграций** [22]- отображает количество внешних информационных систем, для которых произошел сбой интеграции;
- **Доступные интеграции** [23]- количество систем с доступной интеграцией;
- **Подключенные предприятия** - количество предприятий, для которых существует хотя бы одна доступная интеграция.

В случае просрочки обработки сообщений импорта / экспорта или в случае отсутствия PING с внешней системой всем системным администраторам выдается уведомление в браузере и на электронную почту с описанием причины сбоя и ссылкой на систему в которой произошел сбой. Уведомление также дублироваться на электронную почту. Уведомление в браузере и на электронную почту выдается однократно, только после смены статуса на со l «Сбои отсутствуют» на l «Сбой интеграции».

---

[22] *Сбой интеграции - это доступная интеграция, в которой хотя бы для одного обменного класса имеются сообщения с просроченной обработкой на импорт/экспорт*

[23] *Доступная интеграция - это внешняя информационная система, для которой имеется хотя бы один связанный обменный класс со статусом "выполняется" или "приостановлен" и таким же статусом на связи обменного класса с системой*

Ниже основных показателей располагается таблица, в которой показаны доступные интеграции со следующим описанием:

- Работоспособность:
  - красный маркер - есть сбои;
  - зеленый маркер - сбои отсутствуют;
- **Предприятие** - наименование предприятия из карточки внешней системы;
- **Локальная ИС** - наименование внешней информационной системы. При щелчку левой кнопки мыши по наименованию системы, выполняется переход на вкладку **Экспорт/Импорт позиций** в выбранную систему;
- **Справочник** - список наименований справочников. По щелчку левой кнопки мыши на наименовании справочника, выполняется переход на вкладку **Экспорт/Импорт позиций** в соответствующий обменный класс для выбранной системы. Если обменных классов несколько, то, во всплывающем окне, будет показан список наименований обменных классов после выбора одного из них пользователь перейдет в выбранный обменный класс;
- **Дата последнего обмена** - дата и время последнего обработанного сообщения на импорт/экспорт. Обработанным сообщением на импорт считаются сообщения со статусами: «Обработано успешно», «Отклонено». Обработанным сообщением на экспорт считаются сообщения со статусами: «Обработано успешно», «Обработано с ошибкой».

3.30.19 Централизованное управление моделью данных

Централизованное управление моделью данных (ЦУМД) позволяет в рамках холдинговой структуры организовать двухуровневое ведение данных в *Semantic MDM*. В *Semantic MDM* на первом уровне (в управляющей компании) обеспечиваются процессы ведения:

- модели данных;
- правил проверки качества данных;
- правил дедубликации и консолидации данных;
- данных вспомогательных справочников;
- данных основных справочников.

Semantic MDM на втором уровне обеспечивает возможность создания позиций в основных справочниках с последующей их передачей в центральный Sematnic MDM, где они, в соответствии с настроенными правилами дедубликации, будут консолидированы с имеющимися позициями. При этом создание позиций в основных справочниках в Semantic MDM второго уровня будет осуществляться в рамках определенных на первом уровне модели данных, правил качества данных и данных вспомогательных справочников.

Для организации двухуровневого ведения данных в Semantic MDM необходимо:

- Развернуть *Semantic MDM* в управляющей компании и на дочерних предприятиях, включая подсистему SemanticMQ;
- Зарегистрировать в центральном *Semantic MDM* внешние системы (см. пункт 3.30.1) со способом репликации «*Репликация в дочерний Semantic MDM»* (Рисунок 475) для каждого *Semantic MDM*, установленного на дочернем предприятии и указать данные для подключения к SemanticMQ, развернутого на дочернем предприятии;

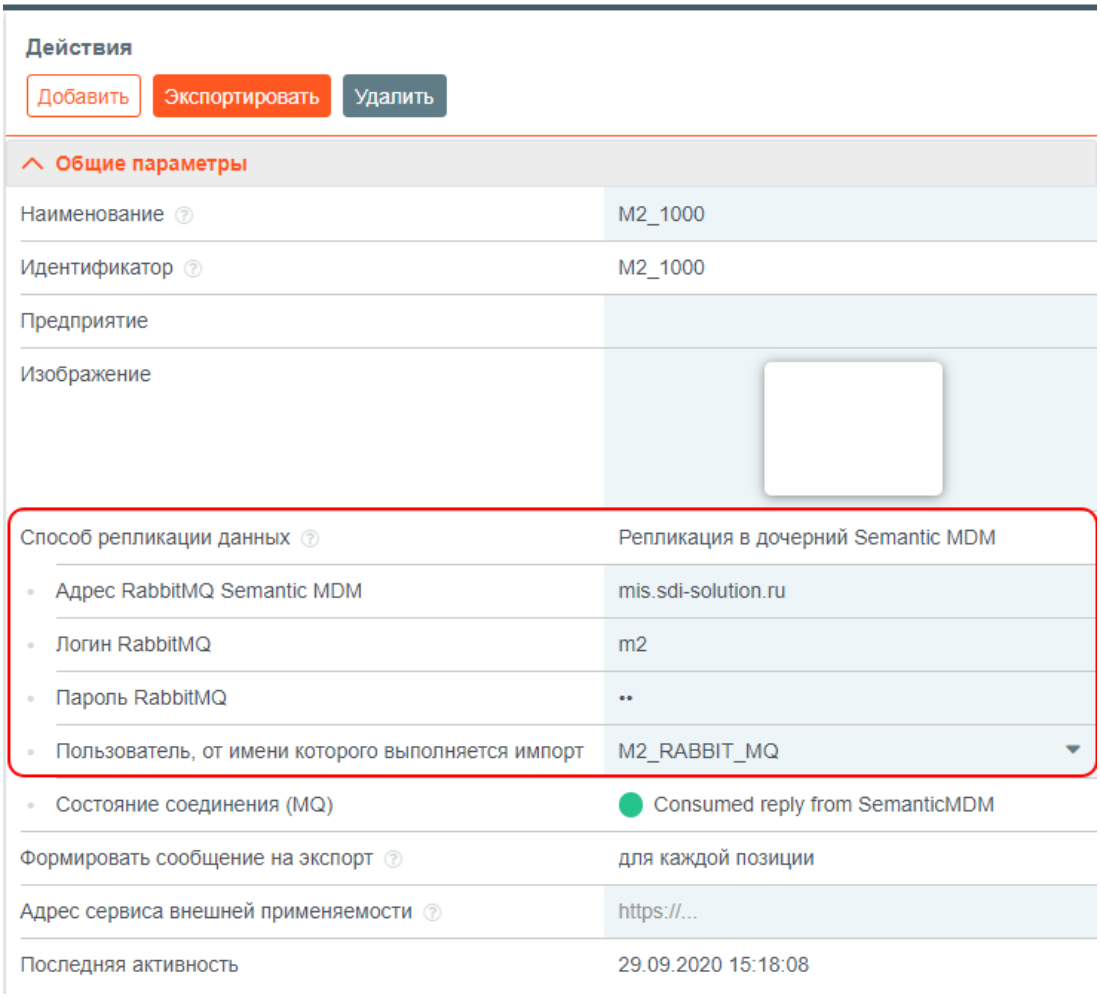


Рисунок 475 - Настройки внешней системы для репликации в дочерний *Semantic MDM*

- Создать в центральном *Semantic MDM* обменные классы с режимом экспорта/импорта «Полный для Semantic» для основных и вспомогательных справочников;
- Во всех *Semantic MDM*, развернутых на дочерних предприятиях, зарегистрировать внешнюю систему со способом репликации «*Репликация в центральный Semantic MDM*» (Рисунок 476) и указать данные для подключения к SemanticMQ, развернутого в управляющей компании;

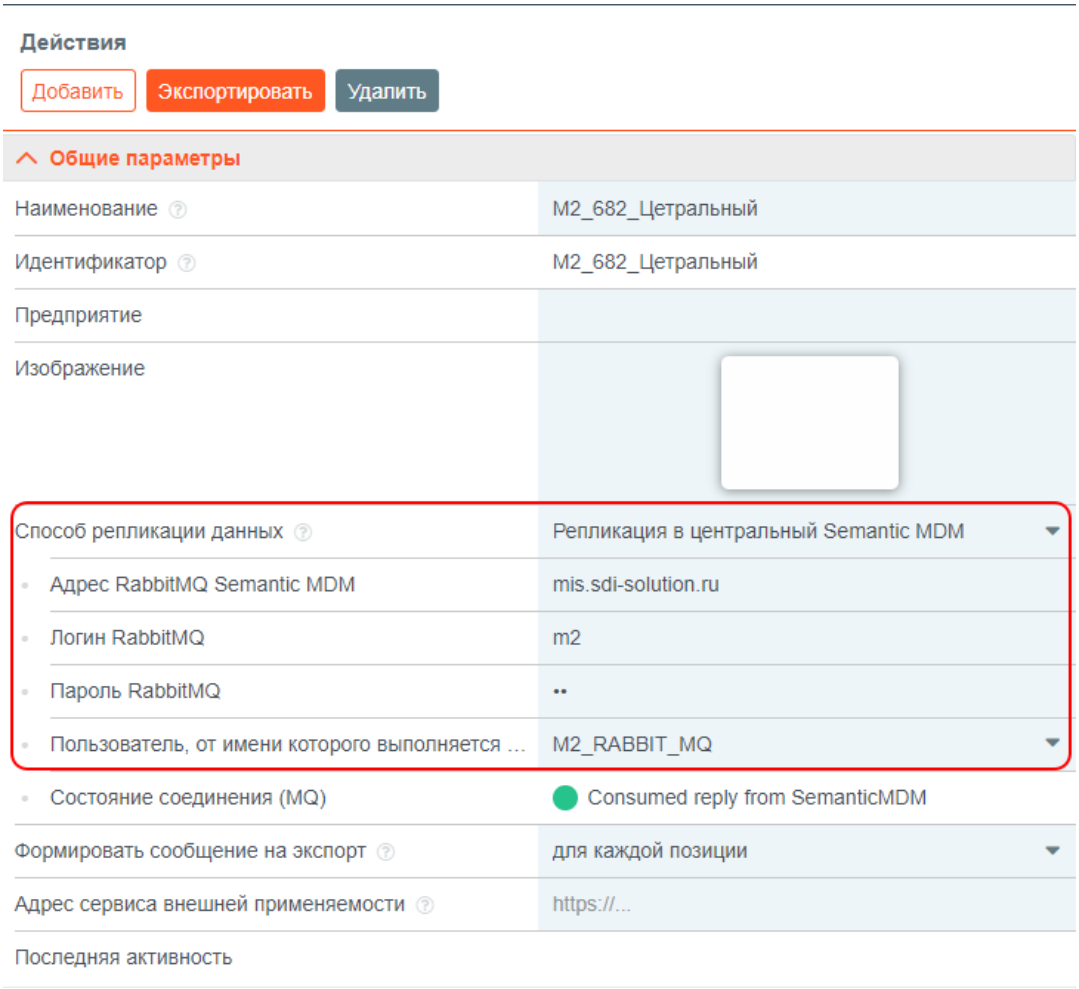


Рисунок 476 - Настройки внешней системы для репликации в центральный *Semantic MDM*

- Выгрузить в XML обменные классы, созданные в центральной MDM системе и загрузить их во все *Semantic MDM*, развернутых на дочерних предприятиях, после чего настроить их связь с внешней системой зарегистрированной для центрального Semantic MDM;
- Выполнить полную синхронизацию модели данных основных справочников, а также модели данных и данных вспомогательных справочников. Синхронизацию можно выполнить путем экспорта/импорта XML со структурой и данными основных и вспомогательных справочников .

Для мониторинга передачи пакетов с изменениями по модели данных, в подсистеме импорта и репликации Semantic MDM первого уровня, на вкладке **Экспорт/импорт позиций**, для системы со способом репликации «*Репликация*

в дочерний Semantic MDM» имеется вкладка **Экспортированные изменения структуры XML** (Рисунок 477), на которой можно увидеть какие XML-пакеты с изменениями по структуре формировались и успешность/не успешность их обработки в дочернем Semantic MDM.

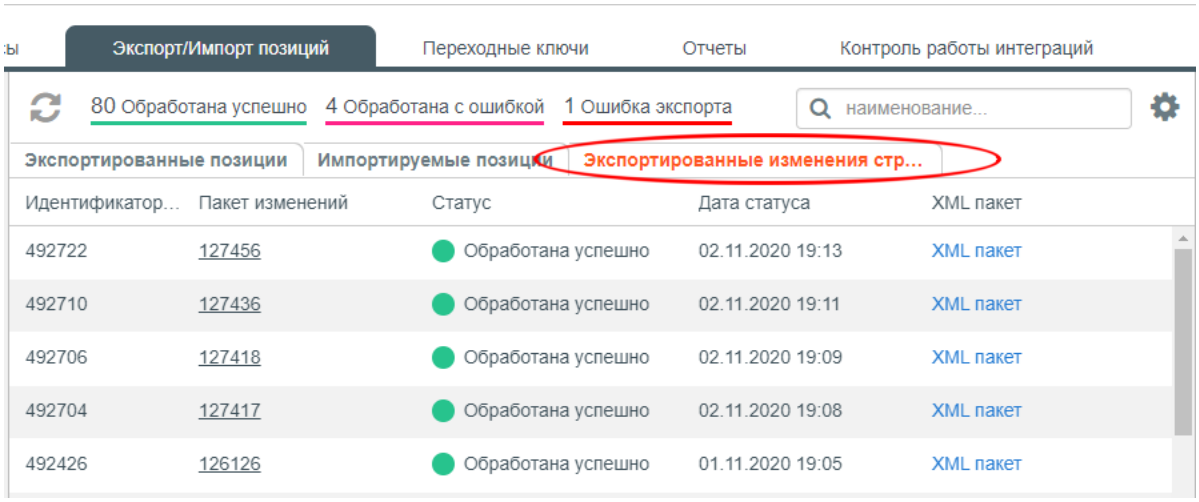


Рисунок 477 - Вкладка **Экспортированные изменения структуры XML**

Для мониторинга приема пакетов с изменениями по модели данных, в подсистеме импорта и репликации Semantic MDM второго уровня, на вкладке **Экспорт/импорт позиций**, для системы со способом репликации «*Репликация в центральный Semantic MDM*» реализовано отображение вкладки **Импортируемые изменения структуры XML**, где можно увидеть какие XML-пакеты с изменениями структуры были получены и статус их обработки.

3.30.20 Описание компонентов

Представление **Описание компонентов**, вызывается соответствующей командой через кнопку навигации и предназначено для сторонних разработчиков, которые могут здесь посмотреть какие визуальные компоненты *Semantic MDM* доступны для разработки собственных (нестандартных) форм справочников и заявок. Нестандартные формы справочников и заявок разрабатываются на Angular и регистрируются в *Системе* как Расширения (см. п. 3.31.4).

В качестве точек расширения интерфейса возможно использование (но не ограничено ими) следующих компонентов Системы:

- замена представления в правой части на странице входа пользователя в систему.
- замена стандартных кнопок управления пакетами изменений.
- добавление собственных кнопок, замены / скрытия штатных кнопок в панели кнопок, расположенной над:
  - деревом групп и над таблицей позиций в подсистеме справочников;
  - деревом классификационных групп и над таблицей позиций в иерархическом классификаторе;
  - таблицей заявок и карточкой заявки в подсистеме управления заявками (Рисунок 478);

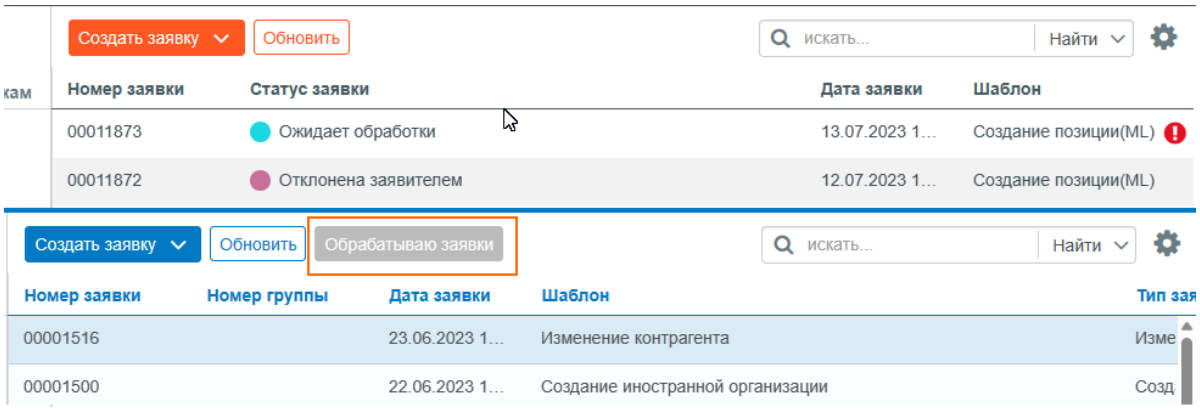


Рисунок 478 - Добавление кнопок в интерфейсе подсистемы **Журнал заявок**: вверху - стандартный интерфейс, внизу - с дополнительной кнопкой

  - деревом и таблицей в подсистеме конфигурирования данных;
- добавление собственных кнопок, замены / скрытия штатных команд в контекстном меню для:
  - дерева групп и таблицы позиций в подсистеме **Справочники**;

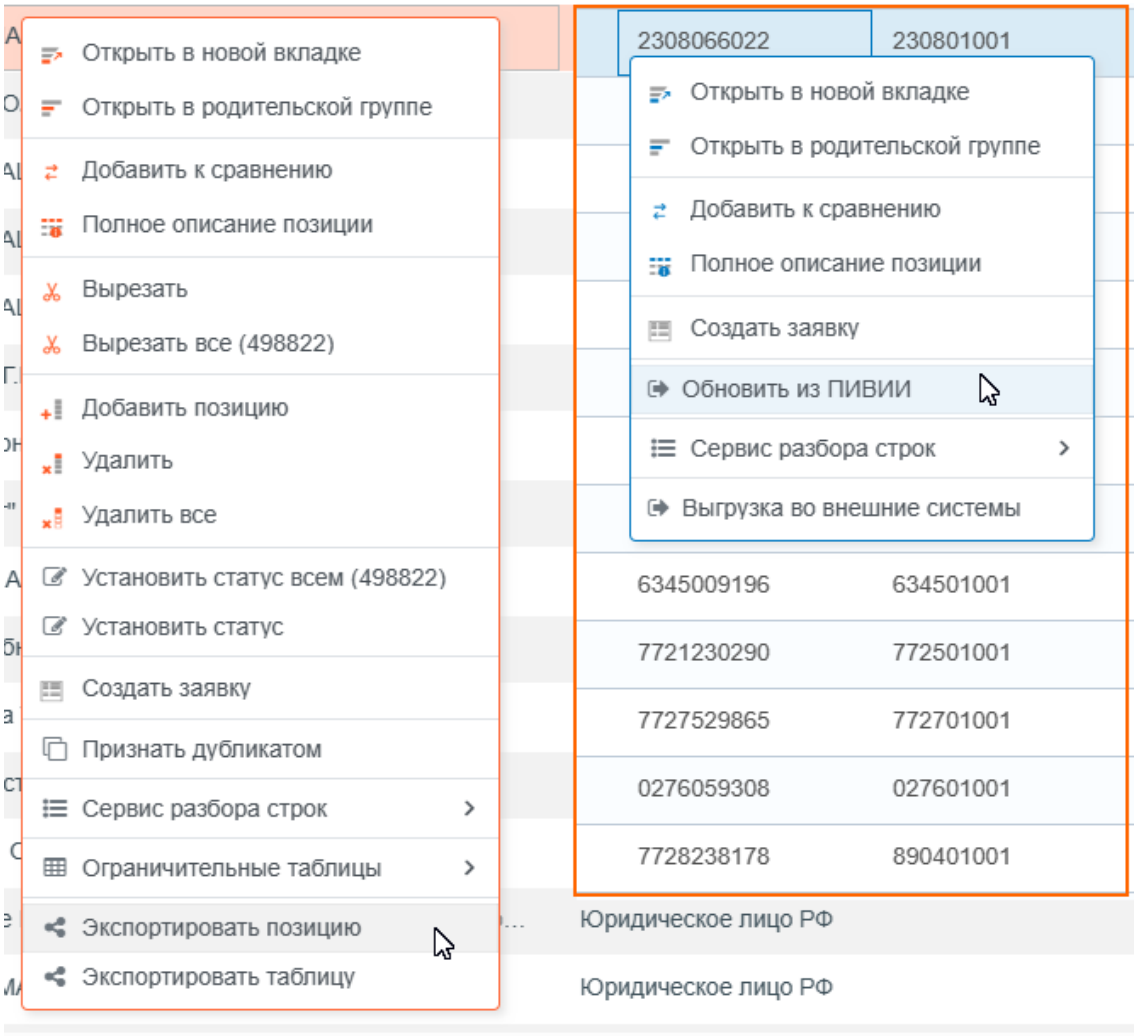


Рисунок 479 - Контекстное меню в подсистеме **Справочники**: слева - стандартное меню; справа - кастомизированное

- дерева классификационных групп и таблицы позиций в иерархическом классификаторе;
- таблицы заявок в подсистеме **Журнал заявок**;
- дерева и таблицы в подсистеме **Конфигурация данных**;
- табличного представления пользователей в подсистеме **Администрирование**;
- таблицы с парольными политиками и таблицы с причинами отклонения;

- переопределение обработчика для следующих событий:
  - выбор шаблона для создания заявки;
  - изменение значения атрибута заявки, инициализирующего заявку позицией / группой;

- нажатие на кнопку отправки заявки в обработку;
- добавление собственных и переопределения существующих панелей фильтрации заявок в подсистеме **Журнал заявок**;
- добавление собственных кнопок в раздел **Сервисные функции** в подсистеме **Настройки**;
- реализация динамического hint для значения атрибута позиции, заявки, строки ограничительной таблицы.

Раздел **Описание компонентов** содержит перечень компонентов и сервисов системы, а также три вкладки:

- **Демо** - для демонстрации работы каждого из компонентов;
- **Пример кода** - содержит пример кода для работы с выбранным компонентом или сервисом;
- **Описание API** - содержит описание API выбранного компонента или сервиса.

Для того, чтобы ознакомиться с компонентами и сервисами Semantic MDM, доступными для разработки расширений следует:

- перейти в представление **Описание компонентов**;
- установить курсор на интересующем компоненте или сервисе в списке. Для поиска интересующего объекта можно воспользоваться ячейкой поиска по наименованию, расположенной над списком. Справа от списка на вкладке **Демо** (Рисунок 480) будет показан (интерактивно) внешний вид компонента. Пользователь может перемещаться и просматривать все интересующие элементы внутри демонстрационного окна;

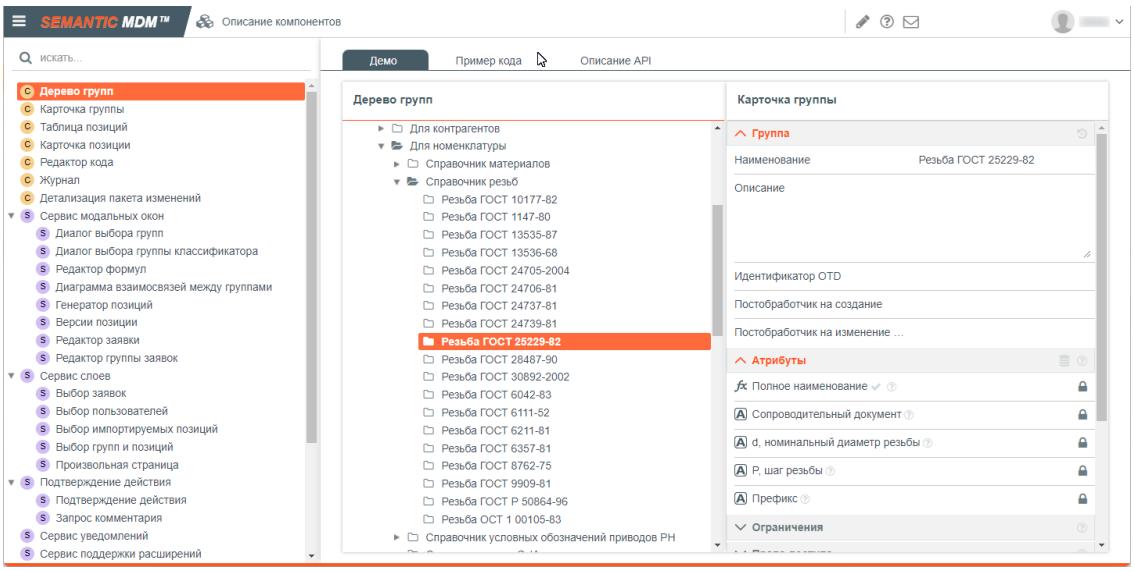


Рисунок 480 - Просмотр описания компонента *Дерево групп*

- для просмотра кода компонента или сервиса следует установить курсор на интересующем объекте и открыть вкладку **Пример кода** (Рисунок 481);

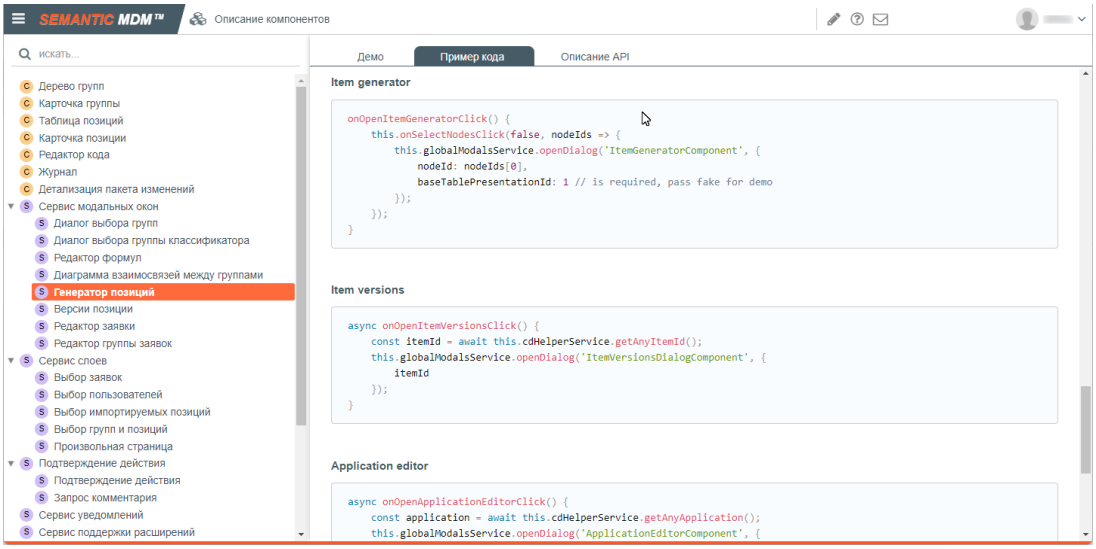


Рисунок 481 - Вкладка **Пример кода** с описанием сервиса *Генератор позиций*

- для уточнения API функций, использующихся в текущем компоненте или сервисе нужно перейти на вкладку **Описание API**[24] (Рисунок 482).

[24] *Фон вкладки **Описание API** зависит от фона, выбранного в операционной системе пользователя (в Windows устанавливается функцией Персонализация). При использовании темного фона в ОС, фон вкладки будет черным (как на рисунке внизу). Фон остальных вкладок не зависит от фона в ОС.*

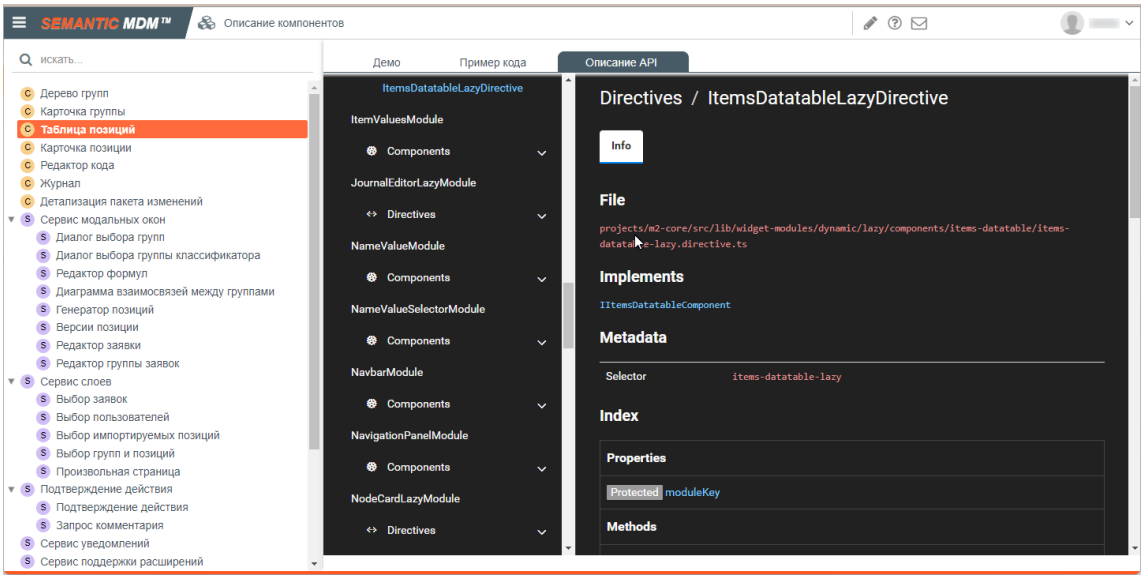


Рисунок 482 - Вкладка **Описание API** для компонента *Таблица позиций*

3.31 Настройки

В подсистеме **Настройки** (Рисунок 483) *Semantic MDM* можно определить разрешенные варианты входа пользователей в систему, а также сконфигурировать интерфейс – внешний вид и стиль окна авторизации, разделов стартовой страницы, вкладок карточки позиции и др. Для удобства работы администратора разные виды настроек распределены по разным представлениям, доступ к которым выполняется щелчком левой кнопки мыши по наименованию на панели слева.

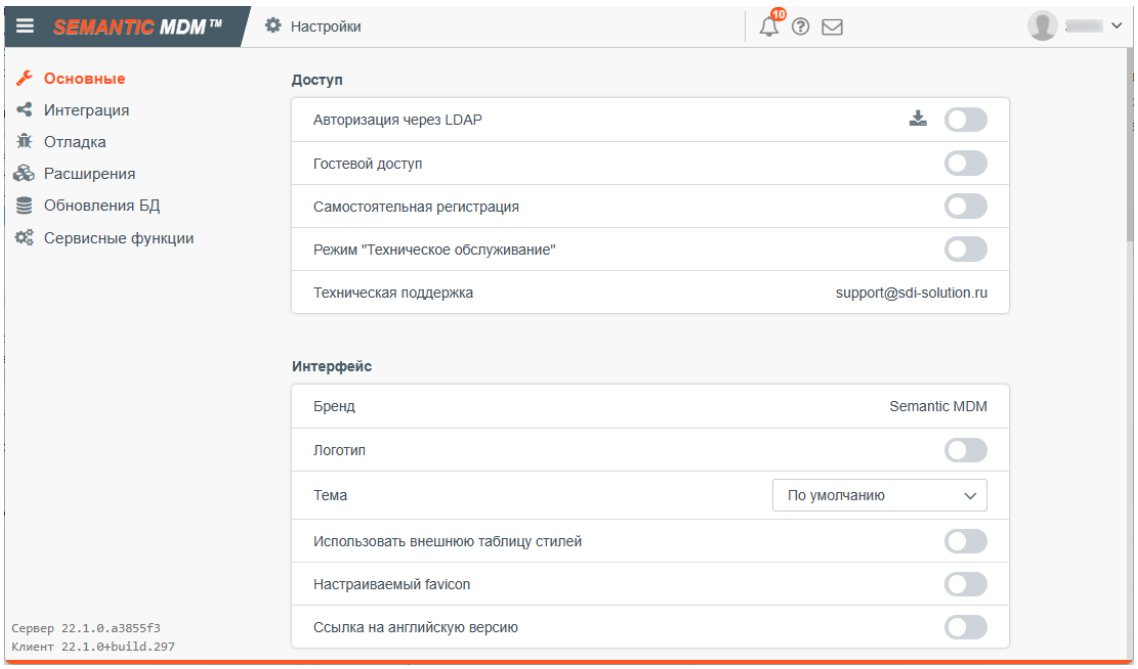


Рисунок 483 - Подсистема **Настройки** (сокращённый вид)

3.31.1 Основные

Основные настройки сгруппированы в следующие блоки:

- **Доступ** (Рисунок 484):
  - *Авторизация через LDAP* – в Системе разрешена авторизация пользователей *Windows*;

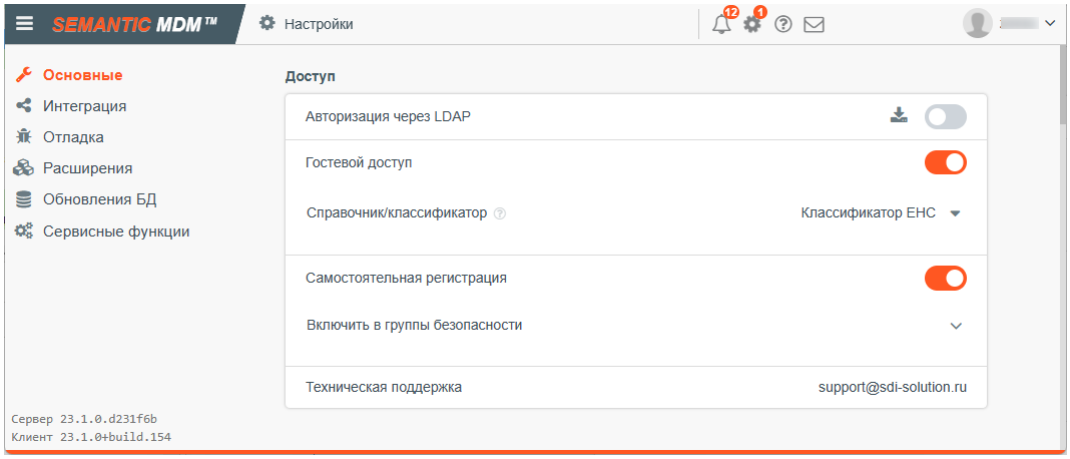


Рисунок 484 - Настройки доступа

  - *Гостевой доступ* (Рисунок 485) – при активации этой опции пользователи получают возможность войти в систему под

автоматически создаваемой временной учётной записью. При этом можно выбрать из выпадающего списка какой справочник или классификатор будет доступен пользователю с гостевым доступом. Если справочник не выбран, то будут доступны все справочники в режиме чтения (просмотра). Пользовательские настройки при этом не сохраняются;

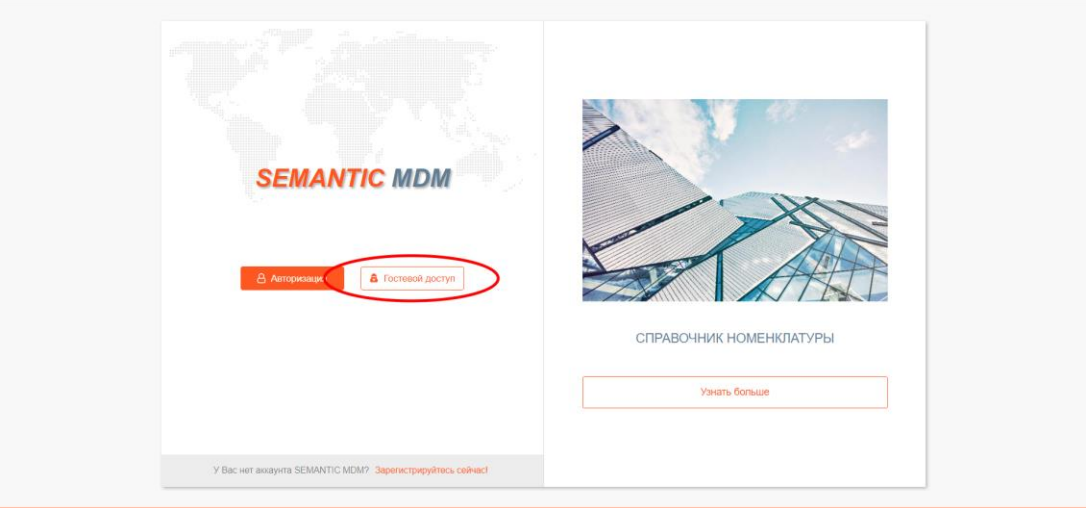


Рисунок 485 - Гостевой доступ

- Самостоятельная регистрация (Рисунок 486) – при активации настройки пользователи имеют возможность самостоятельной регистрации учётной записи и последующей её активации посредством электронной почты. При самостоятельной регистрации можно указать группы с соответствующими правами, в которые будет входить пользователь;

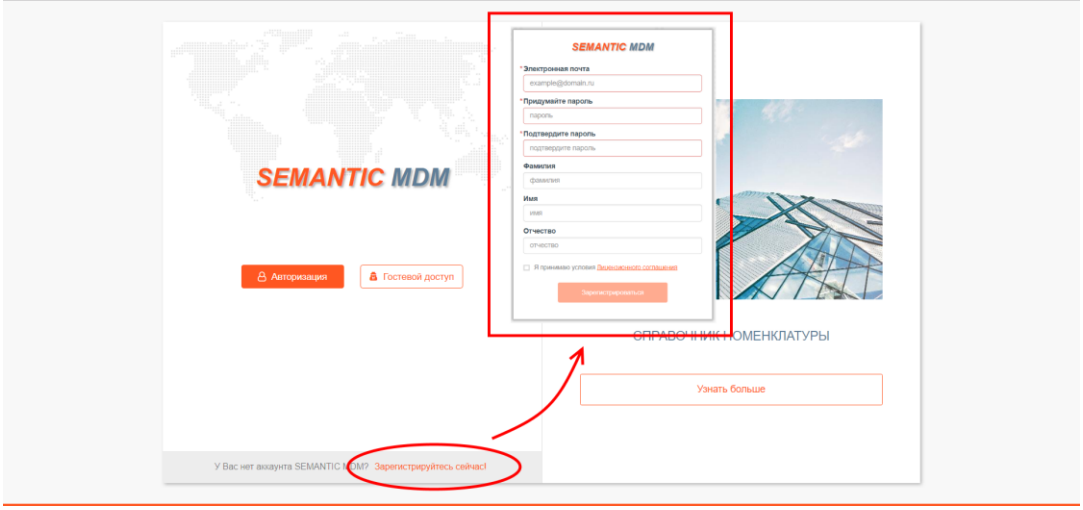


Рисунок 486 - Самостоятельная регистрация

- *Техническая поддержка* – строка с адресом электронной почты, на которую можно отправлять запросы по технической поддержке Системы;

- **Интерфейс** (Рисунок 487)**:**
  - *Бренд* – строка для ввода наименования бренда;
  - *Логотип* – позволяет загрузить рисунок с логотипом компании, который будет отображаться на стартовой странице;
  - *Тема* – набор стилей и изображений, определяющий внешний вид страницы авторизации и стартовой страницы *Системы*;

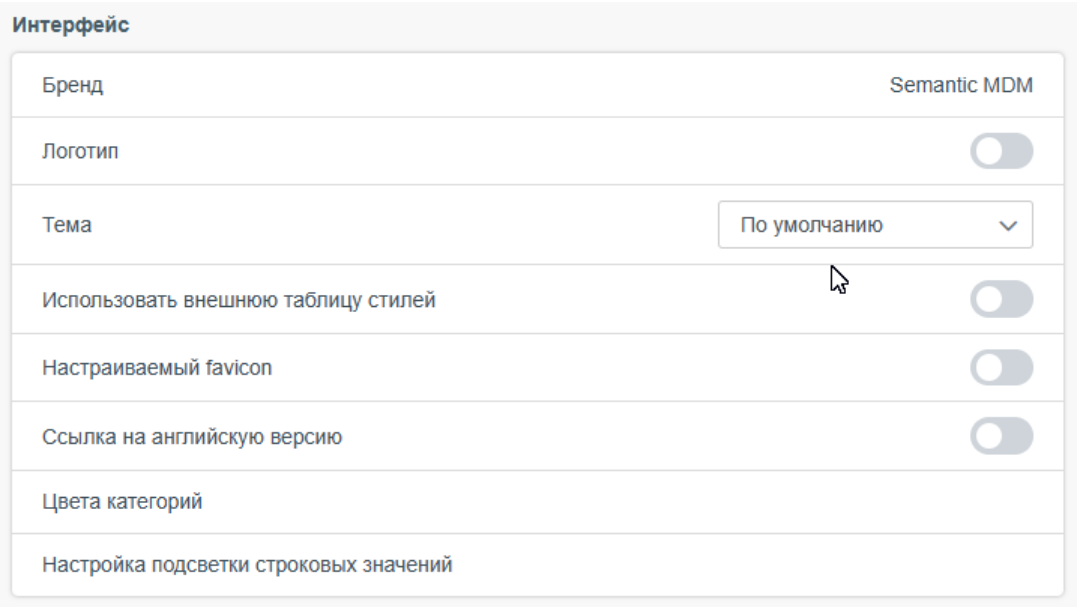


Рисунок 487 - Настройки интерфейса

- *Использовать внешнюю таблицу стилей* – в системе предусмотрена возможность подключения внешней таблицы стилей в формате *.css, в которой можно переопределить визуальное представление любого элемента (группы элементов) *Системы*;
- *Настраиваемый favicon* – значок (иконка), который отображается браузером в строке адреса перед названием страницы и в качестве картинки рядом с закладкой. Могут использоваться файлы с изображениями в форматах ICO, SVG, PNG размером 16x16 пикселей;
- *Ссылка на английскую версию* – при активации данной функции в профиле пользователя появляется команда переключения интерфейса на английскую версию (и обратно);
- *Цвета категорий* - кнопка, вызывающая окно диалога, в котором администратор может изменить ранее настроенные в справочных группах цвета категорий. Для редактирования доступны только те категории, для которых хотя бы один раз выполнялась настройка цвета на уровне справочной группы. Если настройка цвета для категории не выполнялась, то она в списке не отображается (Рисунок 488).

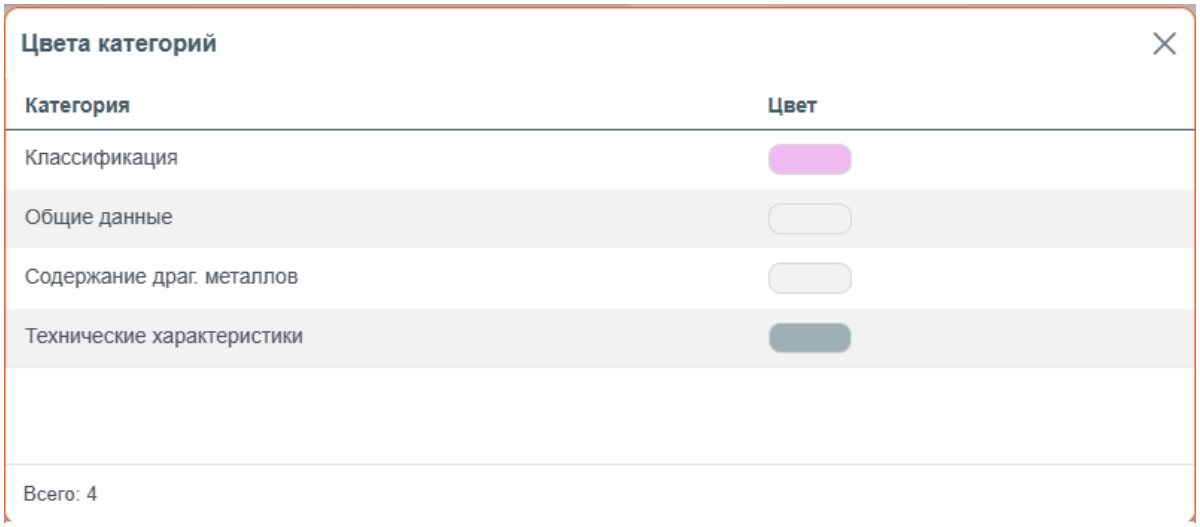


Рисунок 488 - Диалог настройки цветов категорий

Цвет назначается щелчком левой кнопки мыши по значению в колонке **Цвет**. Цвет может быть указан в открывшемся окне выбора цвета (Рисунок 489) непосредственно курсором с использованием цветовой шкалы, цветового поля и шкалы прозрачности либо введен в виде формулы Hex / RGB A / HSL A. Переключение формулы цвета выполняется кнопкой ⇕ справа от поля формулы. Кнопкой ***Ок*** необходимо подтвердить выполненные настройки. Чтобы вернуться к базовому цвету следует в окне выбора цвета нажать кнопку ***По умолчанию***.

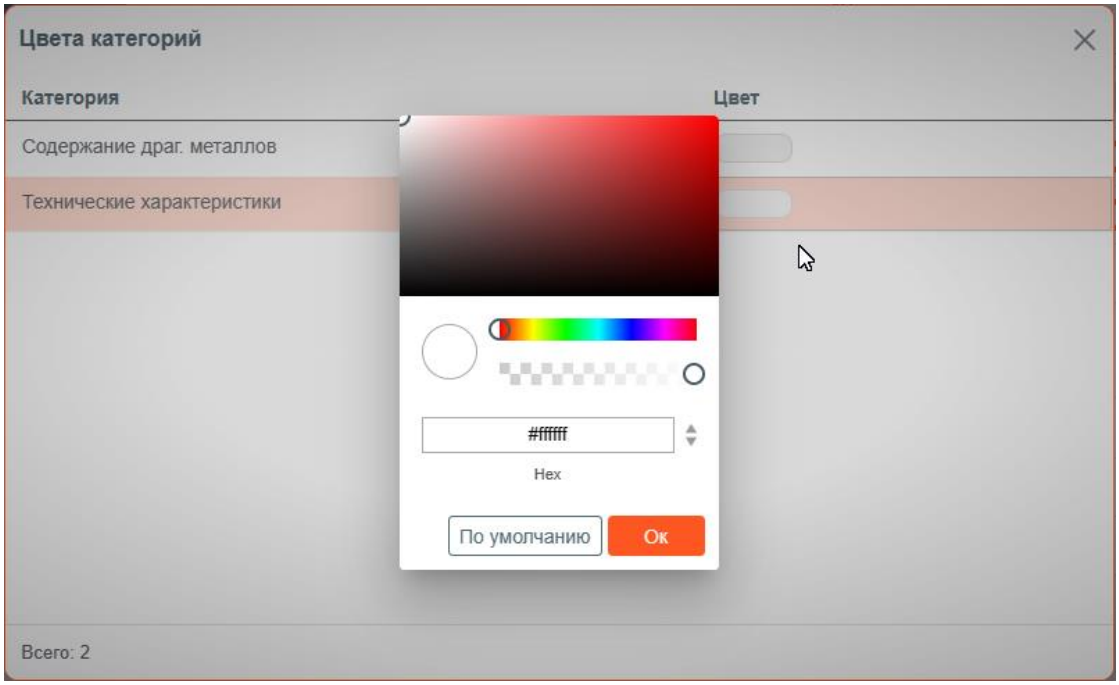


Рисунок 489 - Настройка цвета подсветки категорий

- *Настройка подсветки строковых значений* – вызывает окно диалога настройки подсветки символов в строковых и текстовых атрибутах позиций (Рисунок 490). В диалоге через регулярные выражения можно задать цвет фона, текста и начертание. *Система* назначает указанные настройки цвета и начертания всем символам, найденным через указанное регулярное выражение. В верхней части окна располагается тестовая строка, при помощи которой можно проверить работу регулярных выражений. Новая строка для регулярного выражения появляется после заполнения всех имеющихся;



Рисунок 490 - Диалог настройки подсветки символов

- **Страница авторизации:**
  - *Наименование приложения* – подпись под рисунком на странице авторизации;
  - *Настраиваемое изображение* – замена рисунка в правой части формы авторизации;
  - *Настраиваемый документ «Узнать больше»* – прикрепление документа, открывающийся при нажатии на кнопку «Узнать больше» на странице авторизации;
- **Стартовая страница:**
  - *Показывать «Все справочники»* – регулирует видимость группы «Все справочники» в разделе «Справочники» на стартовой странице;
  - *Показывать «Последние»* – регулирует видимость раздела «Последние» на стартовой странице, в котором отображаются последние посещённые пользователем группы справочников;
  - *Настраиваемое изображение плитки* – рисунок, используемый по умолчанию при создании пиктограмм справочников (Рисунок 491);

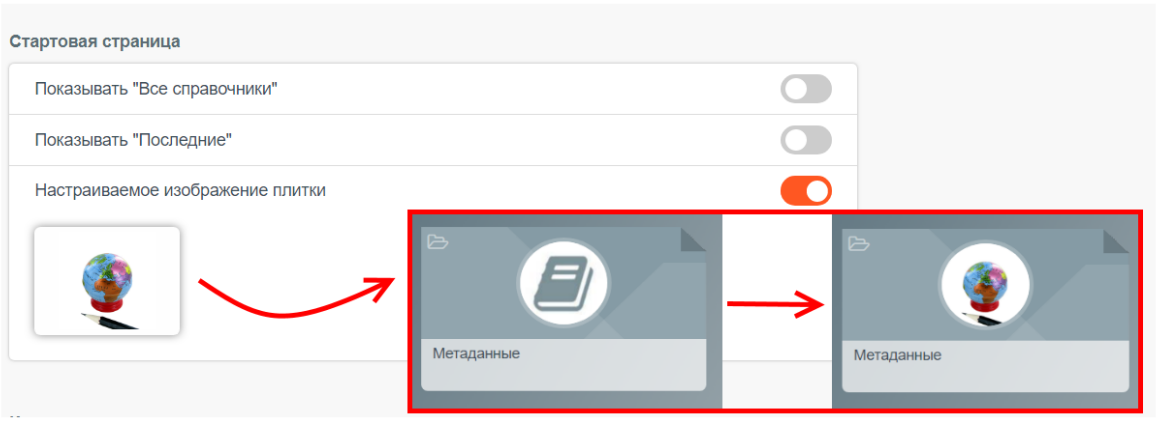


Рисунок 491 - Настраиваемое изображение по умолчанию на пиктограмме

- **Дерево и таблица позиций** (Рисунок 492)**:**
  - *Построение худого дерева* – если опция включена, то в процессе выполнения поиска (вне зависимости от положения курсора в дереве) в дереве справочников будет возможность показать все ветки, которые содержат искомую информацию. При отключенной опции поиск производится только в текущей ветке справочника;
  - *Наследование критериев поиска* – если опция включена, то критерии расширенного поиска, настроенные для родительской группы наследуются всеми дочерними группами. При выключенной опции, критерии расширенного поиска в каждой группе могут быть свои;
  - *Отображать колонку «Наименование»* – при активации данной опции во всех справочниках *Semantic MDM* на сетке данных присутствует системная колонка «*Наименование»*, которая не может быть скрыта и перемещена в таблице. При отключенной опции значение любого выбранного атрибута может быть наименованием позиции;

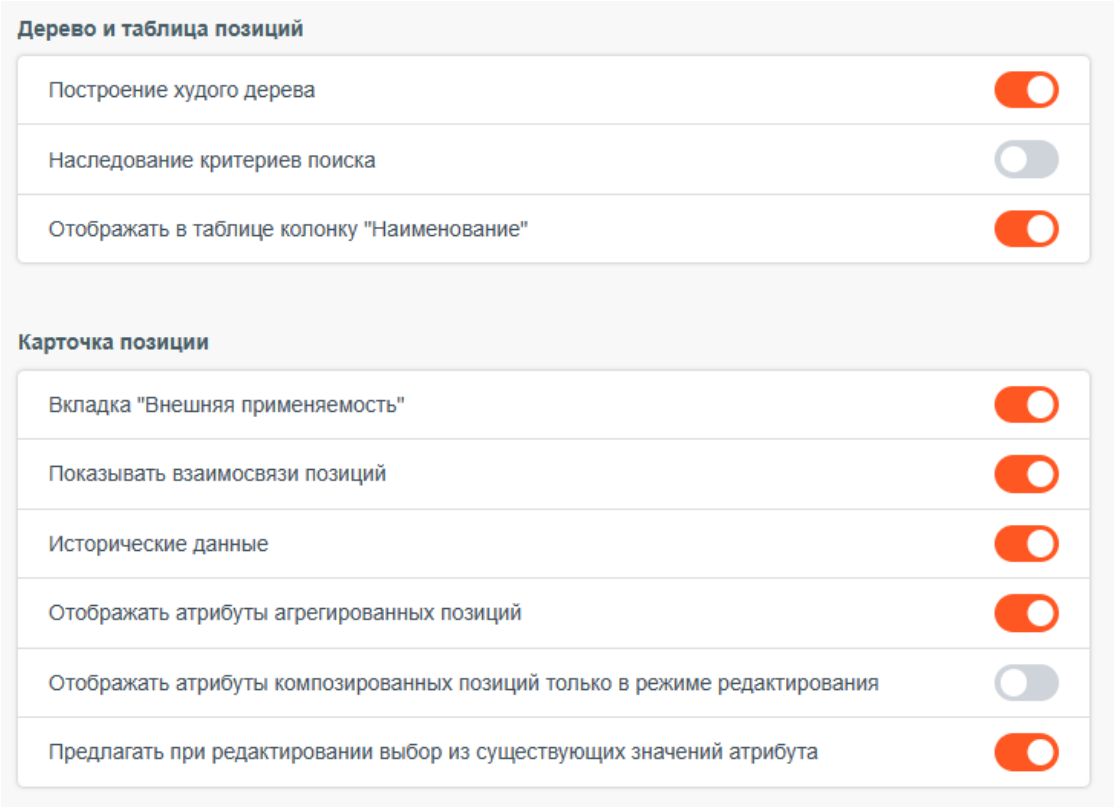


Рисунок 492 - Продолжение основных настроек

- **Карточка позиции** (Рисунок 492)**:**
  - *Вкладка «Внешняя применяемость»*[25] – настройка регулирует видимость раздела «Внешняя применяемость» в карточке позиции;
  - *Показывать взаимосвязи позиций* – настройка регулирует видимость ссылок на редактор связей для позиции;
  - *Исторические данные* - настройка разрешает работу с историческими значениями в Системе;
  - *Отображать атрибуты агрегированных позиций* – при активации данной опции в карточке позиции, содержащие агрегированные значения атрибута, можно просмотреть значения атрибутов агрегированной позиции. При отключенной опции для

[25] *Внешняя применяемость объекта доступна, если для справочника настроен web-сервис, с помощью которого передаётся информация из внешней информационной системы.*

просмотра доступно только наименование агрегированной позиции и её статус (Рисунок 493);

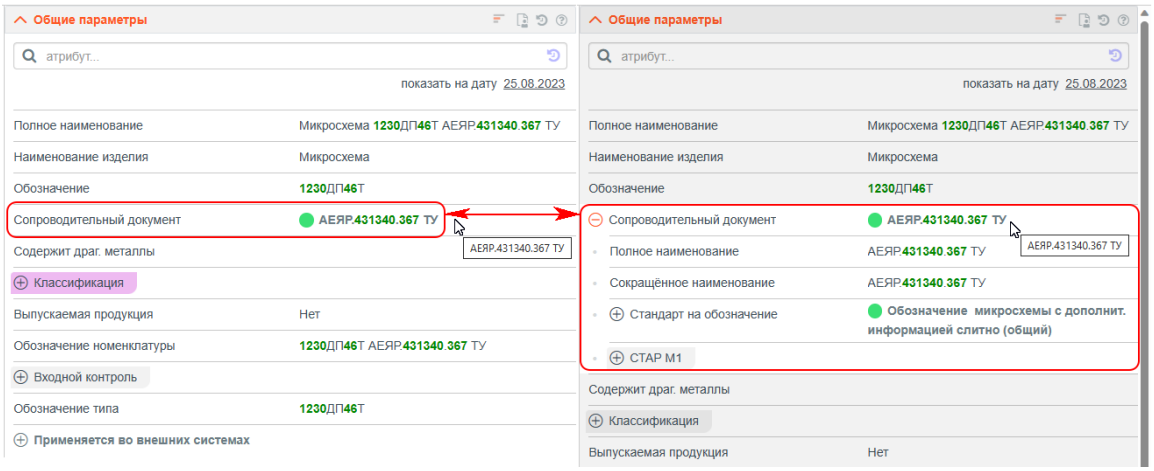


Рисунок 493 - Отображение агрегированной позиции в карточке: слева при выключенной опции *Отображать атрибуты агрегированных позиций*, справа - при включенной

- *Отображать атрибуты композированных позиций только в режиме редактирования* – при включении данной опции в режиме чтения в карточке родительской позиции доступно для просмотра только наименование композируемой позиции и её статус. При выключенной опции в карточке родительской позиции в режиме чтения отображаются все заполненные атрибуты композируемой позиции (Рисунок 494);

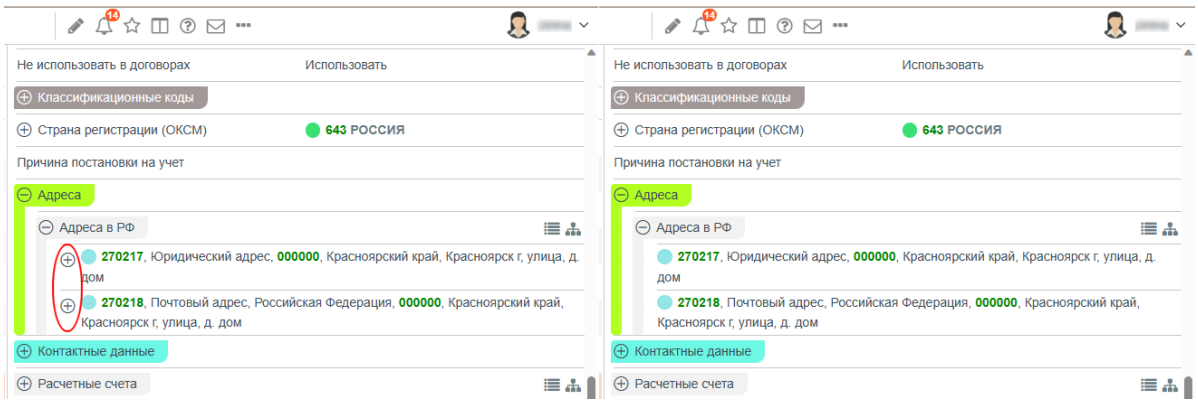


Рисунок 494 - Отображение композируемой позиции в карточке родительской позиции: слева при выключенной опции *Отображать атрибуты композированных позиций только в режиме редактирования*, справа - при включенной, в режиме чтения

- *Предлагать при редактировании выбор из существующих значений атрибута* – опция, при выключении которой пользователь сможет выбирать значение из списка только для атрибутов связи типа *Агрегация* и простых атрибутов, список значений которых определен ограничительной таблицей;

- **Лицензия:**
  - *Текущая лицензия* – открывает окно со сведениями о текущей лицензии пользователя (Рисунок 495);

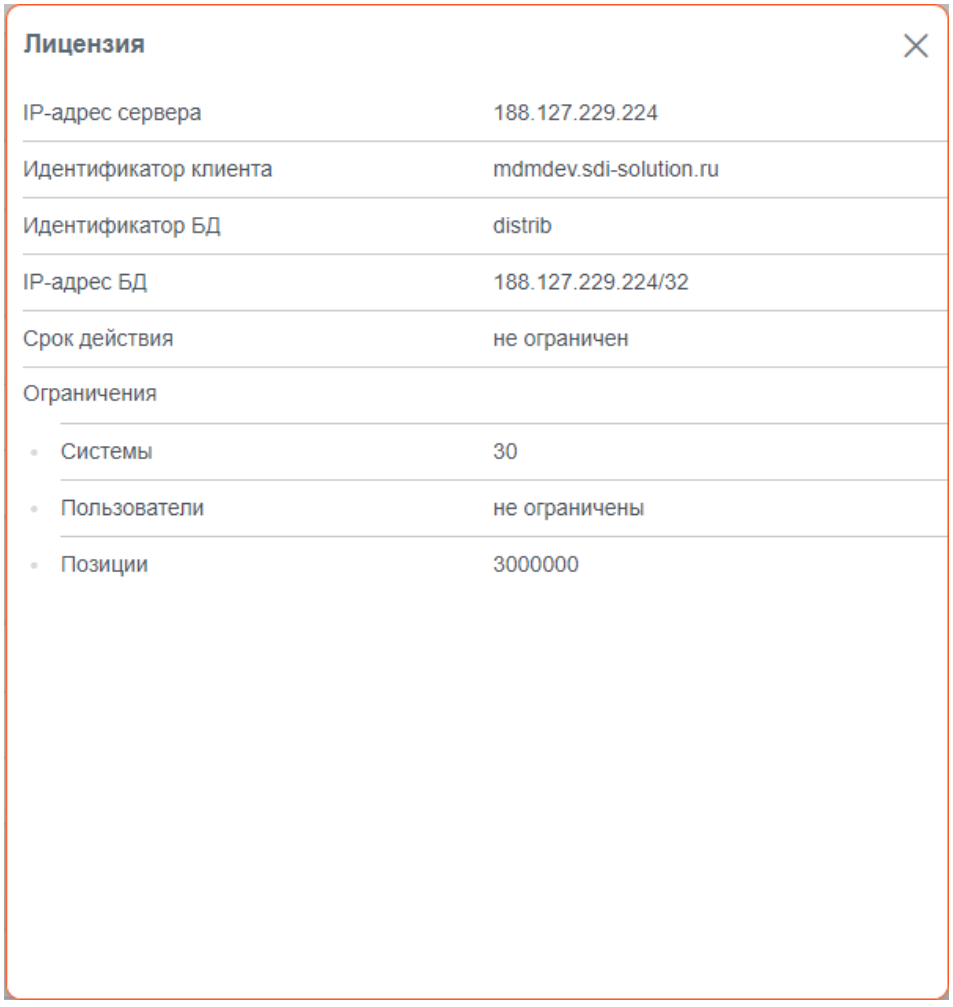


Рисунок 495 - Сведения о текущей лицензии

  - *Файл запроса лицензии* – формирует файл запроса лицензии «license-claim.json», который используется для формирования ключа ПО;

- *Загрузить файл лицензии* – открывает окно проводника для выбора файла лицензии с расширением JSON;

- **Прочее:**
  - *Показывать ссылку на Интегратор SolidWorks* – в случае активизации опции в профиле пользователя появляется ссылка на скачивание интегратора с CAD-системой SolidWorks;
  - *Звуковая нотификация* – включение / выключение звукового сопровождения ошибочных действий пользователя или выбора заблокированного элемента;

3.31.2 Интеграция

Раздел «**Интеграция**» (Рисунок 496) предназначен для настройки работы с **Сервисом разбора строк**, настройки работы веб-сервиса SemanticMQ для модуля импорта и репликации, а также сервера отчетов SemanticReports, обеспечивающего формирование отчетов.

Для **Сервиса разбора строк** необходимо указать адрес сервера, на котором он расположен. Описание работы сервиса дано в отдельном *Руководстве*. При использовании возможностей автоматической генерации позиций номенклатуры, необходимо указать в каких справочниках это будет использоваться. Для добавления нового справочника, работающего с генерацией позиций, нужно нажать на кнопку ***Выбрать группы*** и установить маркеры в открывшемся окне напротив одной или нескольких справочных групп.

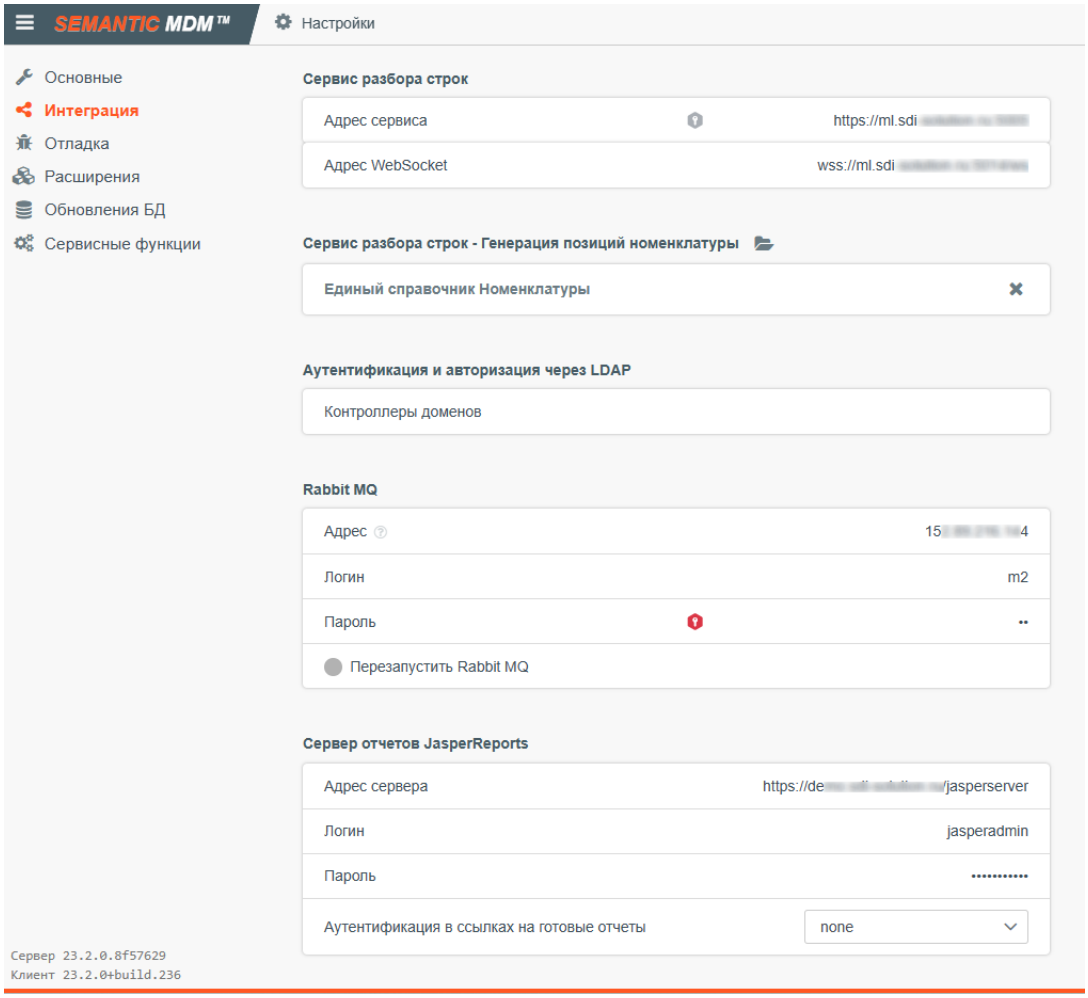


Рисунок 496 - Настройки интеграции

Для пункта **Аутентификация и авторизация через LDAP** в настройках расположена кнопка ***Контроллеры доменов*** (Рисунок 497)**,** которая позволяет в одноименном окне создать и настроить адрес контроллера домена, где расположена служба каталогов, путь поиска пользователей и домен. Новый контроллер добавляется кнопкой ***Создать***.

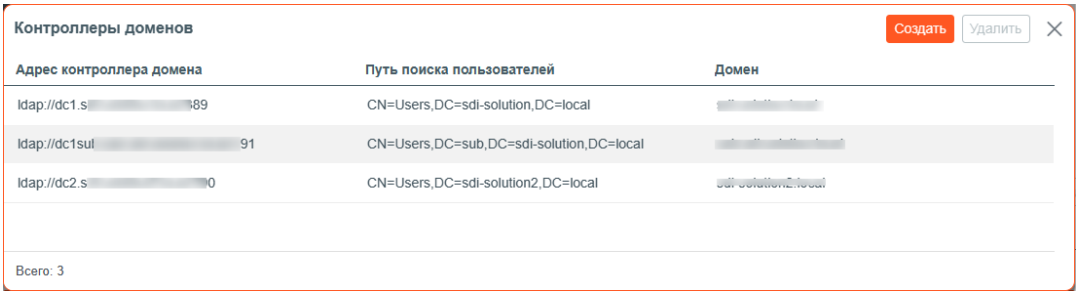


Рисунок 497 - Настройка контроллера доменов для авторизации через LDAP

В разделах **SemanticMQ** и **Сервер отчетов SemanticReports** выполняются настройки адреса сервера, где расположен соответствующий сервис, логин и пароль системного пользователя, от имени которого выполняется доступ в эти сервисы.

В разделе **SemanticMQ** присутствует кнопка ***Перезапустить SemanticMQ***, обеспечивающая включение / перезапуск сервиса после его остановки.

Для сервера отчетов **SemanticReports** в поле *Аутентификация в ссылках на готовые отчеты* доступны следующие варианты, определяющие способ передачи логина и пароля в сервис SemanticReports:

- urlParams - https://domain?j_username=login&j_password=password;
- embedded - https://login:password@domain;
- none - логин и пароль не передаются (например, для сервисов, не требующих авторизации).

3.31.3 Отладка

В разделе **Отладка** (Рисунок 498) имеется возможность настроить уровень логирования клиентского приложения. Вариант выбирается из выпадающего списка.

Включение опции **Логирование WebSocket** позволяет службе технической поддержки получить дополнительные данные о работе *Системы* у пользователя.

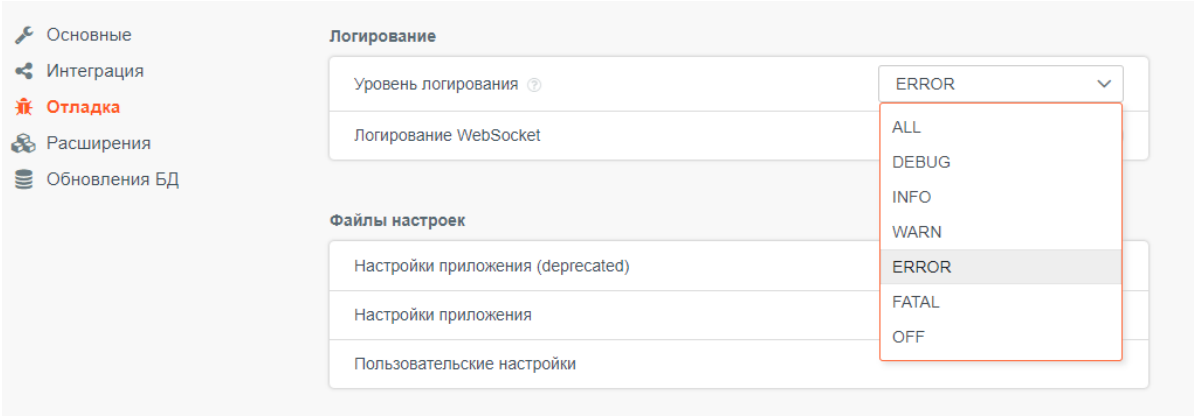


Рисунок 498 - Вид раздела **Отладка**

В этом же разделе располагаются кнопки доступа к сведениям о настройках приложения в целом и настройках текущего пользователя. Кнопка Настройки приложения (deprecated) хранит настройки предыдущей версии *Системы* (до v.20.2) (Рисунок 499).

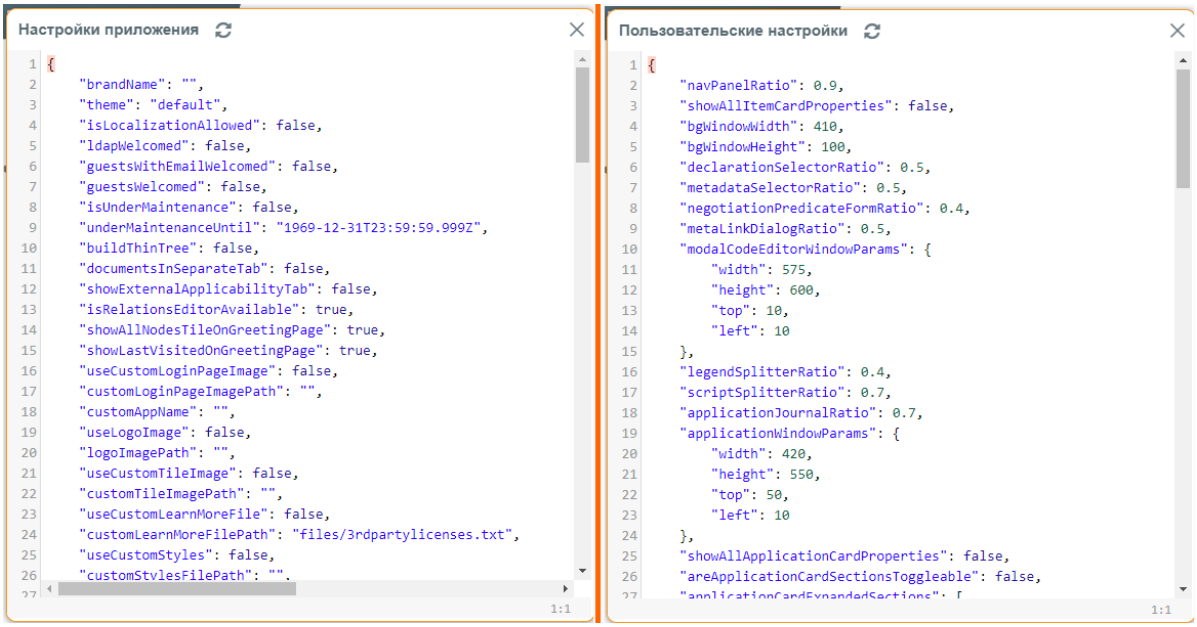


Рисунок 499 - Доступ к сведениям о настройке приложения и пользовательских настройках

3.31.4 Расширения

В разделе **Расширения** (Рисунок 500) можно подключить дополнительные расширения, необходимые для настройки и функционирования *Системы*.

В *Системе* присутствуют два встроенных расширения - **Качество данных** и **Машинное обучение**. Для включения / выключения данных расширений, которые представляют собой отдельные представления в *Semantic MDM* и доступны через кнопку ***Навигация***, достаточно сдвинуть переключатель в ячейке с наименованием расширения влево (выключено) или вправо (подключено).

Интерфейс и функциональность веб-клиента можно расширять за счёт разработки собственных модулей (расширений) на *Angular* (JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений):

- собственных интерфейсов справочников и шаблонов заявок – позволяет для зарегистрированного или пользовательского справочника / шаблона заявки в настройках указать расширение, в котором реализован интерфейс для работы с этим справочником / шаблоном заявки;
- собственных кнопок – позволяет подключать собственные кнопки в инструментальную панель над деревом групп, таблицей позиций, в меню рядом с профилем пользователя, а также в выпадающем меню **Навигация**, расположенном в левой верхней части инструментальной панели основного экрана;
- дополнительных пунктов контекстных меню дерева групп и таблицы позиций;
- автоматических обработчиков заявок.

При разработке собственных интерфейсов справочников можно использовать готовую библиотеку компонентов (см. п. 3.30.20) для работы с атрибутами, позициями и группами, что позволит ускорить разработку новых интерфейсов справочников. Подробнее о создании расширений можно узнать в отдельном *Руководстве* (Рисунок 501) (вызывается кнопкой ***Справка*** в строке **Расширения клиента**).

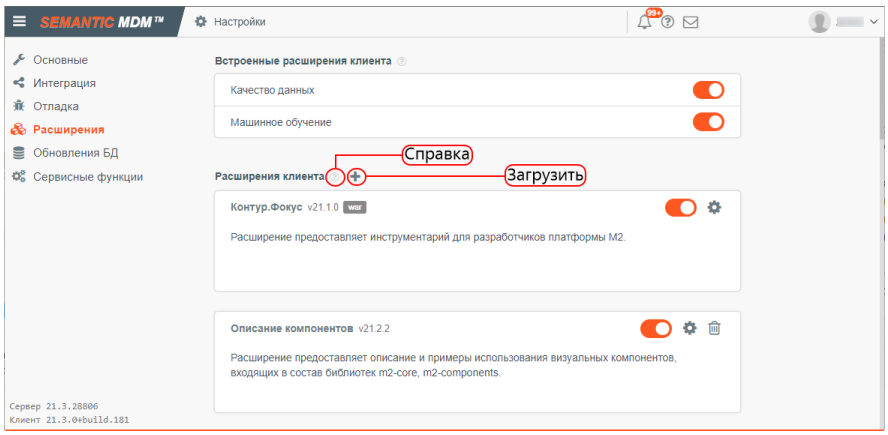


Рисунок 500 - Раздел для подключения служебных расширений

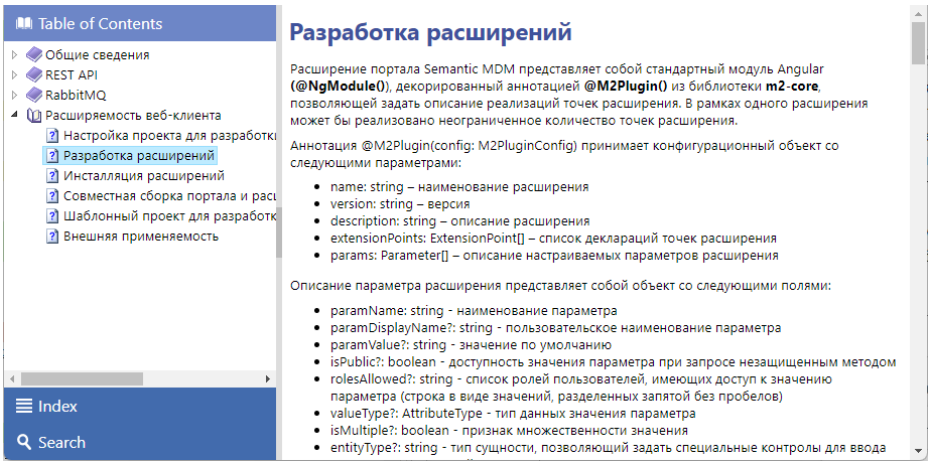


Рисунок 501 - Справка по API *Semantic MDM* и разработке расширений

Любые расширения подключаются через загрузку в *Систему* соответствующего модуля. Для этого нужно нажать кнопку ***Загрузить*** в строке **Расширения клиента** и указать в *Проводнике* файл с требуемым расширением. Наличие настроек у подключаемого расширения определяется разработчиком.

3.31.4.1 Примеры используемых расширений

3.31.4.1.1 Контур.Фокус

Расширение **Контур. Фокус** предназначено для настройки работы *Semantic MDM* с системой *Контур.Фокус*. В настройках интеграции с *Контур.Фокус*, определяются следующие параметры (Рисунок 502):

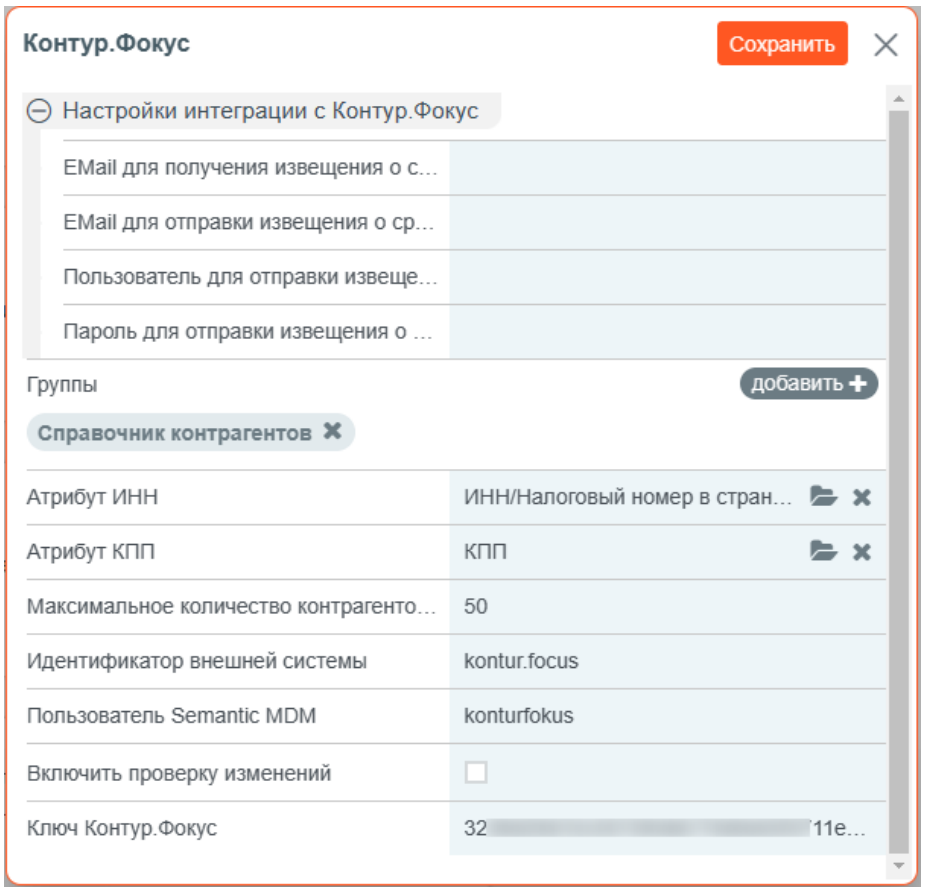


Рисунок 502 - Настройки интеграции с *Контур.Фокус*

- **Настройки интеграции с Контур.Фокус:**
  - *EMail для получения извещений о сроках лицензии* - адрес электронной почты, на которую будут приходит сообщения об окончании срока действия ключа *Контур.Фокус* за три, два и за один месяц до окончания срока лицензии либо за 5 тысяч, 3 тысячи и 1 тысячу до окончания количества запросов;
  - *EMail для отправки извещений о сроках лицензии* - адрес электронной почты, с которой будут рассылаться сообщения об окончании срока действия ключа *Контур.Фокус*;
  - *Пользователь для отправки извещений о сроках лицензии* - пользователь электронной почты, с которой рассылаются сообщения об окончании лицензии;
  - *Пароль для отправки извещений о сроках лицензии* – пароль от электронной почты для рассылки сообщений.

- **Группы** - наименования справочных групп, в которые загружаются данные из *Контур.Фокус.* Выбираются из списка справочных групп во всплывающем окне, вызываемом кнопкой ***Добавить***;
- **Атрибут ИНН** – имя атрибута справочника из Классификатора атрибутов, куда будет записываться значение ИНН;
- **Атрибут КПП** – имя атрибута справочника из Классификатора атрибутов, куда будет записываться значение КПП;
- **Максимальное количество контрагентов, которых можно консолидировать за раз** – ограничивает максимальное количество записей по контрагентам, которые можно выделить в справочнике при вызове из контекстного меню таблицы команды **Консолидировать с контрагентом из ЕГРЮЛ / Обновить [контрагента] из ЕГРЮЛ**.
- **Идентификатор внешней системы** – обозначение системы *Контур.Фокус*, использованное при ее регистрации в **Модуле импорта и репликации**.
- **Пользователь Semantic MDM** – логин пользователя *Системы*, от имени которого выполняется обращение к сервису *Контур.Фокус*;
- **Количество дней для запроса изменений (не более 7)** – количество дней, за которые можно получить изменения по контрагентам из сервиса *Контур.Фокус*;
- **Расписание проверок** – расписание в формате CRON;
- **Включить проверку изменений** – опция, включающая проверку изменений по контрагентам в соответствии с настроенным расписанием. Если опция не установлена, то проверка изменений возможна только вручную
- **Ключ Контур.Фокус** - номер лицензионного ключа, предоставляемого при покупке *Контур.Фокус*.

3.31.4.1.2 Автоматическая обработка заявок

Автоматические обработчики заявки - пункт настройки, в котором возможно подключение специальных обработчиков заявок. Штатно в *Систему* включается стандартный обработчик, который срабатывает на событие изменения статуса заявки с «Зарегистрирована» на «Ожидает обработки» / «Ожидает диспетчеризации» и выполняет следующие действия:

1. Получает из атрибута заявки, заполняемого по умолчанию ссылкой на группу, и проверяет, что группа является конечной. Если группа является конечной, то переходит на следующий шаг, в противном случае работа обработчика завершается.
2. Выполняет имеющиеся поиски дубликатов в конечной группе по атрибутам позиции, заданным в заявке. Если ни одной позиции не нашлось, то переходит на следующий шаг, в противном случае работа обработчика завершается.
3. Проверяет в заявке корректность заполнения ссылочных атрибутов типа *Агрегация*. Все значения должны быть ссылками и соответствовать scope агрегации. Если значения корректны, то переходит на следующий шаг, в противном случае работа обработчика завершается..
4. Создает связанный с заявкой пакет изменений, создает новую позицию, заполняет ее атрибуты из заявки и устанавливает статус «Нормализована».
5. Проверяет позицию на соответствие всем обязательным ограничениям и если она им соответствует, то отправляет заявку на согласование/публикацию, в противном случае, отменяет пакет изменений, а заявку переводит в статус «Ожидает обработки».

Кроме описанного выше стандартного сценария обработки, могут быть реализованы и более сложные сценарии, в которых может выполняться обработка любых переходов заявки из одного статуса в другой.

Реализация таких обработчиков выполняется в виде отдельных сборок (war), которые при загрузке на сервер приложений Wildfly, выполняют свою регистрацию и создают параметры для настройки обработчика. Минимальный набор параметров обработчика должен содержать (Рисунок 503):

- url – адрес сервера приложений Wildfly Semantic MDM;
- Наименование модуля – данное наименование будет видно при выборе обработчика в шаблоне заявки;
- Пароль – пароль пользователя, от имени которого будут выполняться все действия в системе;
- Пользователь – имя пользователя, от которого будут выполняться все действия в системе.

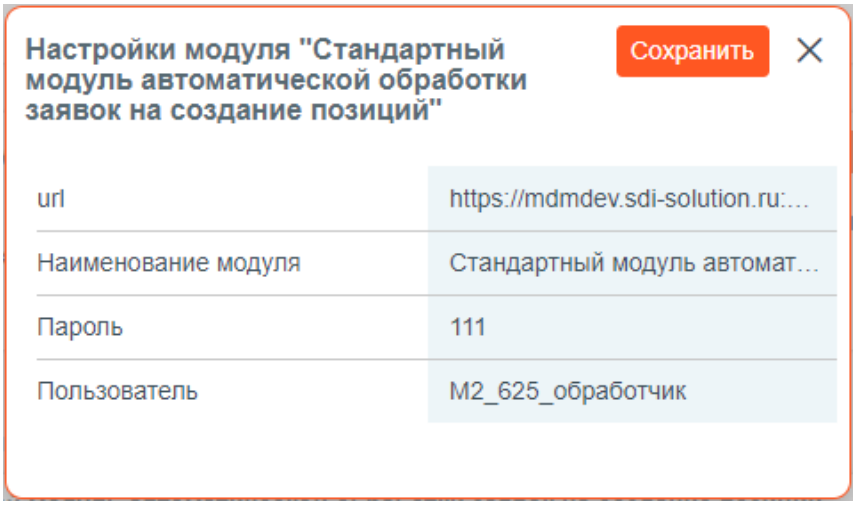


Рисунок 503 - Настройки стандартного автоматического обработчика заявок

3.31.5 Обновления БД

В данном разделе отображается таблица со сведениями о проведенных обновлениях БД (Рисунок 504). В таблице отображаются следующие данные об обновлениях:

- **Статус** – статус обновления:
  - *Успешное* – обновление БД произведено успешно и без ошибок;

- *Некритичная ошибка* – при выполнении обновления БД произошла ошибка, не влияющая на работоспособность *Semantic MDM*;
- *Критичная ошибка* – при выполнении обновления БД произошла ошибка, делающая *Semantic MDM* неработоспособной;

- **Описание** – описание выполненного обновления и его номер;
- **Дата выполнения** – дата и время, в которое было произведено обновление БД;
- **Важность** – степень важности обновления:
  - *Рекомендуемое* - необязательное обновление, установка которого может быть отложена, например, это чистка БД от ненужных более таблиц, добавление функционала;
  - *Важное* - внесения изменений, влияющих на стабильную работу *Системы*. Без установки данного обновления *Semantic MDM* работать будет, но появятся сообщения об ошибках;
  - *Обновление в ручную* - позволяет сократить время обновления и запускать не критичные, но длительные по времени блоки обновления вручную;
  - *Критичное* - без установки такого обновления работоспособность *Системы* в целом не гарантируется;
- **Детализация ошибки** – описание ошибки, в случае если обновление получило статусы *Некритичная ошибка* или *Критичная ошибка*[26].

[26] *Если важное или критичное обновление получило статус Некритичная ошибка или Критичная ошибка, то раз в 30 сек всем пользователям рассылается предупреждение об ошибке в Системе.*

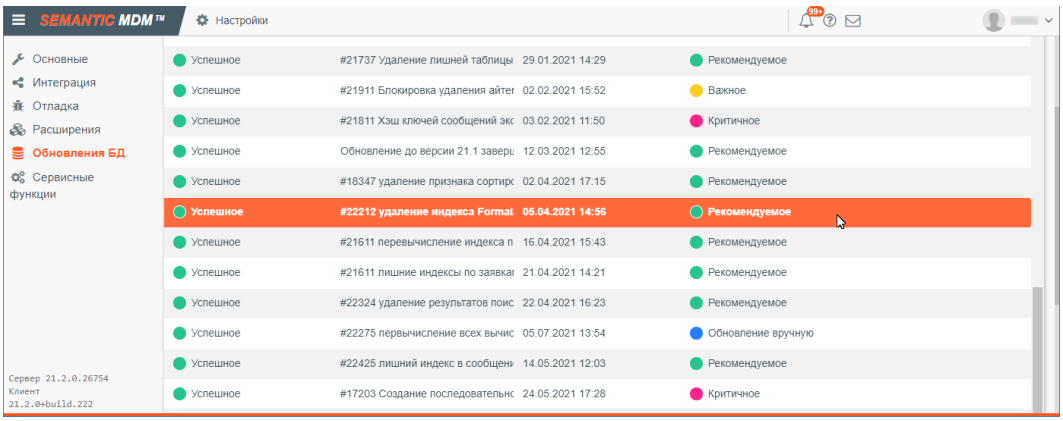


Рисунок 504 - Сведения об обновлениях БД

В таблице возможен быстрый поиск, а также сортировка по любому из столбцов. Для столбцов **Статус** и **Важность** имеется фасетный поиск.

3.31.6 Сервисные функции

Раздел предназначен для выполнения следующих настроек (Рисунок 505):

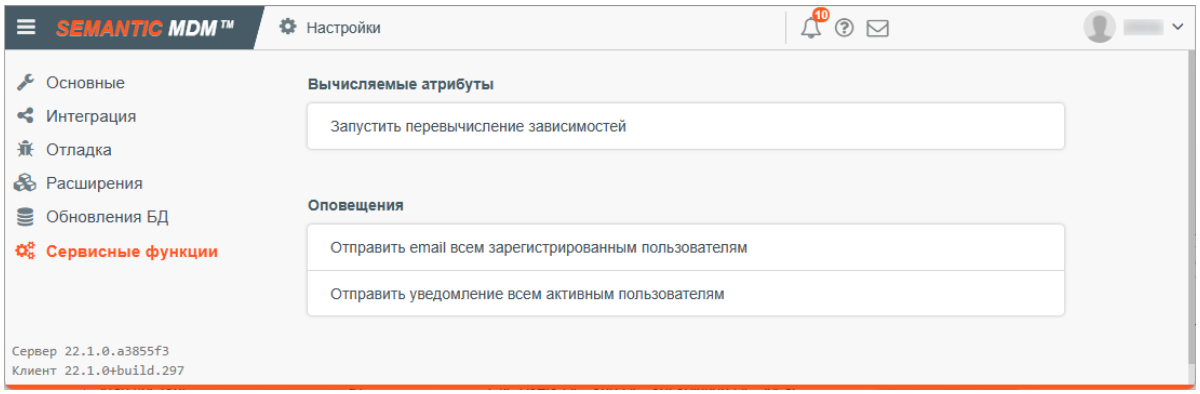


Рисунок 505 - Настройки сервисных функций

- **Запустить перевычисление зависимостей** – кнопка, которая запускает после подтверждения пользователем перевычисление всех вычисляемых атрибутов после сбоя в Системе;
- **Отправить email всем зарегистрированным пользователям** – кнопка открывает окно для ввода сообщения, которое будет отправлено его всем зарегистрированным пользователям, для которых указан email (Рисунок 506). Используется для отправки служебных

сообщений, например о техническом обслуживании или необходимости изменить пароль в *Системе*;

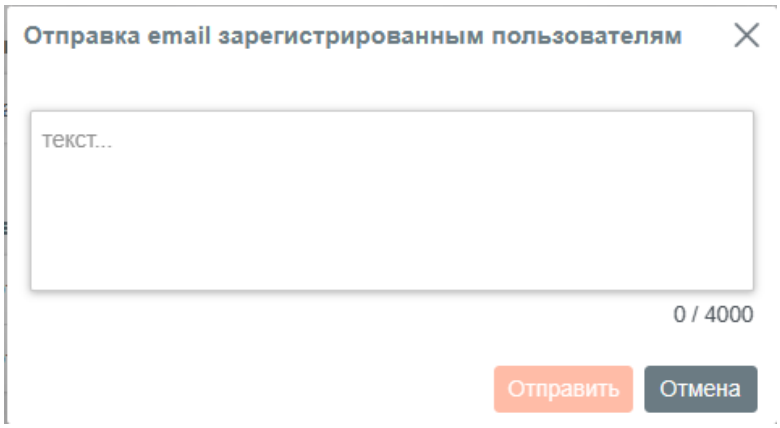


Рисунок 506 - Окно для ввода сообщения

- **Отправить уведомление всем активным пользователям** – кнопка открывает окно для ввода сообщения, которое будет отправлено его всем активным в Системе пользователям, у которых указан email. Используется для отправки служебных сообщений, например о сбое в системе или скором её отключения для технических работ ;

3.31.7 Пользовательские отчеты SemanticReports

Пользовательские отчеты должны располагаться в папке **/public/Reports/User** в репозитории сервера *SemanticReports* (Рисунок 507). Внутри данной папки допускается формирование иерархии неограниченного уровня вложенности. Наименование категории отчета определяется наименованием его родительской папки. Считается, что отчеты, расположенные непосредственно в папке **/public/Reports/User**, категории не имеют.

Предусмотрены 2 системные папки отчетов:

- **/public/Reports/User/Отчеты_по_заявкам**
- **/public/Reports/User /Отчеты_по_импорту_и_репликации**

Отчеты в данных папках доступны на вкладках **Отчеты** в в подсистемах

Журнал заявок и **Импорт и репликация** на вкладках **Отчеты** и в подсистеме **Отчеты**.

В пункте 3.34 приведены описания специальных view в БД системы, используемых для разработки шаблонов отчетов и их формирования.

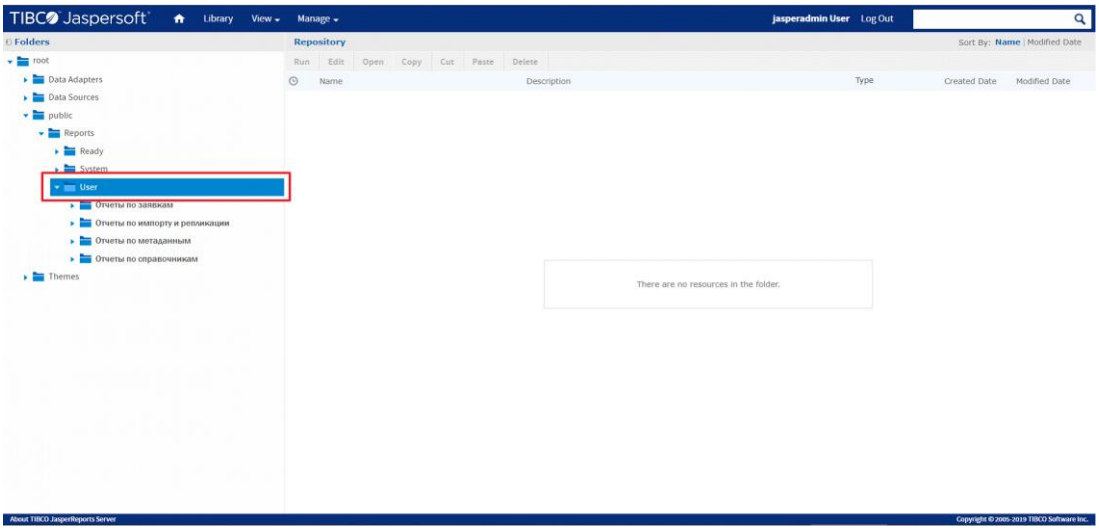


Рисунок 507 - Пользовательские отчеты в репозитории

Для дополнения параметра отчета метаданными используется поле «Description». В нем можно задать JSON-строку со следующей структурой:

interface ReportParamInfo {

label?: string; // локализованное наименование параметра, необязательный. Если не указано, используется имя параметра

type?: ReportParamType; // специальный тип параметра, необязательный

tip?: string; // подсказка, отображаемая при наведении курсора на иконку «?», необязательный. У каждого специального типа параметра есть соответствующая умолчательная подсказка, которую при необходимости можно убрать, задав в JSON-строке “tip”: “”.

}

Пример JSON-строки **Description**: {"label": "Справочники","type":"EntryIds"}

Примитивный тип параметра берется автоматически из его класса (Рисунок 508).

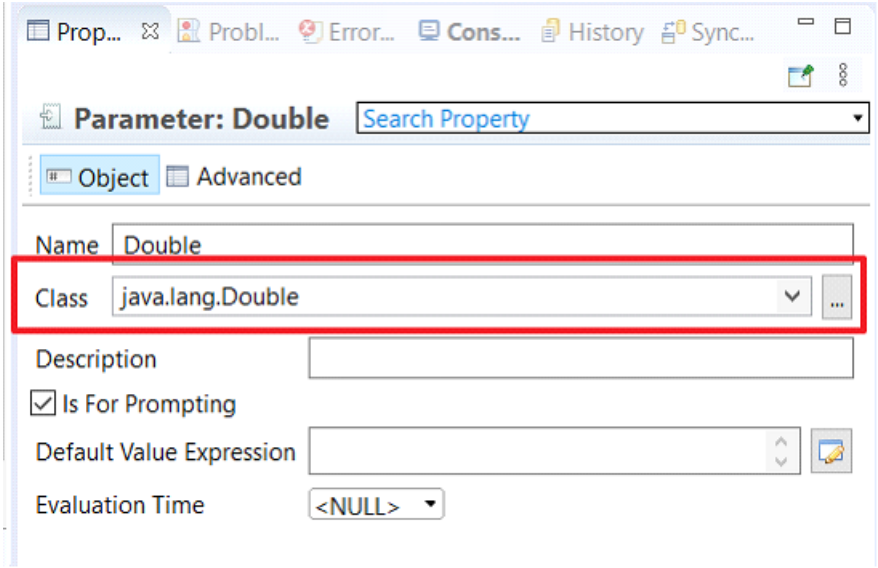


Рисунок 508 - Примитивный тип параметра

В клиенте Semantic MDM предусмотрены контролы для ввода параметров следующих специальных типов (**ReportParamType**):

- **ExecutorIds** – позволяет задать список идентификаторов Экспертов НСИ для заявок. Записывает в параметр значение в формате «id1,id2,…»;
- **EntryIds** – позволяет задать список идентификаторов справочников и классификаторов. Записывает в параметр значение в формате «id1,id2,…»;
- **NodeIds** – позволяет задать список идентификаторов групп. Записывает в параметр значение в формате «id1,id2,…»;
- **ApplicationYears** – позволяет задать список годов, в которые в системе регистрировались заявки. Записывает в параметр значение в формате «2020,2021,…»;
- **PeriodBegin + PeriodEnd** – позволяет задать временной интервал, определяемый двумя датами. Следует обратить внимание на то, что поле **label** в **Description** у параметров PeriodBegin и PeriodEnd должны

совпадать. Записывает в параметры даты в формате «yyyy-MM-ddTHH:mm:ss»

Изменять наименование параметра отчета через интерфейс сервера отчетов SemanticReports нельзя!

Отчет может быть сгенерирован в следующих форматах: PDF, XLS, XLSX, RTF, CSV, XML, DOCX, ODT, ODS, JRPRINT, JSON, HTML.

Форматы JSON и HTML используются исключительно для отображения превью отчета, поэтому соответствующие им кнопки экспорта на интерфейсе портала Semantic MDM не отображаются.

По умолчанию отчеты генерируются в форматах XLSX, PDF и HTML. При необходимости можно указать требуемый список форматов в JSON-строке отчета, задаваемый в поле **Description** (Рисунок 509).

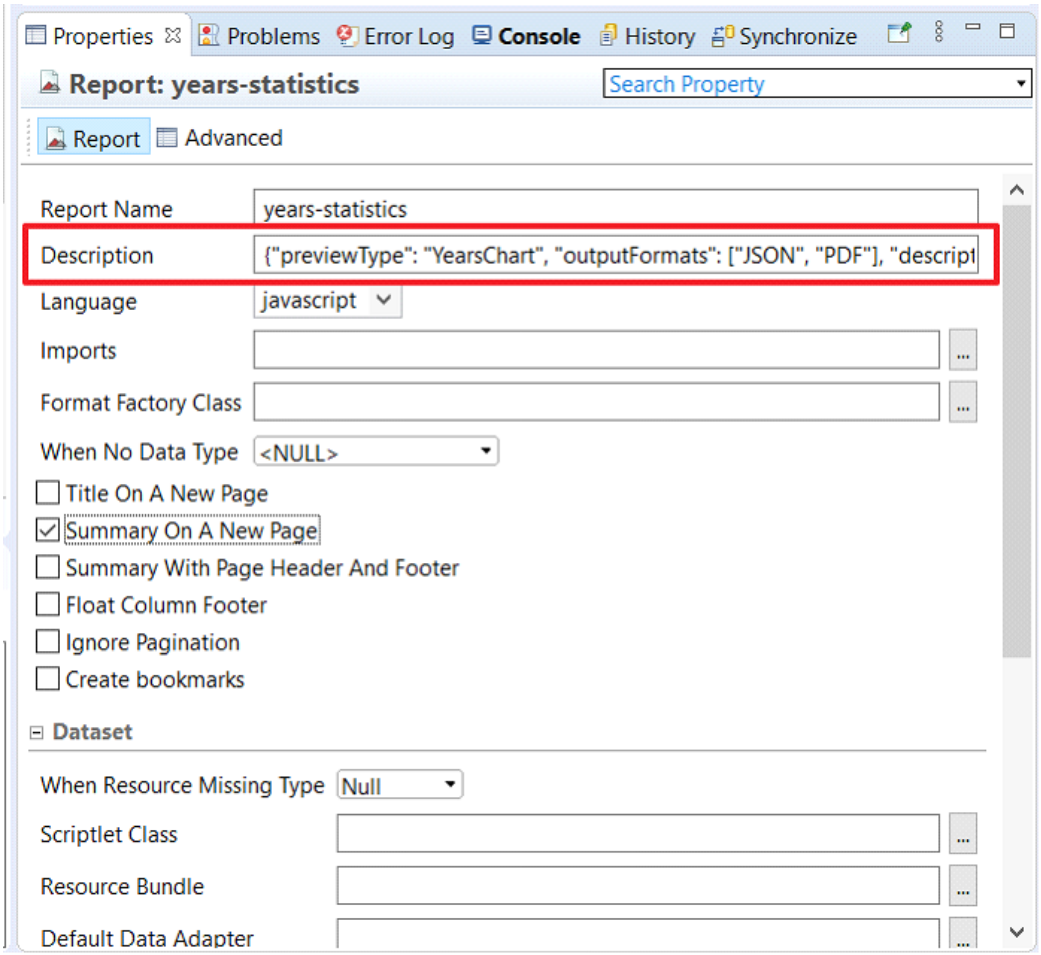


Рисунок 509 - Задание форматов и превью отчета

Структура JSON-строки:

interface ReportUnitInfo {

previewType?: ReportPreviewType; // тип превью, необязательный. По умолчанию отчет в формате HTML

outputFormats?: JrJobOutputFormat[]; // форматы, необязательный. По умолчанию XLSX, PDF и HTML (для превью)

description?: string; // описание, необязательный

}

Пример JSON-строки **Description**: {"previewType": "YearsChart", "outputFormats": ["JSON", "PDF"], "description": "какое-то описание…"}

Нужно иметь в виду, что сервер SemanticReports накладывает ограничение на максимальный размер поля **Description** в 250 символов, поэтому необходимо быть весьма лаконичным при заполнении поля ReportUnitInfo.description.

В качестве превью отчета по умолчанию используется его HTML представление. В поле **previewType** в **Description** можно указать специальный тип превью.

3.32 Методическое обеспечение

В этом разделе даны примеры использования различных подходов к ведению справочников.

3.32.1 Методики обработки заявок

В пункте представлены методики разработки и использования заявок различных типов.

3.32.1.1 Заявки с возможностью автоматической обработки

3.32.1.1.1 Описание задачи (кейса)

Обеспечить сокращение времени на обработку заявок за счет автоматизации процесса создания новых записей НСИ без участия эксперта НСИ.

3.32.1.1.2 Когда востребовано

Компаниями, в которых бизнес-процессом предусмотрено, что *Инициатор* (заявитель) полностью заполняет данные по заявке и отвечает за их качество.

3.32.1.1.3 Условия использования

Для использования автоматической обработки заявок необходимо, чтобы:

- В Semantic MDM на справочных группах настроены все необходимые процедуры проверки качества данных заявки – проверка заполнения значений обязательных атрибутов, проверка значений атрибутов на уникальность, а также используются ограничительные таблицы и функции.
- Шаблон заявки содержит ссылку на автоматический обработчик заявки.
- В системе зарегистрировано специальное расширение – автоматический обработчик заявок. Реализация таких обработчиков выполняется в виде отдельных сборок (war), которые при загрузке на сервер приложений Wildfly, выполняют свою регистрацию и создают параметры для настройки обработчика. Минимальный набор параметров обработчика должен содержать:
  - **url** – адрес сервера приложений Wildfly Semantic MDM;
  - **Наименование модуля** – данное наименование будет видно при выборе обработчика для шаблона заявки;
  - **Логин пользователя** – от имени которого будут выполняться все действия в системе;

- **Пароль пользователя** – от имени которого будут выполняться все действия в системе.

3.32.1.1.4 Работа с заявкой

Заявки с автоматической обработкой не требуют участия эксперта НСИ в создании новой справочной записи. Если в заявке с настроенным автоматическим обработчиком заполнены все необходимые атрибуты, то после ее отправки *Инициатором* (заявителем) и изменения статуса с «Зарегистрирована» на статус «Ожидает обработки», автоматический обработчик запускает процедуру создания нового объекта НСИ по следующей схеме (Рисунок 510).

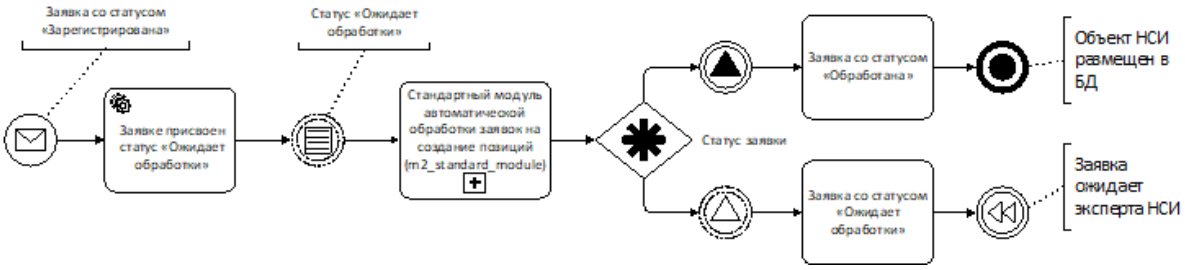


Рисунок 510 - Схема бизнес-процесса работы с заявкой при использовании автоматического обработчика

В качестве примера рассмотрим Стандартный модуль автоматической обработки заявок на создание позиций (m2_standard_module), который зарегистрирован в Системе (Настройки --> Расширения --> Автоматические обработчики заявок). Он реализует стандартный сценарий обработки, со следующим алгоритмом (Рисунок 511):

1) Получить атрибут заявки, заполняемый по умолчанию, и извлечь из него ссылку на группу (ноду). Если атрибут заявки содержит ссылку на позицию, то получить группу (ноду), в которой располагается позиция. Если группа была успешно получена, то перейти к следующему шагу.
2) Проверить что полученная группа является конечной. Если группа является конечной, то перейти к следующему шагу.

3)Выполнить поиск существующих позиций от заявки, через имеющиеся поиски дубликатов. Если ни одной позиции не нашлось, то перейти на следующий шаг.

4) Проверить в заявке корректность заполнения атрибутов позиций типа агрегация. Все значения должны быть ссылками на позиции и соответствовать scope декларации соответствующего атрибута в группе(ноде). Если все ссылочные атрибуты заполнены корректно, то перейти на следующий шаг.

5) Взять заявку в работу и создать связанный с ним пакет изменений

6) Создать новую позицию в группе (ноде), скопировать в нее значения атрибутов из заявки и установить статус «Нормализована»

7) Проверить на соответствие обязательным ограничениям и если позиция соответствует ограничениям, то отправить заявку на согласование/публикацию.

Если на одном из шагов, описанных выше, произошла ошибка или не выполнилось условие перехода на следующий шаг, то система должна добавить в журнал заявки информацию о причине, по которой не удалось автоматически обработать заявку, после чего требуется:

- пакет изменений отменить,
- заявку перевести на статус «Ожидает обработки».

Стандартный сценарий является универсальным и его можно было бы назначать на любые шаблоны заявок, предназначенные для создания позиций. Данный обработчик должен срабатывать при переходе из заявки статуса «Зарегистрировано» в статус «Ожидает обработки» / «Ожидает диспетчеризации».

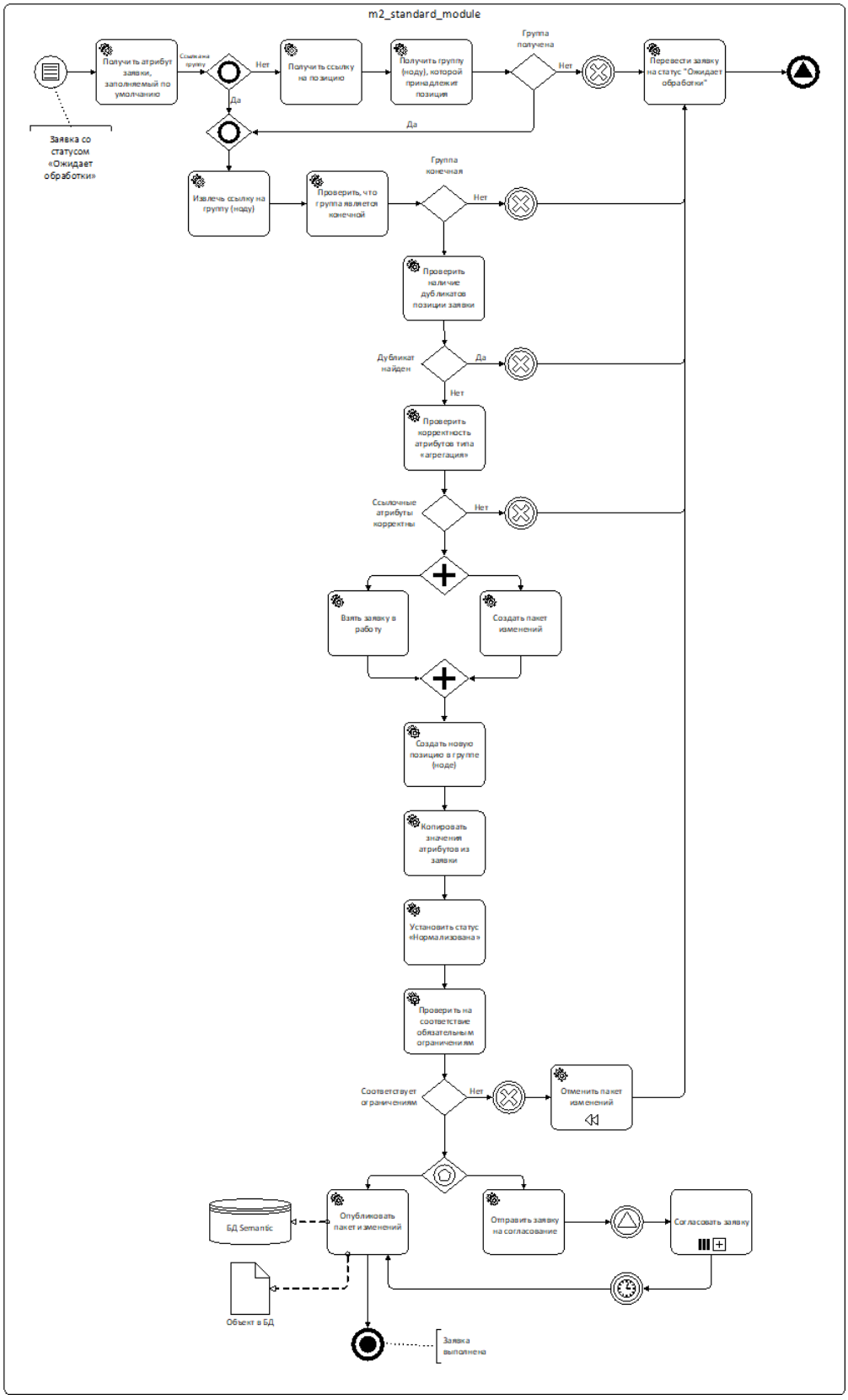


Рисунок 511 - Алгоритм работы автоматического обработчика заявок

Кроме описанного выше стандартного сценария обработки, могут быть реализованы и более сложные сценарии, в которых может выполняться обработка любых переходов заявки из одного статуса в другой.

3.32.1.2 Заявки с динамическим списком атрибутов

3.32.1.2.1 Описание задачи (кейса)

Заявка на добавление новой записи в справочник должна обеспечивать разный набор заполняемых атрибутов, в зависимости от справочной группы, в которую необходимо добавить объект НСИ, тем самым сокращается набор шаблонов заявок, которые необходимо разрабатывать для одного справочника.

Рассмотрим задачу на примере заявки на добавление объекта в справочник *Стандартные изделия*.

3.32.1.2.2 Когда востребовано

Если в разных группах (ветках) справочника содержаться объекты, атрибутивный состав которых имеет различия и при этом существует необходимость сократить число шаблонов заявок за счет их универсализации, например, в рамках одной большой справочной группы.

3.32.1.2.3 Настройка шаблона заявки

1. Перейдите в представление **Конфигурация данных** и откройте раздел **Шаблоны заявок**.
2. Выберите нужную папку с шаблонами и создайте в ней новый шаблон. Создавать новый пакет изменений не требуется.
3. Заполните ячейки:
   - *Наименование* – название шаблона заявки, по которому пользователь сможет найти его в списке шаблонов (английское наименование по необходимости) – Новое СТИ;
   - *Тип заявки* – для заявок на новый объект НСИ – «добавление». Позволяет группировать заявки и осуществлять поиск в списке шаблонов;

- *Формула предобработки* – оставьте не заполненным;
- *Автоматический обработчик заявк*и – оставьте не заполненным.

4. В разделе Атрибуты заявки (Рисунок 512) добавьте новой атрибут с наименованием «Справочная группа».
5. В качестве типа выберите из выпадающего списка «Ссылка на группу/позицию».
6. Признак Множественный не заполняйте.
7. Значение по умолчанию – выберите справочную группу Болты. В этом случае в шаблоне будет по умолчанию предлагаться список атрибутов этой группы.
8. Обязательность – выберите из выпадающего списка вариант Обязательный для Заявителя.
9. В ячейке Область при помощи кнопки Добавить вызовите окно Все справочники, где выберите справочную группу Основные справочники --> Единый справочник Номенклатуры --> B2 Крепежные изделия и нажмите кнопку Выбрать. Группа B2 Крепежные изделия будет добавлена в качестве области для выбора справочной группы.
10. Установите признак Заполнить при создании заявки из справочника. В этом случае при создании заявки из какой-либо ветки справочника B2 Крепежные изделия, в заявку попадут атрибуты именно этой ветки.

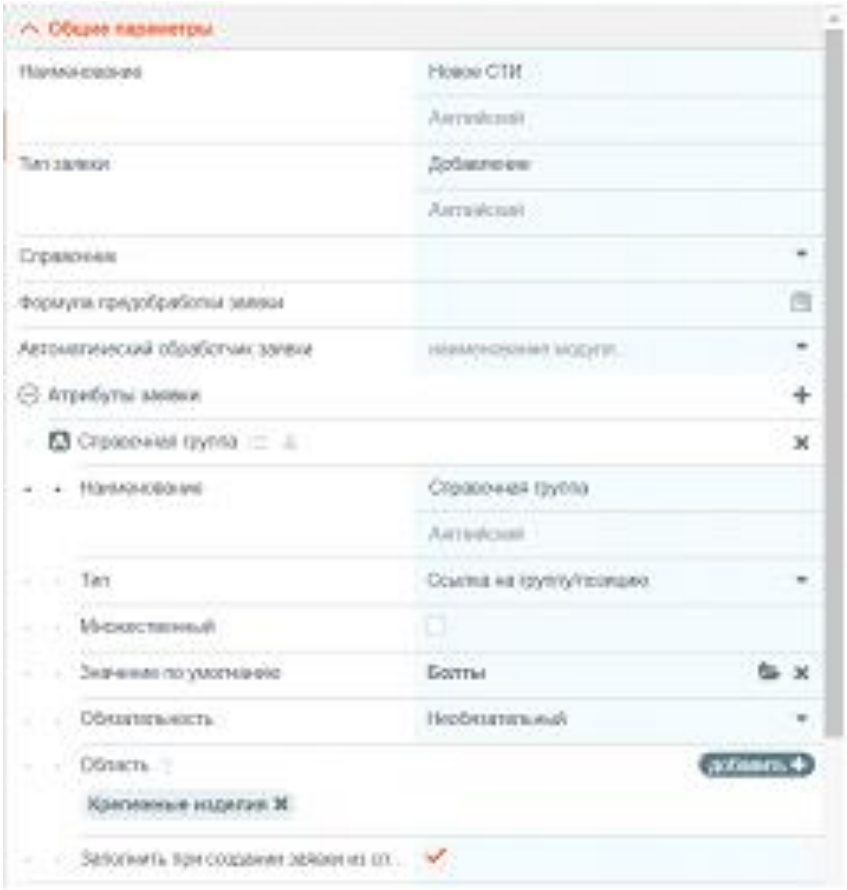


Рисунок 512 - Настройка атрибутов заявки

11. В разделе «Атрибуты позиции» выберите из выпадающего списка значение «Динамические».
12. В ячейке Получить атрибуты позиции из группы из выпадающего списка выберите название атрибута заявки (см. пункт 4), в котором содержится ссылка на группу / позицию. В нашем случае – атрибут «Справочная группа».
13. В ячейке Отображаемые значения останется только один вариант «Новые значения».
14. Настройка экспертов НСИ и маршрута согласования выполняется при необходимости согласно пунктам Эксперт НСИ и Маршрут согласования. В нашей задаче оставьте эти разделы не заполненными.
15. Настройте видимость шаблона. Достаточно указать себя в качестве пользователя, которому доступен данный шаблон.

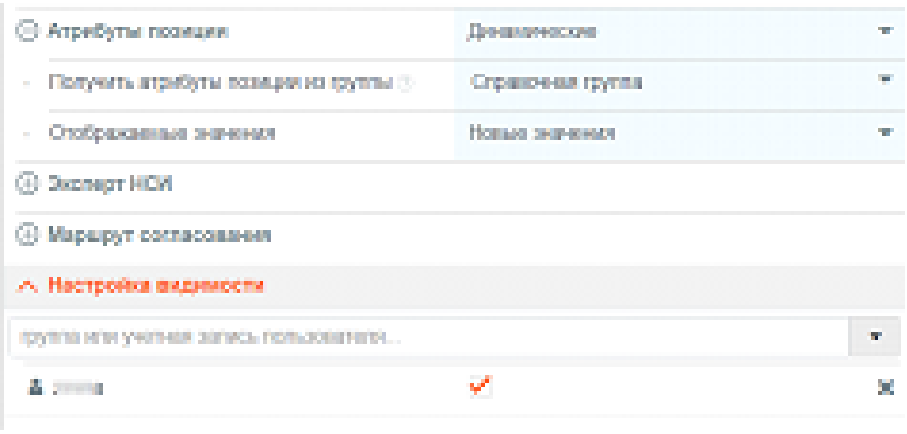


Рисунок 513 - Настройка видимости шаблона заявки

3.32.1.2.4 Работа с заявкой

1. Перейдите в представление Справочники. Переместитесь в справочную группу Основные справочники --> Единый справочник Номенклатуры --> В2 Крепежные изделия --> Гайки.
2. Нажмите над таблицей данных кнопку Заявки и выберите команду Создать заявку.
3. В открывшемся окне выберите созданный шаблон заявки. Если в списке шаблонов их очень много, воспользуйтесь поиском в верхней части окна. Установите курсор на наименовании шаблона и нажмите кнопку ОК.
4. В заявке автоматически заполнен атрибут заявки Справочная группа. Там указана та группа, из которой было вызвано создание заявки. Измените справочную группу нажав на кнопку Выбрать. Укажите группу Болты. Обратите внимание, что состав атрибутов заявки поменялся.
5. Откажитесь от создания заявки нажав на кнопку Удалить.

3.33 Скрипты GROOVY

В разделе изложены методические подходы и особенности разработки скриптов GROOVY для работы с данными в *Semantic MDM*. Описаны методы

и даны примеры решения конкретных практических задач, встречающихся при эксплуатации *Системы*.

3.33.1 Использование скриптов Groovy в вычисляемых атрибутах

Основным назначением вычисляемых атрибутов в *Semantic MDM* является выполнение вычислений на основе имеющихся значений в атрибутах позиции или значений атрибутов из агрегированных/композированных позиций. Для чтения значений атрибутов необходимо использовать только те методы, что предусмотрены в разделе Методы чтения данных из Semantic MDM.

3.33.1.1 Методы чтения данных из Semantic MDM

3.33.1.1.1 Чтение данных о позиции

В специальной переменной **support** предусмотрены следующие функции:

- **@[#itemGuid]** – получить **guid** текущей обогащаемой позиции. Данная сигнатура используется без **support**;
- **getItemUid(Long itemId)** – получить **guid** позиции по ее числовому идентификатору.

3.33.1.1.2 Чтение значений атрибутов позиции

Основным способом чтения значений атрибутов текущей позиции, является чтение через конструкцию **@[имя атрибута]**. Для чтения значения вложенных в агрегации атрибутов, необходимо последовательно перечислить имена атрибутов, например:

- **@[имя атрибута связи][имя вложенного атрибута**]
- **@[имя атрибута связи][имя вложенного атрибута связи][имя вложенного атрибута]**
- **@[#composer][имя вложенного атрибута]**
- **@[#composer][имя вложенного атрибута связи][имя вложенного атрибута]**

Перечисленные конструкции вернут одно значение атрибута, если все перечисленные в конструкции атрибуты хранят единичное значение. Если хотя бы один из перечисленных в конструкции атрибутов хранит массив значений, то данная конструкция вернет List значений.

Кроме получения списка значений (**List**) для вложенного атрибута, есть возможность получить полный путь до этого значения через метод: **getMapValues()**, который вернет **Map<String, Object>**, где **String** будет содержать GUID агрегированной позиции, а **Object** будет содержать либо значение заданного атрибута, либо еще один **Map<String, Object>**, если путь к вложенному атрибуту состоит из более чем одного атрибута связи.

Примеры возвращаемых значений:

- **@[имя множественного атрибута связи][имя вложенного строкового атрибута].getMapValues()** – данная конструкция вернет **Map<String, String>**
- **@[имя множественного атрибута связи] [имя единичного атрибута связи] [имя вложенного строкового атрибута].getMapValues()** – данная конструкция вернет **Map<String, Map<String, String>>**
- **@[имя множественного атрибута связи] [имя единичного атрибута связи] [имя вложенного множественного строкового атрибута].getMapValues()** – данная конструкция вернет **Map<String, Map<String, List<String>>>**

Для каждого из типов данных атрибутов, будет возвращен свой тип данных:

- **строка**: String
- **целое число**: Integer
- **число с плавающей запятой**: valueDouble - методы для работы с **ValueDouble** см. в разделе Методы для работы с числами (valueDouble)
- **дата**: Date
- **логический тип**: Boolean
- **цвет**: Long. – число в формате Blue, Green, Red, Alpha, где Blue –

страший байт, а Alpha – младший байт. Пример красного цвета в шестнадцатиричном строковом представлении: “241CEDFF”, где: Blue=24, Green=1C, Red=ED, Alpha=FF.

- Преобразование цвета из шестнадцатеричного строкового представления в число: Long.parseLong ("241CEDFF",16);
- Преобразование числа в строковое шестнадцатеричное представление: Long.toString (605875711,16);

- **файл**: String - имя файла;
- **атрибут связи**: String - с guid позиции.

3.33.1.2 Методы для работы с числами (valueDouble)

Для более удобной работы с числом с плавающей запятой (**valueDouble**), в **support** реализованы вспомогательные методы, позволяющие получить необходимые данные как для вычисления, так и для формирования строковых обозначений:

- **BigDecimal getDoubleValue(valueDouble value, Integer precision, BigDecimal koef)** – получить основное значение в базовой единице измерений, домноженное на коэффициент (**koef**) и округленное до заданного количества знаков (**precision**). Если **value** равно **null**, то функция вернет 0. В любом из параметров может быть передан **null**, в этом случае функция вернет число без домножения на коэффициент (**koef**) и/или округления до заданного количества знаков (**precision**). Функция вернет значение только для **getValueType()**: **NORMAL, DEFLECTION**, в противном случае функция вернет ошибку;
- **BigDecimal getLeftDoubleValue(valueDouble value, Integer precision, BigDecimal koef)** – получить значение слева в базовой единице измерений, домноженное на коэффициент (**koef**) и округленное до заданного количества знаков (**precision**). Если **value** равно **null**, то функция вернет 0. В любом из параметров может быть передан **null**, в этом случае функция вернет число без домножения на коэффициент

(**koef**) и/или округления до заданного количества знаков (**precision**). Функция вернет значение только для **getValueType: dtRange, dtDeflection**, в противном случае функция вернут ошибку;

- **BigDecimal getRightDoubleValue(valueDouble value, Integer precision, BigDecimal koef)** – получить значение справа в базовой единице измерений, домноженное на коэффициент (**koef**) и округленное до заданного количества знаков (**precision**). Если **value** равно **null**, то функция вернет 0. В любом из параметров может быть передан **null**, в этом случае функция вернет число без домножения на коэффициент (**koef**) и/или округления до заданного количества знаков (**precision**). Функция вернет значение только для **getValueType: dtRange, dtDeflection**, в противном случае функция вернут ошибку;
- **DoubleFormatType getValueType(valueDouble value)** – получить тип числа:
    - **NORMAL** – простое число;
    - **RANGE** – диапазон;
    - **DEFLECTION** – число с отклонениями;
- I**nteger getEiPrecision(valueDouble value)** – получить количество знаков после запятой (точность), заданное пользователем при вводе. Если точность не определена, то вернуть **null**;
- **String getEiCode(valueDouble value)** – получить GUID единицы измерения, в которой пользователь задал значение. Если ЕИ не определена для атрибута, то вернет **null**;
- **BigDecimal getEiKoef(valueDouble value)** – получить коэффициент пересчета в базовую ЕИ. Если коэффициент запрашивается для базовой ЕИ, то будет возвращена 1;
- **String getEiGuid(valueDouble value)** - получить GUID единицы измерения. Если value==null, то вернуть пустую строку;
- **BigDecimal getValueInBaseEi(BigDecimal value, String EiGuid)** - пересчитать значение (**value**) из заданной ЕИ (**EiGuid**) в базовую ЕИ.

Если EiGuid является базовой или содержит null или пустую строку, то вернуть значение value без модификаций;

- **createDblSimpleValue(BigDecimal value, Integer precision, String eiGuid)** - создать **valueDouble** на основе переданного числа **value**, в заданной единице измерения (**eiGuid**), которое надо отображать пользователю с заданной точностью **precision**. Если точность не указана (**null**), то значение не будет округляться для пользователя. Если **eiGuid** не указана (**null**), то считается что значение **value** передается в базовой единице измерения;
- **createDblRangeValue(BigDecimal leftValue, BigDecimal rightValue, Integer precision, String eiGuid)** - создать **valueDouble** на основе переданного диапазона **leftValue** и **rightValue**, в заданной единице измерения (**eiGuid**), которые надо отображать пользователю с заданной точностью. Если точность не указана (**null**), то значение не будет округляться для пользователя. Если **eiGuid** не указана (**null**), то считается что значения **leftValue** и **rightValue** передаются в базовой единице измерения;
- **createDblDeflectionValue(BigDecimal value, leftValue, rightValue, Integer precision, String eiGuid)** - создать **valueDouble** на основе переданного числа **value** с отклонениями **leftValue** и **rightValue**, в заданной единице измерения (**eiGuid**), которые надо отображать пользователю с заданной точностью **precision**. Если точность не указана (**null**), то значение и отклонения не будут округляться для пользователя. Если **eiGuid** не указана (**null**), то считается что значение **value** и отклонения **leftValue** и **rightValue** передаются в базовой единице измерения.

3.33.1.3 Пример решения расчетной задачи по расчету нормы расхода

На данном примере, наглядно показано, как через вычисляемые атрибуты выполнять вычисления с помощью скриптов Groovy, для решения

расчетных задач. Для проведения вычислений, рекомендуется использовать только переменные типа **BigDecimal**, которые позволяют получить нужную точность в десятичной системе исчисления. Для получения числовых значений из атрибутов, рекомендуется использовать методы: **getDoubleValue**, **getLeftDoubleValue**, **getRightDoubleValue**, которые позволяют получить числовое значение атрибута с требуемой точностью и пересчетом на необходимый коэффициент. Возврат вычисленного значения, рекомендуется выполнять через одну из функций: **createDblSimpleValue**, **createDblRangeValue**, **createDblDeflectionValue**. Если в формулах вычисления используются числовые константы, то к ним необходимо добавлять литеру “G”, чтобы указать системе что это число типа **BigDecimal**, например: “2G”, “3.1416G”. При необходимости выполнения округления числа типа **BigDecimal**, необходимо использовать метод **setScale()**, в который необходимо требуемую точность округления (количество знаков после запятой) и тип округления (**RoundingMode**):

Таблица 33 – Типы округления

Число	**Тип округления**						
	UP	**DOWN**	**CEILING**	**FLOOR**	**HALF_UP**	**HALF_DOWN**	**HALF_EVEN**
5.5	6	5	6	5	6	5	6
2.5	3	2	3	2	3	2	2
1.6	2	1	2	1	2	2	2
1.1	2	1	2	1	1	1	1
1.0	1	1	1	1	1	1	1
-1.0	-1	-1	-1	-1	-1	-1	-1
-1.1	-2	-1	-1	-2	-1	-1	-1
-1.6	-2	-1	-1	-2	-2	-2	-2
-2.5	-3	-2	-2	-3	-3	-2	-2
-5.5	-6	-5	-5	-6	-6	-5	-6

Пример округления до 5 знаков: D = D.setScale(5, RoundingMode.HALF_UP);. Для использования округления в скрипте, в начало скрипта требуется добавить: import java.math.RoundingMode.

Описание задачи по расчету нормы расхода: требуется вычислить норму расхода материала(@[Материал]) для заготовки по следующим данным:

- @[Материал заготовки][Форма сортамента] – форме материала;
- @[Материал заготовки][Марка материала][Плотность] – плотности материала, заданного в марке материала;
- @[Длина заготовки], @[Ширина заготовки] – геометрических параметров заготовки;
- Геометрических параметров материала заготовки.

Норму расхода требуется вычислить исходя из заданных параметров заготовки, геометрических параметров материала заготовки и плотности материала заготовки. Формула расчета, определяется формой сортамента материала заготовки. Все расчеты требуется проводить с использованием типа данных **BigDecimal**, который дает правильную точность при округлениях в десятичной системе исчисления.

Скрипт вычисления нормы расхода для кругов, лент, листов и труб:

```
ValueDouble getValue(){
   BigDecimal Res;
   try {
    // Форма сортамента
    String SortForm = @[Материал заготовки][Форма сортамента];
    if (SortForm != null) {
     //Получим длину заготовки и плотность материала
     BigDecimal L = support.getDoubleValue(@[Длина заготовки], 5, null);
     BigDecimal Ro = support.getDoubleValue(@[Материал заготовки][Марка материала][Плотность], 5, null);
```

```
switch(SortForm) {
  case "Круг":
  case "Проволока":
  case "Пруток":
  case "Стержень":
    BigDecimal D = support.getDoubleValue(@[Материал заготовки][Диаметр], 5, null);
    Res = L * D * D * Ro * 3.1416G * 1000G / 4G; // приводим к граммам
    return support.createDblSimpleValue(Res,2,null);// округляем до сотых грамма
    break;
  case "Лента":
    BigDecimal B = support.getDoubleValue(@[Ширина заготовки], 5, null);
    BigDecimal Hlent = support.getDoubleValue(@[Материал заготовки][Толщина], 5, null);
    Res = Hlent * B * L * Ro * 1000G; // приводим к граммам
    return support.createDblSimpleValue(Res,2,null);// округляем до сотых грамма
    break;
  case "Лист":
  case "Листы":
  case "Полоса":
  case "Полосы":
  case "Листовое органическое стекло":
  case "Войлок":
  case "Паронит":
  case "Плита":
    BigDecimal Hl = support.getDoubleValue(@[Материал
```

```
заготовки][Толщина], 5, null);
                BigDecimal B = support.getDoubleValue(@[Ширина заготовки], 5,
null);
                Res = Hl * B * L * Ro * 1000G; // приводим к граммам
                return  support.createDblSimpleValue(Res,2,null);// округляем до
сотых грамма
                break;
              case "Труба":
                // Марка материала
                BigDecimal      Dn      =      support.getDoubleValue(@[Материал
заготовки][Наружный диаметр], 5, null);
                BigDecimal      Hst     =      support.getDoubleValue(@[Материал
заготовки][Толщина стенки], 5, null);
                Res = L * 3.1416G * (Dn * Dn – 4G * (Dn / 2G - Hst) * (Dn / 2G -
Hst)) * Ro * 1000 / 4; // приводим к граммам
                return  support.createDblSimpleValue(Res,2,null); // округляем до
сотых грамма
                break;
              case "Шестигранник":
                BigDecimal      Dvk     =      support.getDoubleValue(@[Материал
заготовки][Диаметр], 5, null);
                Res = 0.86605G * Dvk * Dvk * L * Ro * 1000G; // приводим к
граммам
                return  support.createDblSimpleValue(Res,2,null); // округляем до
сотых грамма
                break;
              default:
                return support.createDblSimpleValue(0,null,null);
                break;
            }
```

```
        }
        else
          return support.createDblSimpleValue(0,null,null);
      }
      catch (Exception ex) {
        return support.createDblSimpleValue(0,null,null);
      }
    }
```

3.33.1.4 Пример решения расчетной задачи по расчету содержания драг. металлов по марке материала

На данном примере, наглядно показано, как через вычисляемые атрибуты читать необходимые данные из вложенных атрибутов множественных композиций и проводить необходимые вычисления с помощью скриптов Groovy. Для проведения вычислений, рекомендуется использовать переменные типа **BigDecimal**, которые позволяют получить нужную точность в десятичной системе исчисления. Для получения числовых значений из атрибутов, рекомендуется использовать методы: **getDoubleValue**, **getLeftDoubleValue**, **getRightDoubleValue**, которые позволяют получить числовое значение атрибута с требуемой точностью и пересчетом на необходимый коэффициент. Для получения списка значений из атрибутов множественной композиции, необходимо использовать метод **getMapValues()**. Возврат вычисленного значения, рекомендуется выполнять одну из функций: **createDblSimpleValue**, **createDblRangeValue**, **createDblDeflectionValue**.

```
ValueDouble getValue(){
    //Получим процентное содержание серебра в марке материала
    BigDecimal valDragWeight = support.getDoubleValue(
@[Материал][Содержание серебра (Ag)],null,null);
    //Получим наименование меры (класса ЕИ)
```

```
String EI_class = @[Базовая ЕИ][Класс ЕИ][Наименование класса ЕИ];

double RecalcKoef=1;
String MassaEIGuid=null;
Map<Integer, Object> mapKoef = [:];
Map<Integer, Object> mapEI = [:];
//Получим долю содержания серебра в 1 грамме.
BigDecimal DragPercent=support.getDoubleValue(@[Материал][Содержание серебра (Ag)], 2, 0.01G);
//Если отсутствуют данные по содержанию серебра, то вернем пустое значение
if (DragPercent == 0)
  return null;
  //Если базовая ЕИ не относится к Массе, то найдем среди логистических ЕИ, то ЕИ что относится к Массе и через ее коэф. пересчета получим содержание серебра
if (EI_class!="Масса") {
  //Получим карту соответствия значений: "GUID логистической ЕИ" : "Коэффициент пересчета"
  mapKoef = @[Логистические ЕИ][Коэффициент пересчета в базовую ЕИ].getMapValues();
  //Получим карту соответствия значений: "гуид логистической ЕИ" : "GUID ЕИ, назначенной в качестве логистической"
  mapEI = @[Логистические ЕИ][Логистическая ЕИ].getMapValues();
  //Получим список GUID логистических ЕИ
  def objGuids = mapEI.keySet();
  //Выполним поиск ЕИ, относящейся к Массе и посчитаем в граммах содержание для базовой ЕИ
  for (String objGuid in objGuids) {
```

```
            //Получим по ключу гуид логистической ЕИ
            String eiLogGuid = mapEI.get(objGuid);
            //Получим коэффициент пересчета
            BigDecimal eiRecalcKoef =
support.getDoubleValue(mapKoef.get(objGuid),null,null);
            //return (eiRecalcKoef * DragPercent).toPlainString();
            if
(((eiLogGuid=="5h6GkqQcBRlsBUJqCgvBka")||(eiLogGuid=="Kknb4s8zFcKeJ
RTo.6rlFb")||(eiLogGuid=="YfCXjsngrBA.0FPpIT0Y7a"))&&(eiRecalcKoef!=0))
{
              //Вернем долю содержания драг. металла в граммах
              if (eiLogGuid=="5h6GkqQcBRlsBUJqCgvBka")//расчет содержания
для ЕИ: Грамм
                return support.createDblSimpleValue(eiRecalcKoef * DragPercent,
null, "5h6GkqQcBRlsBUJqCgvBka");
              else if (eiLogGuid=="Kknb4s8zFcKeJRTo.6rlFb")//расчет
содержания для ЕИ: килограмм
                return support.createDblSimpleValue(eiRecalcKoef * DragPercent *
0.001G, null, "5h6GkqQcBRlsBUJqCgvBka");
              else if (eiLogGuid=="YfCXjsngrBA.0FPpIT0Y7a")//расчет
содержания для ЕИ: тонна
                return support.createDblSimpleValue(eiRecalcKoef * DragPercent *
0.000001G, null, "5h6GkqQcBRlsBUJqCgvBka");
              }
            }
          }
          else {
            //Получим гуид базовой единицы измерения
            MassaEIGuid=@[Базовая ЕИ];
```

```
        //Вернем долю содержания драг. металла в граммах для базовой ЕИ
        if (MassaEIGuid=="5h6GkqQcBRlsBUJqCgvBka")//расчет содержания для ЕИ: Грамм
            return support.createDblSimpleValue( DragPercent, null, "5h6GkqQcBRlsBUJqCgvBka");
        else if (MassaEIGuid=="Knb4s8zFcKeJRTo.6rlFb")//расчет содержания для ЕИ: килограмм
            return support.createDblSimpleValue( DragPercent * 0.001G, null, "5h6GkqQcBRlsBUJqCgvBka");
        else if (MassaEIGuid=="YfCXjsngrBA.0FPpIT0Y7a")//расчет содержания для ЕИ: тонна
            return support.createDblSimpleValue( DragPercent * 0.000001G, null, "5h6GkqQcBRlsBUJqCgvBka");
        }
        return null;
    }
```

3.33.1.5 Пример формирования строкового обозначения емкости для конденсаторов KEMET CDR 01-35

Емкость конденсаторов KEMET серии CDR 01-35 должна задаваться в пикофарадах и выводится в обозначение по следующему правилу: <первые две цифры из значения ёмкости><количество нулей после первых двух цифр значения ёмкости>. Например, для ёмкости “22000” должно быть сформировано обозначение “223”, где “22” первые две цифры из значения емкости, а “3” это количество нулей после первых двух цифр.

Все значения в системе хранятся в базовой единице измерения. Для электрической емкости, базовой единицей измерения является “Фарад”. Соответственно при вводе значения в “Пикофарад”, система переводит введенное значение в “Фарад” (путем деления на “1 000 000 000 000”) и сохраняет введенное значение в БД. Для получения в скрипте Groovy значения

в “Пикофарад”, требуется домножить полученное значение на “1 000 000 000 000G”(литера G необходима для определения числа как **BigDecimal**) и округлить значение до целого.

```
import java.text.*;

String getValue(){
 String res = "";
 //Получим значение в пикофарадах, округленное до целого
 BigDecimal value = support.getDoubleValue(@[Номинальная емкость(ориг)],0, 1000000000000G);
  //Если число меньше 100, то выведем две цифры
 if (value<100) {
  res = new DecimalFormat("00").format(value);
 }
 //Если число больше или равно 100, то посчитаем количество нулей после первых двух цифр(ZeroCount) и выведем первые две цифры и количество нулей
 else {
  Integer ZeroCount = 0;
  while (value>=100G) {
   value = value / 10G
   ZeroCount = ZeroCount + 1;
  }
  res = new DecimalFormat("00").format(value)+ ZeroCount.toString();
 }
 //вернуть полученное обозначение
 return res;
}
```

3.33.1.6 Пример формирования строкового обозначения максимального рабочего давления для датчиков давления

Данный пример форматирует вывод максимального рабочего давления в обозначение датчика давления по следующим правилам:

- Если значение в Мпа от 0 до 10, то значение давления должно быть округлено и выведено с двумя знаками после запятой;
- Если значение в Мпа от 10 до 100, то значение давление должно быть округлено и выведено с одним знаком после запятой;
- ·Если значение в Мпа от 100 и более, то значение давления должно быть округлено до целого и выведено без знаков после запятой.

Все значения в системе хранятся в базовой единице измерения. Для давления, базовой единицей измерения является «Паскаль». Соответственно при вводе значения в «Мегапаскаль», система переводит введенное значение в «Паскаль» (путем деления на 0,000001) и сохраняет введенное значение в БД. Для получения в скрипте Groovy значения в «Мегапаскаль», требуется домножить полученное значение на 0,000001G (литера G необходима для определения числа как **BigDecimal**) и округлить значение до требуемой точности «2».

```
import java.text.*;
String getValue(){
    // для обращения к значению свойства позиции использовать конструкцию вида:
    // @[Наименование свойства]
    // для обращения к значению свойства агрегированной позиции:
    // @[Наименование агрегации][Наименование свойства]
    // вернуть вычисленную строку
    BigDecimal value = support.getDoubleValue(@[Максимальное рабочее давление],2,0.000001G);
    if ((value>0)&&(value<10)) {
      //округлим значение до двух знаков
```

```
    return '-'+new DecimalFormat("#0.00").format(value);
   }
   else if ((value>=10)&&(value<100)) {
    //округлим значение до одного знака
    return '-'+new DecimalFormat("#.0").format(value);
   }
   else if (value>=100) {
    //округлим значение до целого
    return '-'+new DecimalFormat("#").format(value);
   }
   else
    return '';
}
```

3.33.1.7 Пример форматирования обозначения типоразмера металлической сетки по ГОСТ 3826-82

Данный пример показывает, как форматировать вывод размера ячейки и диаметра проволоки в соответствии с ГОСТ 3826-82, согласно которому, разделитель дробной части (запятая) не должен выводиться, если значение меньше единицы.

Все значения в системе хранятся в базовой единице измерения. Для длины, базовой единицей измерения является “Метр”. Соответственно при вводе значения в “Миллиметр ”, система переводит введенное значение в “Метр” (путем деления на “ 1000”) и сохраняет введенное значение в БД. Для получения в скрипте Groovy значения в “Миллиметр”, требуется домножить полученное значение на “ 1000G”(литера G необходима для определения числа как **BigDecimal**) и округлить значение до требуемой точности. Точность должна быть получена из значения, введенного пользователем, с помощью функции **support.getEiPrecision()**.

```
import java.text.*;
```

```
String getValue(){
    //Получим точность значения, заданного в размере ячейки при вводе значения пользователем
    Integer precision = support.getEiPrecision(@[Размер ячейки]);
    BigDecimal vCellSize = support.getDoubleValue(@[Размер ячейки],precision,1000G);

    //Получим точность значения, заданного в номинальном диаметре проволоки при вводе значения пользователем
    precision = support.getEiPrecision(@[Номинальный диаметр проволоки]);
    BigDecimal vDiam = support.getDoubleValue(@[Номинальный диаметр проволоки],precision,1000G);

    //Преобразуем значение в строку и удалим запятую, если значение меньше 1
    String strCellSize = new DecimalFormat("#0.###").format(vCellSize);
    if (vCellSize<1)
      strCellSize = strCellSize.replaceAll(',',"");

    //Преобразуем значение в строку и удалим запятую, если значение меньше 1
    String strDiam = new DecimalFormat("#0.###").format(vDiam);
    if (vDiam<1)
      strDiam = strDiam.replaceAll(',',"");

    //Вернем полученный результат
    return strCellSize+'-'+strDiam;
}
```

3.33.1.8 Пример вывода диапазона измерений для калибров

Данный пример форматирует вывод диапазона измерений калибров по следующей формуле: “(<нижний диапазон измерений> - <верхний диапазон измерений> <обозначение ЕИ>)”. Задачей данного скрипта является:

1. Получить точность, заданную пользователем при вводе значения (**support.getEiPrecision()**).
2. Получить коэффициент пересчета из базовой единицы измерения, в единицу измерений, в которой ввел значение пользователь (**support.getEiKoef()**).
3. Получить обозначение единицы измерения, в которой ввел значение пользователь (**support.getEiCode()**).
4. Получить нижнюю и верхнюю границу диапазона измерений с пересчетом и округлением до требуемой точности (**support.getLeftDoubleValue()** и **support.getRightDoubleValue()**).
5. Определить формат преобразования значений атрибутов из **BigDecimal** в **String**.
6. Сформировать строку по заданной формуле.

```
import java.text.DecimalFormat;
String getValue(){
    //Получить тип числа с плавающей запятой и проверить что тип соответствует диапазону
    DoubleFormatType vType = support.getValueType(@[Диапазон измерения]);
    if (vType == DoubleFormatType.RANGE) {
     //Получим точность, коэффициент и обозначение ЕИ в котором пользователь задал значение
      Integer precision = support.getEiPrecision(@[Диапазон измерения]);
      BigDecimal eiKoef = support.getEiKoef(@[Диапазон измерения]);
      String eiCode = support.getEiCode(@[Диапазон измерения]);
```

```
            //Получим левую и правую границу диапазона
            leftValue = support.getLeftDoubleValue(@[Диапазон измерения], precision, eiKoef);
            rightValue = support.getRightDoubleValue(@[Диапазон измерения], precision, eiKoef);

            //Настроим формат преобразования значения к строке
            DecimalFormat df;
            if (precision == 0) {
              df = new DecimalFormat("#"); //Формат для целого числа
            }
            else {
              df = new DecimalFormat("#.".padRight(precision+2,'#')); //Формат для числа с десятичной частью
            }
            return '('+df.format(leftValue)+' - '+df.format(rightValue)+' '+eiCode+')'
          }
          return "";
        }
```

3.33.2 Использование скриптов Groovy для обогащения данных

Для реализации обогащения данных, в *Semantic MDM* предусмотрена возможность разработки на скриптовом языке Groovy постобработчиков (см. п. 3.14.8) для следующих событий:

- **Создание позиции** – постобработчик разрабатывается для справочной группы в свойстве **Постобработчик на создание** и срабатывает автоматически, после создания позиции в Semantic MDM, в независимости от того, каким способом была создана позиция: по кнопке ***Добавить*** (см. п. 3.14.11), через импорт из Excel (см. п. 3.19.1)

или через импорт из подсистемы Импорта и репликации (см. раздел 3.30). Постобработчик не сработает, при выполнении перемещения позиции в группу, для которой разработан постобработчик.

- **Изменение статуса позиции** – постобработчик разрабатывается для справочной группы в свойстве **Постобработчик на изменение статуса** и срабатывает автоматически, после изменения статуса позиции в Semantic MDM, в независимости от того, каким способом было выполнено изменение статуса: в карточке позиции, через массовое назначение статуса или через импорт из подсистемы Импорта и репликации
- **Изменение значения атрибута** – постобработчик разрабатывается для декларации атрибута с единичным значением и типом данных: *Строка*, *Целое число*, Ч*исло с плав. запятой*, *Логический тип*, *Дата* в свойстве **Постобработчик на изменение** и срабатывает автоматически, после изменения значения атрибута и/или по запросу пользователя. Настройка автоматического запуска выполняется в свойстве **Постобработчик на изменение**, путем установки флага в опции “Автоматический запуск”, которая появляется после разработки Groovy-скрипта. Возможность запуска постобработчика вручную есть всегда, при любом значении опции “Автоматический запуск”. Допускается разработка постобработчика для вычисляемого атрибута, в этом случае запуск может выполняться только пользователем вручную, после ввода значения в вычисляемый атрибут

При создании позиции и изменении статуса, значений атрибутов, через импорт из Excel или импорт подсистемы импорта и репликации, последовательность вызова постобработчиков будет следующей:

1.Вызов постобработчика создания позиции.

2.Вызов постобработчиков изменения значений атрибутов и изменения статуса в произвольном порядке.

3.33.2.1 Методы для обогащения данных в Semantic MDM

Для обогащения данных, в системе Semantic MDM предусмотрена возможность разработки постобработчиков для следующих событий:

- Создание позиции;
- Изменение статуса;
- Изменение значения атрибута.

Для реализации обогащения данных в постобработчиках, в системе предусмотрены следующие методы:

- Создание новой позиции;
- Удаление существующей позиции;
- Изменение статуса позиции;
- Изменение одного или нескольких атрибутов позиции;
- Запуск других постобработчиков из реализуемого постобработчика: создания позиции, изменения статуса, изменения значения атрибута;
- Генерация уникальных номеров через счетчики.

Создание и удаление позиции

Методы для создания и удаления позиции нужны, чтобы обеспечить возможность создания / удаления позиций во вспомогательных справочниках. Например, если у контрагента имеется атрибут **Лица, имеющие право подписи** типа «*множественная композиция*», в котором указываются ссылки на позиции вспомогательного справочника *Лица, имеющие право подписи*, то для создания новых актуальных позиций и удаления неактуальные позиций необходимы данные функции. Для записи ссылки на созданную позицию в атрибут композиции, необходимо использовать **UpdateItemBuilder**.

- **String enrichCreateItem(String nodeUid)** – создать новую позицию в указанной группе (**nodeUid**) и вернуть GUID созданной позиции;
- **void enrichDeleteItem(String itemUid)** – выполнить удаление заданной позиции (**itemUid**).

Изменение атрибутов позиции

Для изменения атрибутов позиции, в системе предусмотрена специальная конструкция, позволяющая выполнять изменения атрибутов позиции одним запросом. Для этого необходимо использовать **UpdateItemBuilder**, который необходимо получить через метод **enrichUpdateItem()**. В **UpdateItemBuilder** предусмотрены специальные методы **Set()**, **SetStatus()** и **Clear()**, с помощью которых в **UpdateItemBuilder** записываются инструкции по изменению значений атрибутов позиции и изменению статуса. После записи всех необходимых инструкций в **UpdateItemBuilder**, необходимо выполнить изменение позиции с помощью метода **Invoke()**.

- **UpdateItemBuilder enrichUpdateItem(String itemUid)** – получить **UpdateItemBuilder** для записи инструкций на изменение значений атрибутов и статуса позиции для заданной позиции (**itemUid**);
- **UpdateItemBuilder** – позволяет сформировать список инструкций для изменения значений атрибутов и статуса:
    - **set(String value, Long propertyId)** – записать новое значение атрибута (**value**) в атрибут (**propertyId**). Получить идентификатор атрибута можно так: **@[имя атрибута].toId()**. Если в **value** будет передана пустая строка или **null**, то значение атрибута будет удалено, таким образом, данный вызов будет эквивалентен вызову функции **clear(Long propertyId)**. В **value** требуется передать строковое представление значения атрибута:
        - **строка**: строка;
        - **целое число**: строковое представление числа;
        - **число с плав. запятой**: строковое представление числа с плав. запятой и обозначением ЕИ при ее наличии. Примеры: “10 мм”, “10 – 20 кг”, 10(-0,05;0,06);
        - **дата**: строковое представление даты в формате: dd.MM.yyyy;

- **логический тип**: “true”/”false”;
- **цвет**: строковое представление int32, где первый, второй и третий байт кодируют цвет RGB;

- **clear(Long propertyId)** – удалить значение заданного атрибута (**propertyId**);
- **setStatus(ItemStatus value)** – задать статус (**value**) позиции:
  - Возможные значения **ItemStatus**:
    - **ItemStatus.Abnormal** (Не нормализована);
    - **ItemStatus.Standardized** (Нормализована);
    - **ItemStatus.Blocked** (Заблокирована);
    - **ItemStatus.Removed** (Удалёна);
    - **ItemStatus.Duplicate** (Дубликат);
- **invoke()** – выполнить изменение позиции в соответствии с записанными в **UpdateItemBuilder** инструкциями.

Вызов других постобработчиков

При выполнении изменений значений атрибутов позиции, при изменении статуса и создании новой позиции через постобработчик, другие постобработчики, заданные на изменение значения атрибута, изменение статуса или создание позиции не будут вызываться. При необходимости выполнить постобработчик из другого постобработчика, в системе предусмотрены следующие функции:

- **callEnrichCreateItem(String itemUid, String nodeUid)** – выполнить постобработчик на создание позиции из группы (**nodeUid**) для позиции (**itemUid**);
- **callEnrichChangeItemStatus(String itemUid, String nodeUid)** – выполнить постобработчик на изменение статуса позиции из группы (**nodeUid**) для позиции (**itemUid**);
- **callEnrichChangeValue(String itemUid, Long propertyId)** – выполнить постобработчик на изменение значения атрибута (**propertyId**) для позиции (**itemUid**).

Использование счетчиков для генерации кодов

В системе реализована возможность работы со счетчиками, для генерации кодов номенклатурных объектов, групп и прочего. Счетчик генерирует последовательно числа для заданного наименования счетчика и дополняет полученный код нулями до указанного количества символов. Для этого в системе реализован метод:

- **String getNextSequence(String name, int digits)** – получить следующее значение счетчика с именем name и дополнить его лидирующими нулями до указанного количества знаков в **digits**. Если счетчик с заданным наименованием **name** ранее отсутствовал, то будет автоматически создан, для которого будет сгенерирован нулевой номер.

Наличие данного метода позволяет организовать как сквозную нумерацию в рамках системы, путем использования одного единого наименования счетчика, так и локальную нумерацию в рамках любых классификационных групп, путем использования в качестве наименования счетчика, наименования или кода классификационной группы.

3.33.2.2 Методы для чтения данных из Semantic MDM

Выполнение поиска позиций

Возможны различные сценарии обогащения данных, при которых необходимо выполнить поиск позиции по заданным атрибутам в справочнике Semantic MDM. Например, найти в классификаторе группу по коду или найти материал по марке и стандарту на материал и далее прописать ссылку на найденную позицию.

Для реализации этих сценариев, в скрипт Groovy добавлены функции выполнения поиска позиции в заданной группе или в заданном классификаторе:

- **SearchParamsBuilder findInNode(String nodeUid)** – выполнить поиск в заданной группе (**nodeUid**) и вернуть первые 100 позиций;
- **SearchParamsBuilder findInNodes(List<String> nodeUids)** – выполнить

поиск в заданном списке групп (**nodeUids**) и вернуть первый 100 позиций;

- **SearchParamsBuilder findInClassifier(String nodeUid)** – выполнить поиск в заданной классификационной группе (**nodeUid**) классификатора и вернуть первые 100 позиций. Данный метод востребован, при необходимости выполнить поиск позиции, среди позиций, классифицированных группой отдельного классификатора, например группой классификатора: ОКВЭД2, ОКПД2, ЕКПС и т.д..

Данные функции возвращают построитель критериев поиска (**SearchParamsBuilder**) в котором можно с помощью функций **byValue(String value, String propertyId)** и **byStatus(ItemStatus status**) добавить параметры поиска и далее выполнить поиск с помощью функции **Invoke()**, которая вернет список **Uuid позиций(List<String>)**, найденных в Semantic MDM. Каждая из функций: **byValue()** и **byStatus()** возвращает построитель запросов **SearchParamsBuilder**, дополненный критерием по значению атрибута (**byValue**) или статуса (**byStatus**), что позволяет путем последовательного вызова этих функций сформировать требуемый поисковый запрос и выполнить его. Примеры вызова, ниже.

Пример поиска страны в классификаторе ОКСМ по наименованию и статусу “Нормализована” :

```
List<String> itemUids = support.findInNode("55805ab0-15cd-4523-9fc2-f926fcf3b376").byValue("РОССИЯ" , @[Наименование ОКСМ].toPath()).byStatus(ItemStatus.Standardized).invoke();
```

Пример поиска в справочнике материалов по марке материала, стандарту на материал и статусу “Нормализована”:

```
List<String> itemUids = support.findInNode("2003b902-7bee-49d5-9401-02fb79e1d6e9").byValue("Ст3кп", @[Марка материала].toPath()).byValue("ГОСТ 380-94", @[Стандарт на материал][Сокращенное наименование].toPath()).byStatus(ItemStatus.Standardized).invoke();
```

3.33.2.2.1 Чтение данных о позиции

В сценариях обогащения данных, могут встречаться задачи получения текущего статуса позиции, получения идентификатора внешней системы или идентификатора Semantic MDM из переходного ключа, а также получения информации о группе.

В специальной переменной **support** предусмотрены следующие функции:

- **@[#itemGuid]** – получить **guid** текущей обогащаемой позиции. Данная сигнатура используется без **support**;
- **ItemStatus getItemStatus(String itemUid)** – получить статус позиции по GUID (**ItemUid**) позиции. Функция вернет одно из следующих значений:
  - **NULL** - если позиция с переданным itemUid не существует;
  - I**temStatus.Abnormal** – соответствует статусу «Не нормализована»;
  - **ItemStatus.Standardized** – соответствует статусу «Нормализована»;
  - **ItemStatus.Blocked** – соответствует статусу «Заблокирована»;
  - **ItemStatus.Removed** – соответствует статусу «Удалёна»;
  - **ItemStatus.Duplicate** – соответствует статусу «Дубликат»;
- **String getMDMId(String sourceId, String systemId)** – получить GUID позиции по идентификатору объекта (**sourceId**) во внешней системе и идентификатору внешней системы (**systemId**). Вернет **NULL** если переходной ключ отсутствует;
- **String getSourceId(String itemUid, String systemId)** – получить идентификатор объекта во внешней системе по GUID позиции (**itemUid**) и идентификатору внешней системы (**systemId**). Если переходной ключ отсутствует, то вернет **NULL**;
- **String getItemNodeUid(String itemUid)** – получить GUID группы, которой принадлежит позиция с указанным GUID (**itemUid**). Если

позиция с переданным GUID не существует, то будет возвращена ошибка.

Пример получения идентификатора в системе 1C_ERP для обогащаемой позиции:

String itemUid = support.getSourceId(@[#itemUid], "1C_ERP")

3.33.2.2.2 Чтение значений атрибутов позиции

Основным способом чтения значений атрибутов текущей позиции, является чтение через конструкцию @[имя атрибута]. Для чтения значения вложенных в агрегации атрибутов, необходимо последовательно перечислить имена атрибутов, например:

- **@[имя атрибута связи][имя вложенного атрибута];**
- **@[имя атрибута связи][имя вложенного атрибута связи][имя вложенного атрибута];**
- **@[#composer][имя вложенного атрибута];**
- **@[#composer][имя вложенного атрибута связи][имя вложенного атрибута]**.

Перечисленные конструкции вернут одно значение атрибута, если все перечисленные в конструкции атрибуты хранят единичное значение. Если хотя бы один из перечисленных в конструкции атрибутов хранит массив значений, то данная конструкция вернет **List** значений.

Кроме получения списка значений (**List**) для вложенного атрибута, есть возможность получить полный путь до этого значения через метод: **getMapValues()**, который вернут **Map<String, Object>**, где **String** будет содержать GUID агрегированной позиции, а **Object** будет содержать либо значение заданного атрибута, либо еще один **Map<String, Object>**, если путь к вложенному атрибуту состоит из более чем одного атрибута связи.

Примеры возвращаемых значений:

- **@[имя множественного атрибута связи][имя вложенного строкового атрибута].getMapValues()** – данная конструкция вернет **Map<String, String>**

- **@[имя множественного атрибута связи] [имя единичного атрибута связи] [имя вложенного строкового атрибута].getMapValues()** – данная конструкция вернет **Map<String, Map<String, String>>**
- **@[имя множественного атрибута связи] [имя единичного атрибута связи] [имя вложенного множественного строкового атрибута].getMapValues()** – данная конструкция вернет **Map<String, Map<String, List<String>>>**

Для каждого из типов данных атрибутов, будет возвращен свой тип данных:

- **строка**: String
- **целое число**: Integer
- **число с плав. запятой**: valueDouble - методы для работы с **ValueDouble**
- **дата**: Date
- **логический тип**: Boolean
- **цвет**: Long. – число в формате Blue, Green, Red, Alpha, где Blue – старший байт, а Alpha – младший байт. Пример красного цвета в шестнадцатеричном строковом представлении: «241CEDFF», где: Blue=24, Green=1C, Red=ED, Alpha=FF.
    - Преобразование цвета из шестнадцатеричного строкового представления в число: Long.parseLong(«241CEDFF»,16);
    - Преобразование числа в строковое шестнадцатеричное представление: Long.toString(605875711,16);
- **файл**: String - имя файла
- **атрибут связи**: String - с guid позиции

Если основной способ чтения значений атрибутов текущей позиции по каким-либо причинам не позволяет организовать чтение требуемых данных или требуется прочитать значения атрибутов найденных позиций через поиск,

то для чтения значений атрибутов позиций, в системе предусмотрены следующие методы:

- **List<PropertyValueDto> getItemPropertyValues(String itemUid, List<long> PropertyIds)** – получить список значений атрибутов по GUID позиции (**itemUid**). Если атрибут содержит множественные значения, то для каждого значения будет возвращен свой **PropertyValueDto**, с одинаковым **PropertyValueDto.propertyId**.

  Свойства **PropertyValueDto**:

  - **String itemUi**d – идентификатор позиции
  - **long propertyId** – числовой идентификатор атрибута
  - **String stringValue** – содержит строковое значение атрибута, если тип данных: **String** или **Text**. Содержит GUID позиции, если тип данных **Aggregation** или **Composition**
  - **ExchangeComplexDouble doubleValue** – содержит число с плавающей запятой, если тип данных: **Double**.
  - **long integerValue** – содержит целое число, если тип данных **Int**
  - **ExchangeFile fileValue** – содержит наименование файла и хэш содержимого. Если тип данных **c**, то наименование файла будет пустым.

  Свойства **ExchangeFile**:

    - **name** – наименование файла, для изображения не указывается
    - **bodyHash** – хэш тела файла

  - **Boolean booleanValue** – содержит булевое значение, если тип данных **Bool**
  - **Date dateValue** – содержит дату, если тип данных **DateTime**
  - **String nameValue** – содержит наименование контекста для контекстных значений атрибута

Свойства **ExchangeComplexDouble**:

- **String getMeasurementUnit** – получить GUID единицы измерения Semantic MDM, в которой она должна отображаться в Semantic MDM
- **ValueDoubleTransfer getValue** – получить объект класса **ValueDoubleTransfer** со следующими методами:
    - **double value** – содержит основное значение атрибута в базовой единице измерения, если из внешней системы передается простое число с плав. запятой или число с отклонениями. Отклонения для числа с плав. запятой будут храниться в min и max соответственно. Для диапазона, должно содержать центральное значение диапазона, к которому при прибавлении значения из min, получаем нижнюю границу отклонения, а при прибавлении значения из max, верхнее отклонение
    - **double min** – получить значение нижнего отклонения
    - **double max** – получить значение верхнего отклонения
    - **encodedString** – строковое кодированное значение, содержит значения **value**, **min** и **max** в той ЕИ, в которой задал значение пользователь. Значение в encodedString кодируется следующим образом: **value**=<среднее число >; **min**=<нижнее отклонение, всегда меньше нуля>; **max**=<верхнее отклонение, всегда больше нуля>; **measure**=<числовой идентификатор ЕИ>; **precision**=<количество знаков после запятой для округления>; **type**=<тип ЕИ: “NORMAL”, “RANGE”, “DEFLECTIONS”>. Примеры кодированной строки **encodedString**:
        - **200 мм**:

“value=200;min=0;max=0;measure=313;precision=5;type=NORMAL”

• **300 – 400 мм**:

“value=0;min=300;max=400;measure=313;precision=5;type=RANGE”

• **10(-0.1;0.15) мм**:

“value=10;min=-0.1;max=0.15;measure=313;precision=5;type=DEFLECTIONS”

• **List<PropertyValueDto> getItemsPropertyValueList(List<String> itemUids, List<long> PropertyIds)** – получить список значений для атрибутов указанных в **PropertyIds** и позиций указанных в **itemUids**. Если **PropertyIds** содержит пустой список, то будут возвращены значения всех атрибутов. Для каждого **PropertyValueDto** в свойстве **itemUid** будет указан guid позиции, к которому относится возвращенное значение атрибута. Если атрибут содержит множественные значения, то для каждого значения будет возвращен свой **PropertyValueDto**, с одинаковым **PropertyValueDto.propertyId** и одинаковым **PropertyValueDto.itemUid**.

Метод **getItemPropertyValues** возвращает список всех значений атрибутов для заданной позиции (**itemUid**). Для чтения значений конкретного атрибута из полученного списка, необходимо знать идентификатор и тип данных атрибута, а также множественное у него значение или нет. Для получения этой информации в скрипте, необходимо использовать метод **getPropertyDeclarationsByNodeUid**.

Ниже представлены примеры чтения значений атрибутов.

Пример чтения значений атрибутов: «Наименование изделия», «Диаметр», «Материал->Марка материала»:

```
List<PropertyValueDto> itemValues = support.getItemPropertyValues("a1ddfe35-6fca-4f25-b874-604d082d9d3c");
//Получим значение атрибута "Наименование изделия"
PropertyValueDto attrValue = itemValues.find {element -> element.
```

```
propertyId()==@[Наименование изделия].toId()};
    if (attrValue!=null)
    {String prodName = attrValue.stringValue}
    //Получим значение атрибута "Диаметр" в миллиметрах
    attrValue = itemValues.find {element -> element.
propertyId()==@[Диаметр].toId()};
    If (attrValue!=null)
    //Получим диаметр в миллиметрах
    {Double prodDiam = doubleValue.getValue().value/1000}
    //Получим значение атрибута "Материал->Марка материала" в
миллиметрах
    attrValue = itemValues.find {element -> element.
propertyId()==@[Материал].toId()};
    if (attrValue!=null)
    {
    prodMatUid = attrValue.stringValue;
    List<PropertyValueDto> MaterialValues =
support.getItemPropertyValues(prodMatUid, false);
    attrValue = itemValues.find {element -> element. propertyId()==@[Марка
материала].toId()};
    if (attrValue!=null)
    {String prodMatMark = attrValue.stringValue}
    }
```

3.33.2.2.3 Чтение данных по атрибутам

- **PropertyInfoDto getPropertyInfoByUid(String propertyUid)** – получить информацию об атрибуте по GUID атрибута. Если атрибут с заданным **PropertyUid** не существует, то функция вернет **NULL**.

  Свойства **PropertyInfoDto**:

  - **long id** – числовой идентификатор атрибута$
  - **String uuid** – GUID атрибута;

- **String eOTDi**d – идентификатор OTD;
- **dataType** – тип данных атрибута:
    - **PropertyDataType.String** – строка;
    - **PropertyDataType.Double** – число с плав. запятой;
    - **PropertyDataType.Int** – целое число;
    - **PropertyDataType.Bool** – булевый;
    - **PropertyDataType.Image** – картинка;
    - **PropertyDataType.Text** – текст;
    - **PropertyDataType.DateTime** – дата;
    - **PropertyDataType.Colour** – цвет;
    - **PropertyDataType.File** – файл;
    - **PropertyDataType.Formula** – формула;
    - **PropertyDataType.Aggregation** – агрегация;
    - **PropertyDataType.Composition** – композиция;
- **Map<String, String> names** – возвращается карта «Идентификатор языка» – «Наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **Map<String, String> aggregateNames** – для атрибута связи возвращается карта «Идентификатор языка» – «Обратное наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **String commen**t– описание атрибута;
- **String sourceDocument** – значение свойства “Источник (документ)”;
- **boolean isMultiple** – флаг массив;
- **boolean isHierarchical** – флаг *Иерархический*;
- **boolean isContextual** – флаг контекстного атрибута;

- **List<PropertyInfoDto> getPropertyInfoByUidList(List<String>**

propertyUids) – получить информацию по заданному списку атрибутов. Если один из атрибутов с заданным **PropertyUid** не существует, то функция не вернет информации по данному атрибуту;

- **PropertyInfoDto getPropertyInfoByOtdId(String propertyOtdId)** – получить информацию об атрибуте по идентификатору OTD атрибута Semantic MDM. Если атрибут с заданным **propertyOtdId** не существует, то функция вернет **NULL**
- **List<PropertyInfoDto> getPropertyInfoByOtdIdList (List<String> propertyOtdIds)** – получить информацию по заданному списку атрибутов. Если один из атрибутов с заданным **PropertyOtdId** не существует, то функция не вернет информации по данному атрибуту
- **PropertyInfoDto getPropertyInfoById(Long propertyId)** – получить информацию об атрибуте по числовому идентификатору атрибута Semantic MDM. Если атрибут с заданным **PropertyId** не существует, то функция вернет ошибку.
- **List<PropertyInfoDto> getPropertyInfoByIdList (List< Long> propertyIds)** – получить информацию по заданному списку атрибутов. Если один из атрибутов с заданным **PropertyId** не существует, то функция вернет ошибку.
- **List<PropertyDeclarationInfoDto> getPropertyDeclarationsByNodeUid(String nodeUid, boolean resolveMultilinguals)** – получить список деклараций для группы с указанным гуид (**nodeUid**). Флаг **resolveMultilinguals** определяет необходимость получения данных о мультиязычных наименованиях в **PropertyInfoDto.Names**, **PropertyInfoDto.aggregateNames**, **PropertyDeclarationInfoDto.categories**.

  Свойства **PropertyDeclarationInfoDto:**

  - **List<String> scopeNodes** – список GUID групп, на позиции которых может ссылаться агрегация/композиция (**PropertyDeclarationInfoDto.dataType** равен **Aggregation** или

Composition);

- **Map<String, String> categories** – возвращается карта «Идентификатор языка» – «Наименование категории». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **long id** – числовой идентификатор атрибута;
- **String uuid** – GUID атрибута;
- **String eOTDid** – идентификатор OTD;
- **PropertyDataType dataType** – тип данных атрибута:
    - **PropertyDataType.String** – строка;
    - **PropertyDataType.Double** – число с плав. запятой;
    - **PropertyDataType.Int** – целое число;
    - **PropertyDataType.Bool** – булевый;
    - **PropertyDataType.Image** – картинка;
    - **PropertyDataType.Text** – текст;
    - **PropertyDataType.DateTime** – дата;
    - **PropertyDataType.Colour** – цвет;
    - **PropertyDataType.File** – файл;
    - **PropertyDataType.Formula** – формула;
    - **PropertyDataType.Aggregation** – агрегация;
    - **PropertyDataType.Composition** – композиция;
- **Map<String, String> Names** – возвращается карта «Идентификатор языка» – «Наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **Map<String, String> aggregateNames** – для атрибута связи возвращается карта «Идентификатор языка» – «Обратное наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;

- **String comment**– описание атрибута;
- **String sourceDocument** – массив значений «Источник (документ)»;
- **boolean isMultiple** – флаг массив;
- **boolean isHierarchical** – флаг Иерархический.

3.33.2.2.4 Чтение данных по группам

- **ClassifiableElementInfoDto getNodeInfo(String NodeUid, Boolean fetchParents)** – получить информацию о группе по ее гуиду (**NodeUid**) Semantic MDM. Флаг **fetchParents** позволяет вернуть информацию по всем родительским группам. Если группа с заданным **NodeUid** не существует, то функция вернет **NULL**

  Свойства **ClassifiableElementInfoDto**:

  - **Long id** – числовой идентификатор группы;
  - **String uid** – GUID группы;
  - **String displayName** – наименование группы;
  - **List<ClassifiableElementDto> parents** – список родительских групп, первой идет корневая группа, последней идет родительская группа для указанной группы (**NodeUid**).

3.33.2.3 Методы для работы с числами (valueDouble)

Для более удобной работы с числом с плавающей запятой (**valueDouble**), в **support** реализованы вспомогательные методы, позволяющие получить необходимые данные как для вычисления, так и для формирования строковых обозначений:

- **BigDecimal getDoubleValue(valueDouble value, Integer precision, BigDecimal koef)** – получить основное значение в базовой единице измерений, домноженное на коэффициент (**koef**) и округленное до заданного количества знаков (**precision**). Если **value** равно **null**, то функция вернет 0. В любом из параметров может быть передан **null**, в этом случае функция вернет число без домножения на коэффициент

(**koef**) и/или округления до заданного количества знаков (**precision**). Функция вернет значение только для **getValueType(): NORMAL**, **DEFLECTION**, в противном случае функция вернет ошибку;

- **BigDecimal getLeftDoubleValue(valueDouble value, Integer precision, BigDecimal koef)** – получить значение слева в базовой единице измерений, домноженное на коэффициент (**koef**) и округленное до заданного количества знаков (**precision**). Если **value** равно **null**, то функция вернет 0. В любом из параметров может быть передан **null**, в этом случае функция вернет число без домножения на коэффициент (**koef**) и/или округления до заданного количества знаков (**precision**). Функция вернет значение только для **getValueType: dtRange**, **dtDeflection**, в противном случае функция вернет ошибку;
- **BigDecimal getRightDoubleValue(valueDouble value, Integer precision, BigDecimal koef)** – получить значение справа в базовой единице измерений, домноженное на коэффициент (**koef**) и округленное до заданного количества знаков (**precision**). Если value равно **null**, то функция вернет 0. В любом из параметров может быть передан **null**, в этом случае функция вернет число без домножения на коэффициент (**koef**) и/или округления до заданного количества знаков (**precision**). Функция вернет значение только для **getValueType: dtRange**, **dtDeflection**, в противном случае функция вернет ошибку;
- **DoubleFormatType getValueType(valueDouble value)** – получить тип числа:
    - **NORMAL** – простое число;
    - **RANGE** – диапазон;
    - **DEFLECTION** – число с отклонениями;
- **Integer getEiPrecision(valueDouble value)** – получить количество знаков после запятой (точность), заданное пользователем при вводе. Если точность не определена, то вернуть **null**;
- **String getEiCode(valueDouble value)** – получить гуид единицы

измерения, в которой пользователь задал значение. Если ЕИ не определена для атрибута, то вернет **null**;

- **BigDecimal getEiKoef(valueDouble value)** – получить коэффициент пересчета в базовую ЕИ. Если коэффициент запрашивается для базовой ЕИ, то будет возвращена 1;
- **String getEiGuid(valueDouble value)** - получить гуид единицы измерения. Если **value==null**, то вернуть пустую строку;
- **BigDecimal getValueInBaseEi(BigDecimal value, String EiGuid)** - пересчитать значение (**value**) из заданной ЕИ (**EiGuid**) в базовую ЕИ. Если **EiGuid** является базовой или содержит null или пустую строку, то вернуть значение value без модификаций;
- **createDblSimpleValue(BigDecimal value, Integer precision, String eiGuid)** - создать **valueDouble** на основе переданного числа value, в заданной единице измерения (**eiGuid**), которое надо отображать пользователю с заданной точностью precision. Если точность не указана (**null**), то значение не будет округляться для пользователя. Если **eiGuid** не указана (**null**), то считается что значение **value** передается в базовой единице измерения;
- **createDblRangeValue(BigDecimal leftValue, BigDecimal rightValue, Integer precision, String eiGuid)** - создать **valueDouble** на основе переданного диапазона **leftValue** и **rightValue**, в заданной единице измерения (**eiGuid**), которые надо отображать пользователю с заданной точностью precision. Если точность не указана (null), то значение не будет округляться для пользователя. Если **eiGuid** не указана (**null**), то считается что значения **leftValue** и **rightValue** передаются в базовой единице измерения.
- **createDblDeflectionValue(BigDecimal value, leftValue, rightValue, Integer precision, String eiGuid)** - создать **valueDouble** на основе переданного числа **value** с отклонениями **leftValue** и **rightValue**, в заданной единице измерения (**eiGuid**), которые надо отображать

пользователю с заданной точностью **precision**. Если точность не указана (**null**), то значение и отклонения не будут округляться для пользователя. Если **eiGuid** не указана (**null**), то считается что значение **value** и отклонения **leftValue** и **rightValue** передаются в базовой единице измерения.

Методы ExchangeComplexDouble

- **String getMeasurementUnit** – получить гуид единицы измерения Semantic MDM, в которой она должна отображаться в Semantic MDM;
- **ValueDoubleTransfer getValue** – получить объект класса **ValueDoubleTransfer** со следующими методами:
  - **double value** – содержит основное значение атрибута в базовой единице измерения, если из внешней системы передается простое число с плав. запятой или число с отклонениями. Отклонения для числа с плав. запятой будут храниться в min и max соответственно. Для диапазона, должно содержать центральное значение диапазона, к которому при прибавлении значения из min, получаем нижнюю границу отклонения, а при прибавлении значения из max, верхнее отклонение;
  - **double min** – получить значение нижнего отклонения;
  - **double max** – получить значение верхнего отклонения;
  - **encodedString** – строковое кодированное значение, содержит значения value, min и max в той ЕИ, в которой задал значение пользователь. Значение в **encodedString** кодируется следующим образом: **value**=<среднее число >; **min**=<нижнее отклонение, всегда меньше нуля>; **max**=<верхнее отклонение, всегда больше нуля>; **measure**=<числовой идентификатор ЕИ>; **precision**=<количество знаков после запятой для округления>; **type**=<тип ЕИ: “NORMAL”, “RANGE”, “DEFLECTIONS”>. Примеры кодированной строки **encodedString**:
    - **200 мм**:

“value=200;min=0;max=0;measure=313;precision=5;type=NORMAL”;

- **300 – 400 мм**: “value=0;min=300;max=400;measure=313;precision=5;type=RANGE”;
- **10(-0.1;0.15) мм**: “value=10;min=-0.1;max=0.15;measure=313;precision=5;type=DEFLECTIONS”.

3.33.2.4 Методы для работы с XML и JSON форматом

В сценариях обогащения, могут встречаться задачи по получению данных в формате XML или JSON в произвольной структуре. Для работы с XML и JSON форматами, в Groovy-скрипт добавлена поддержка следующих библиотек:

- groovy.xml.XmlSlurper – для работы с XML: https://docs.groovy-lang.org/latest/html/gapi/index.html?overview-summary.html
- groovy.json.JsonSlurper – для работы с JSON: https://docs.groovy-lang.org/latest/html/gapi/index.html?groovy/json/JsonSlurper

Для работы с ними в Groovy-скрипт, требуется в первой строке скрипта выполнить импорт (**import**) соответствующей библиотеки.

Пример чтения данных из XML:

```
import groovy.xml.XmlSlurper


String getSampleDSEFullName() {
StringBuilder XMLText = new StringBuilder();
//Сформируем пример XML с описанием объекта ДСЕ из внешней системы
XMLText.append('<ExternalObject>')
    .append('<FullName>Штуцер 123.01.01.40</FullName>')
    .append('<Attributes>')
```

```
            .append('<Attribute name="Наименование">Штуцер</Attribute>')
            .append('<Attribute name="Обозначение">123.01.01.40</Attribute>')
            .append('<Attribute name="Базовая ЕИ"><Object><Attributes><Attribute name="Обозначение ЕИ">шт.</Attribute></Attributes></Object></Attribute>')
            .append('</Attributes>')
            .append('</ExternalObject>');
        //Выполним парсинг XML
        def rootNode = new XmlSlurper().parseText(XMLText.toString());
        //Извлечем значения атрибутов из присланного в XML объекта
        String FullName = rootNode.'FullName'.text();
        String DSECode = rootNode.'Attributes'.'*'.find {attribute -> attribute.'@name'=="Обозначение"}.text();
        String DSEName = rootNode.'Attributes'.'*'.find {attribute -> attribute.'@name'=="Наименование"}.text();
        String EICode = rootNode.'Attributes'.'*'.find {attribute -> attribute.'@name'=="Базовая ЕИ"}.'Object'.'Attributes'.'Attribute'[0].text();
        //Вернем наименование по формуле: "<Наименование> <Обозначение> (<Обозначение ЕИ>)"
        return DSEName + ' '+DSECode + ' ('+EICode+')';
        }
```

Пример чтения данных из JSON

```
        import groovy.json.JsonSlurper

        String getSampleDSEFullName () {
        StringBuilder XMLText = new StringBuilder();
        //Сформируем пример JSON с описанием объекта ДСЕ из внешней системы
        XMLText.append('{')
            .append('  "FullName": "Штуцер 123.01.01.40",')
```

```
            .append('   "Attributes": [')
            .append('     {')
            .append('       "name": "Наименование",')
            .append('       "value": "Штуцер"')
            .append('     },')
            .append('     {')
            .append('       "name": "Обозначение",')
            .append('       "value": "123.01.01.40"')
            .append('     },')
            .append('     {')
            .append('       "name": "Базовая ЕИ",')
            .append('       "Object": {')
            .append('         "Attributes": [')
            .append('           {')
            .append('             "name": "Обозначение ЕИ",')
            .append('             "value": "шт."')
            .append('           }')
            .append('         ]')
            .append('       }')
            .append('   ]')
            .append('}');
        //Выполним парсинг JSON
        def rootNode = new JsonSlurper().parseText(XMLText.toString());
        //Извлечем значения атрибутов из присланного в JSON объекта
        String FullName = rootNode.'FullName';
        String DSECode = rootNode.'Attributes'.find {it.'name'=="Обозначение"}.'value';
        String DSEName = rootNode.'Attributes'.find {it.'name'=="Наименование"}.'value';
```

```
String EICode = rootNode.'Attributes'.find {it.'name'=="Базовая ЕИ"}.'Object'.'Attributes'[0].'value';
//Вернем наименование по формуле: "<Наименование> <Обозначение> (<Обозначение ЕИ>)"
return DSEName + ' '+DSECode + ' ('+EICode+')';
```

3.33.2.5 Пример обогащения позиции данными, полученными из ML-сервиса

Данный пример наглядно демонстрирует решение следующих задач:

- подключение к ML-сервису и выполнение предсказания по наименованию позиции;
- чтение предсказанных данных и заполнение атрибутов позиции данными;
- поиск позиций во вспомогательных справочниках и запись ссылки на найденную позицию в атрибут связи.

Разработанный groovy-скрипт должен быть размещен в постобработчике на изменение значения атрибута «*Полное наименование*» справочника «*Единый справочник номенклатуры*».

Сценарий работы пользователя: пользователь создает новую позицию и в карточке новой позиции в атрибут «*Полное наименование»* вводит требуемое полное наименование и нажимает кнопку запуска пост-обработчика. Система выполняет предсказание через ML-сервис, выполняет автоматическое заполнение атрибутов новой позиции. Пользователь корректирует неверно предсказанные значения атрибутов, после чего проводит дозаполнение требуемых атрибутов и публикует пакет изменений.

Скрипт постобработчика

```
// Предсказание значений атрибутов по полному наименованию
import groovy.json.JsonSlurper
String postProcessValue(String value){
  // value - изменённое значение
```

```
// @[...] - атрибутика обрабатываемой позиции
// @[#currentuser][...] - атрибутика пользователя, выполняеющего функцию
// вернуть строку!=null, если требуется сигнализировать ошибку
String EI_uid=null;
String DoubleVal=null;

//Получим полное наименование, которое ввел пользователь
String FullName = value;
 // Получить из сервиса машинного обучения предсказанные значения атрибутов, на основе полного наименование объекта(FULL_NAME)
HttpURLConnection connection = new URL("https://ml.sdi-solution.ru/extract1?full_name=" +
 ru.sdi.m2.resources.util.StringUtil.urlEncod
e(FullName)).openConnection() as HttpURLConnection;
 //Выполним парсинг JSON, с результатом предсказания
def rootNode = new JsonSlurper().parseText(connection.inputStream.text);
 //Получим гуид группы, в которой требуется создать позицию
//гуид группы предсказывается атрибутом attr_name=="groupId". Идентификатор даннаго атрибута attr_id='31415926'
String NodeUid = rootNode.extracted_attrs.find {it.attr_name=="groupId"}.attr_value;
//Убедимся что предсказанная группа и текущая группа совпадают
String curNodeUid = support.getItemNodeUid(scriptData.getItemUid(null));
if (NodeUid != curNodeUid)
 return "Предсказанная машинным обучением группа не совпадает с текущей. Выберите группу с GUID: "+NodeUid;
```

```
//Получим список продекларированных атрибутов в предсказанной группе
List<PropertyDeclarationInfoDto> nodeDeclarations = support.getPropertyDeclarationsByNodeUid(NodeUid,false);
//Получим построитель запроса на изменение значений текущей позиции
UpdateItemBuilder itemValueUpdates = support.enrichUpdateItem(scriptData.getItemUid(null));

List<PropertyDeclarationInfoDto> processedDeclarations = [];

//Пройдемся в цикле по извлеченным атрибутам
for (int i = 1; i < rootNode.extracted_attrs.size(); i++) {
 String PropPath = rootNode.extracted_attrs[i].attr_id;
 String PropValue = rootNode.extracted_attrs[i].attr_value;

 //Атрибут "groupId" с идентификатором '31415926' игнорируем, т.к. обработали его ранее
 if (PropPath!='31415926')
 {
  //Если атрибут связь
  int delimPos=PropPath.indexOf('->');
  if (delimPos>0)
  {
   //Получим гуид агрегации(атрибут связи) из пути
   String AggrPropUid = PropPath.substring(0,delimPos);

   PropertyDeclarationInfoDto attrDeclaration = nodeDeclarations.find {element -> element.uuid == AggrPropUid};
   //Если агрегация с таким гуид не существует, то выдаем ошибку
```

```
            if (attrDeclaration == null)
              continue; //return 'не удалось найти декларацию атрибута связи с гуид '"+AggrPropUid+"';

            //Для агрегации может быть предсказано несколько вложенных атрибутов,
            //поэтому одну и ту же корневую
            if (!processedDeclarations.contains(attrDeclaration))
            {
              processedDeclarations.add(attrDeclaration);
              //Найдем все атрибуты, для обрабатываемой агрегации
              def allAggrs = rootNode.extracted_attrs.findAll {it.attr_id.contains(AggrPropUid)}
              //Получим область поиска агрегации
              List<String> aggrScope = attrDeclaration.scopeNodes;
              if (attrDeclaration.scopeNodes.size()==0)
                return 'Область для атрибута связи '"+allAggrs[0].attr_name.subString(0,allAggrs[0].attr_name.indexOf('->'))+"' не задана';
              String aggrItemUid = getAggregationItemUid(rootNode, AggrPropUid, aggrScope);

              //Зададим новое значение для атрибута связи
              if (aggrItemUid!=null)
                itemValueUpdates.set(aggrItemUid, attrDeclaration.id)
              else
                itemValueUpdates.clear(attrDeclaration.id);
                //return 'Не удалось найти агрегируемую позицию '"+PropValue+"' для атрибута '"+allAggrs[0].attr_name.substring(0,allAggrs[0].attr_name.indexOf('->'))+"';
```

```
                }
            }
            //Если атрибут простой
            else
            {
              //Проверим наличие атрибута, если атрибут не декларирован, то выдаем ошибку
              PropertyDeclarationInfoDto attrDeclaration = nodeDeclarations.find {element -> element.uuid == PropPath};
              if ((attrDeclaration==null)||(attrDeclaration.uuid=='98fb1b1d-89e6-475c-b2fb-64e6b5424eaa'))
                continue; //return 'Не удалось найти атрибут с гуид "'+PropPath+'"';

              //Зададим новое значение для простого атрибута
              if ((PropValue!=null)&&(PropValue!=""))
                itemValueUpdates.set(PropValue, attrDeclaration.id);
              else
                itemValueUpdates.clear(attrDeclaration.id);
            }
          }
        }
        //Выполним изменение позиции
        itemValueUpdates.invoke();

        return null;
    }

    //Найти агрегированную позицию по вложенным атрибутам и вернуть ее гуид
    //rootNode - JSON с ответом подсистемы машинного обучения
```

```
//aggrPath - путь к агрегации из идентификаторов, для которой нужно найти агрегируемую позицию
//aggrAttrId - числовой идентификатор атрибута
String getAggregationItemUid(def rootNode, String aggrPath, List<String> aggrScope) {
 //Проверим что область агрегации где будем искать не пуста
 if (aggrScope.size()==0)
  throw new Exception('не определена область для агрегации "'+aggrPath+'"');

 //Получим список вложенных в агрегацию атрибутов
 def aggrAttrs = rootNode.extracted_attrs.findAll {it.attr_id.contains(aggrPath)};
 //Здесь будем хранить список обработанных вложенных агрегаций
 List<String> aggrProcessed = [];
 //Соберем поисковый запрос
 SearchParamsBuilder SearchBuilder=support.findInNodes(aggrScope);
 for (int i = 0; i < aggrAttrs.size(); i++) {
  //Получим полный путь к вложенному атрибуту
  String attrPath = aggrAttrs[i].attr_id;
  //Удалим из пути во вложенный атрибут, путь до агрегации
  String NestedAttr = attrPath.minus(aggrPath+'->');
  if ((aggrAttrs[i].attr_value != '')&&(aggrAttrs[i].attr_value != null)) {
   //Если во вложенном атрибуте есть еще вложенные атрибуты, то обработаем их
   if (NestedAttr.indexOf('->')>0)
   {
    String nestedAggrPropUid = NestedAttr.subString(0,NestedAttr.indexOf('->'));
    if (!aggrProcessed.contains(nestedAggrPropUid))
```

```
            {
              //Получим числовой идентификатор свойства
              Long nestedAggrPropId = support.getPropertyId(nestedAggrPropUid);
              //Получим область поиска агрегации
              List<String> nestedAggrScope =
support.getPropertyScopeInNode(nestedAggrPropId,root.getMdmNodeId());
              //Получим агрегируемую позицию, найденную по вложенным
атрибутам
              String AggregationItemUid=getAggregationItemUid(rootNode,
aggrPath+'->'+nestedAggrPropUid, nestedAggrScope);
              //Добавляем критерий поиска
              if (AggregationItemUid!=null)
                SearchBuilder = SearchBuilder.byValue(AggregationItemUid,
nestedAggrPropId);
              else
                throw new Exception('не удалось найти позицию для агрегации
'"+aggrAttrs[i].nestedAggrPropUid+"");
            }
          }
        //Если вложенный атрибут является простым атрибутом и значение не
пустое, то добавим его в критерии поиска
          else
          {
            PropertyInfoDto searchedProp =
support.getPropertyInfoByUid(NestedAttr);
            String searchedVal = aggrAttrs[i].attr_value.toString();
            //Добавляем критерий поиска
            SearchBuilder = SearchBuilder.byValue(searchedVal,
searchedProp.id.toString());
          }
```

```
        }
          //Выполним поиск по сформированным критериям
        List<String> itemUids =
SearchBuilder.byStatus(ItemStatus.Standardized).invoke();
        //Если нашлась одна единственная позиция, то возвращаем ее
        if (itemUids.size()==1)
          return itemUids[0];
      }
      //Вернем пустоту, если ничего не нашли
      return null;
    }
```

3.33.2.6 Пример обогащения позиции данными, полученными из Контур.Фокус

Данный пример наглядно демонстрирует решение следующих задач:

- подключение к веб-сервису Контур.Фокус и получение данных по контрагенту;
- чтение полученных из Контур.Фокус данных и заполнение атрибутов контрагента данными;
- синхронизация списков, полученных из Контур.Фокуса со массивом позиций в атрибуте связи.

Разработанный groovy-скрипт должен быть размещен в постобработчике на изменение значения атрибута «*ИНН*» *справочника «Контрагенты»*.

Сценарий работы пользователя: пользователь создает новую позицию и в карточке новой позиции, в атрибут «*ИНН*» вводит ИНН и нажимает кнопку запуска пост-обработчика. Система выполняет запрос данных по ИНН в *Контур.Фокус*, после чего выполняет автоматическое заполнение атрибутов новой позиции полученными данными из *Контур.Фокус*. Пользователь проводит дозаполнение требуемых атрибутов и публикует пакет изменений.

Скрипт постобработчика

```
// Получение данных по контрагенту из Контур.Фокус по ИНН
import groovy.json.JsonSlurper
String postProcessValue(String value){
    // value - изменённое значение
    // @[...] - атрибутика обрабатываемой позиции
    // @[#currentuser][...] - атрибутика пользователя, выполняеющего функцию
    // вернуть строку!=null, если требуется сигнализировать ошибку
    // Получить из сервиса машинного обучения предсказанные значения атрибутов, на основе полного наименование объекта(FULL_NAME)
    HttpURLConnection connection = new URL("https://focus-api.kontur.ru/api3/req?key=3208d29d15c507395db770d0e65f3711e40374df&inn=" + value
                                    ).openConnection() as HttpURLConnection;

    //Выполним парсинг JSON, с результатом предсказания
    def rootNodes = new JsonSlurper().parseText(connection.inputStream.text);
    def rootNode = rootNodes[0];

    //Получим построитель запроса на изменение значений текущей позиции
    UpdateItemBuilder itemValueUpdates = support.enrichUpdateItem(scriptData.getItemUid(null));
    String stringValue = "";

    //Получим экспресс-статус
    String ca_statement = "информация об экспрес-статусе отсутствует";
    stringValue = rootNode.briefReport.summary.greenStatements;
```

```
if (stringValue == "true")
  ca_statement = "С данным контрагентом можно заключать договора";
stringValue = rootNode.briefReport.summary.yellowStatements;
if (stringValue == "true")
  ca_statement = "Данный контрагент либо прекратил свою деятельность, либо скоро может прекратить. Требуется проверка контрагента.";
stringValue = rootNode.briefReport.summary.redStatements;
if (stringValue == "true")
  ca_statement = "Данного контрагента нужно обязательно проверить";
itemValueUpdates.set(ca_statement, @[Экспресс статус].toId())

//Добавим "Нормализован через ЕГРЮЛ/ЕГРИП"
itemValueUpdates.set("true", @[Нормализован через ЕГРЮЛ/ЕГРИП].toId());
//Добавим страну регистрации РОССИЯ
itemValueUpdates.set("p8iGlqNQ7dTbAqMjj1Jcxa", @[Страна регистрации (ОКСМ)].toId());

//Дата закрытия
stringValue = rootNode.UL.dissolutionDate;
if ((stringValue!=null)&&(stringValue!=""))
  itemValueUpdates.set(stringValue, @[Дата закрытия].toId());
else
  itemValueUpdates.clear(@[Дата закрытия].toId());

//КПП
stringValue = rootNode.UL.kpp;
//return stringValue
if ((stringValue!=null)&&(stringValue!=""))
```

```
            itemValueUpdates.set(stringValue, @[КПП].toId());
          else
            itemValueUpdates.clear(@[КПП].toId());

          //Полное наименование контрагента
          stringValue = rootNode.UL.legalName.full;
          if ((stringValue!=null)&&(stringValue!=""))
            itemValueUpdates.set(stringValue, @[Полное наименование
контрагента].toId());
          else
            itemValueUpdates.clear(@[Полное наименование
контрагента].toId());

          //Сокращённое наименование контрагента
          stringValue = rootNode.UL.legalName.short;
          if ((stringValue!=null)&&(stringValue!=""))
            itemValueUpdates.set(stringValue, @[Сокращённое наименование
контрагента].toId());
          else
            itemValueUpdates.clear(@[Сокращённое наименование
контрагента].toId());

          //@[Код ОКАТО]
          stringValue = rootNode.UL.okato;
          if ((stringValue!=null)&&(stringValue!=""))
            itemValueUpdates.set(stringValue, @[Код ОКАТО].toId());
          else
            itemValueUpdates.clear(@[Код ОКАТО].toId());

          //@[Организационно-правовая форма]
```

```
stringValue = rootNode.UL.opf;
if ((stringValue!=null)&&(stringValue!=""))
  itemValueUpdates.set(stringValue, @[Организационно-правовая форма].toId());
else
  itemValueUpdates.clear(@[Организационно-правовая форма].toId());

//@[Дата регистрации]
/*stringValue = rootNode.UL.registrationDate;
if ((stringValue!=null)&&(stringValue!=""))
  itemValueUpdates.set(stringValue, @[Дата регистрации].toId());
else
  itemValueUpdates.clear(@[Дата регистрации].toId());*/

//@[Статус контрагента]
stringValue = rootNode.UL.status.statusString;
if ((stringValue!=null)&&(stringValue!=""))
  itemValueUpdates.set(stringValue, @[Статус контрагента].toId());
else
  itemValueUpdates.clear(@[Статус контрагента].toId());

//@[ОГРН/Регистрационный номер в стране регистрации]
stringValue = rootNode.ogrn;
if ((stringValue!=null)&&(stringValue!=""))
  itemValueUpdates.set(stringValue, @[ОГРН/Регистрационный номер в стране регистрации].toId());
else
  itemValueUpdates.clear(@[ОГРН/Регистрационный номер в стране регистрации].toId());
```

```
//@[ИНН/Налоговый номер в стране регистрации]
stringValue = value;
if ((stringValue!=null)&&(stringValue!=""))
  itemValueUpdates.set(stringValue, @[ИНН/Налоговый номер в стране регистрации].toId());
else
  itemValueUpdates.clear(@[ИНН/Налоговый номер в стране регистрации].toId());

//@[ОКОПФ]
stringValue = rootNode.UL.okopf;
List<String> itemUids = support.findInNode("20cde407-f1a7-41f3-8d16-6df887f2c119").byValue(stringValue,@[Код ОКОПФ].toPath()).invoke();
//return stringValue
if (itemUids.size()>0)
  itemValueUpdates.set(itemUids[0], @[ОКОПФ].toId());
else
  itemValueUpdates.clear(@[ОКОПФ].toId());

//@[ОКФС]
stringValue = rootNode.UL.okfs;
itemUids = support.findInNode("ad59292a-1ef9-42c3-8373-2b099158b648").byValue(stringValue,@[Код ОКФС].toPath()).invoke();
//return stringValue
if (itemUids.size()>0)
  itemValueUpdates.set(itemUids[0], @[ОКФС].toId());
else
  itemValueUpdates.clear(@[ОКФС].toId());

//ЗАПОЛНИМ АТРИБУТЫ АДРЕСА
```

```
            if (rootNode.UL.legalAddress.parsedAddressRF != null) {
              stringValue = @[Адрес (место нахождения)];
              if (stringValue==null)
                stringValue = support.enrichCreateItem("c63b55da-eda9-4e8b-9ff5-e32f79126b5f");
              itemValueUpdates.set(stringValue, @[Адрес (место нахождения)].toId());
              //Получим построитель запроса на изменение значений позиции с адресом
              UpdateItemBuilder itemAggValueUpdates = support.enrichUpdateItem(stringValue);
              if (rootNode.UL.legalAddress.parsedAddressRF.bulk!=null) {
                //@[Тип строения]
                stringValue = rootNode.UL.legalAddress.parsedAddressRF.bulk.topoShortName;
                if ((stringValue!=null)&&(stringValue!=""))
                  itemAggValueUpdates.set(stringValue, @[Тип строения].toId());
                else
                  itemAggValueUpdates.clear(@[Тип строения].toId());
                //@[Строение]
                stringValue = rootNode.UL.legalAddress.parsedAddressRF.bulk.topoValue;
                if ((stringValue!=null)&&(stringValue!=""))
                  itemAggValueUpdates.set(stringValue, @[Строение].toId());
                else
                  itemAggValueUpdates.clear(@[Строение].toId());
              }
              else {
                itemAggValueUpdates.clear(@[Тип строения].toId());
                itemAggValueUpdates.clear(@[Строение].toId());
```

```
        }

        if (rootNode.UL.legalAddress.parsedAddressRF.city!=null) {
         //@[Тип города]
         stringValue                                                =
rootNode.UL.legalAddress.parsedAddressRF.city.topoShortName;
         if ((stringValue!=null)&&(stringValue!=""))
          itemAggValueUpdates.set(stringValue, @[Тип города].toId());
         else
          itemAggValueUpdates.clear(@[Тип города].toId());
         //@[Город]
         stringValue                                                =
rootNode.UL.legalAddress.parsedAddressRF.city.topoValue;
         if ((stringValue!=null)&&(stringValue!=""))
          itemAggValueUpdates.set(stringValue, @[Город].toId());
         else
          itemAggValueUpdates.clear(@[Город].toId());
        }
        if (rootNode.UL.legalAddress.parsedAddressRF.city!=null) {
         //@[Тип района]
         stringValue                                                =
rootNode.UL.legalAddress.parsedAddressRF.district.topoShortName;
         if ((stringValue!=null)&&(stringValue!=""))
          itemAggValueUpdates.set(stringValue, @[Тип района].toId());
         else
          itemAggValueUpdates.clear(@[Тип района].toId());
         //@[Район]
         stringValue                                                =
rootNode.UL.legalAddress.parsedAddressRF.district.topoValue;
         if ((stringValue!=null)&&(stringValue!=""))
```

```
            itemAggValueUpdates.set(stringValue, @[Район].toId());
          else
            itemAggValueUpdates.clear(@[Район].toId());
        }
        if (rootNode.UL.legalAddress.parsedAddressRF.house!=null) {
          //@[Тип дома]
          stringValue                                                    =
rootNode.UL.legalAddress.parsedAddressRF.house.topoShortName;
          if ((stringValue!=null)&&(stringValue!=""))
            itemAggValueUpdates.set(stringValue, @[Тип дома].toId());
          else
            itemAggValueUpdates.clear(@[Тип дома].toId());
          //@[Дом]
          stringValue                                                    =
rootNode.UL.legalAddress.parsedAddressRF.house.topoValue;
          if ((stringValue!=null)&&(stringValue!=""))
            itemAggValueUpdates.set(stringValue, @[Дом].toId());
          else
            itemAggValueUpdates.clear(@[Дом].toId());
        }

        if (rootNode.UL.legalAddress.parsedAddressRF.regionName!=null) {
          //@[Тип региона]
          stringValue                                                    =
rootNode.UL.legalAddress.parsedAddressRF.regionName.topoShortName;
          if ((stringValue!=null)&&(stringValue!=""))
            itemAggValueUpdates.set(stringValue, @[Тип региона].toId());
          else
            itemAggValueUpdates.clear(@[Тип региона].toId());
          //@[Регион]
```

```
            stringValue                                                =
rootNode.UL.legalAddress.parsedAddressRF.regionName.topoValue;
            if ((stringValue!=null)&&(stringValue!=""))
              itemAggValueUpdates.set(stringValue, @[Регион].toId());
            else
              itemAggValueUpdates.clear(@[Регион].toId());
          }
          if (rootNode.UL.legalAddress.parsedAddressRF.settlement!=null) {
          //@[Тип населённого пункта]
          stringValue                                                =
rootNode.UL.legalAddress.parsedAddressRF.settlement.topoShortName;
          if ((stringValue!=null)&&(stringValue!=""))
            itemAggValueUpdates.set(stringValue,        @[Тип        населённого
пункта].toId());
          else
            itemAggValueUpdates.clear(@[Тип населённого пункта].toId());
          //@[Населённый пункт]
          stringValue                                                =
rootNode.UL.legalAddress.parsedAddressRF.settlement.topoValue;
          if ((stringValue!=null)&&(stringValue!=""))
            itemAggValueUpdates.set(stringValue, @[Населённый пункт].toId());
          else
            itemAggValueUpdates.clear(@[Населённый пункт].toId());
          }
          if (rootNode.UL.legalAddress.parsedAddressRF.street!=null) {
          //@[Тип улицы]
          stringValue                                                =
rootNode.UL.legalAddress.parsedAddressRF.street.topoShortName;
          if ((stringValue!=null)&&(stringValue!=""))
            itemAggValueUpdates.set(stringValue, @[Тип улицы].toId());
```

```
            else
              itemAggValueUpdates.clear(@[Тип улицы].toId());
            //@[Улица]
            stringValue                                                    =
rootNode.UL.legalAddress.parsedAddressRF.street.topoValue;
            if ((stringValue!=null)&&(stringValue!=""))
              itemAggValueUpdates.set(stringValue, @[Улица].toId());
            else
              itemAggValueUpdates.clear(@[Улица].toId());
            }
            //@[Почтовый индекс]
            stringValue = rootNode.UL.legalAddress.parsedAddressRF.zipCode;
            if ((stringValue!=null)&&(stringValue!=""))
              itemAggValueUpdates.set(stringValue, @[Почтовый индекс].toId());
            else
              itemAggValueUpdates.clear(@[Почтовый индекс].toId());
            //Выполним изменение позиции с адресом
            itemAggValueUpdates.invoke();
           }

           def aggrItemsValues = [:];
           List<PropertyValueDto> propValues;
           PropertyValueDto propValue;
           String FindeditemUid;
           List<String> aggrItems;
           //ОБРАБОТАЕМ: Лица, имеющих право подписи
           aggrItems = scriptData.getValues(@[Лицо, имеющее право подписи]);
           for(String itemUid in aggrItems) {
            propValues = support.getItemPropertyValues(itemUid);
            aggrItemsValues.put(itemUid, propValues);
```

```
        }
        int ulHeadCount = rootNode.UL.heads.size();
        for(int i = 0; i < ulHeadCount; i++) {
         if ((rootNode.UL.heads[i]!=null)) {
          headDateValue = rootNode.UL.heads[i].date;
          headFioValue = rootNode.UL.heads[i].fio;
          headInnflValue = rootNode.UL.heads[i].innfl;
          headPositionValue = rootNode.UL.heads[i].position;
          FindeditemUid = null;
          for(String itemUid in aggrItems) {
           propValues = aggrItemsValues.get(itemUid);
           propValue = propValues.find {element -> element.propertyId==@[ФИО лица имеющего право подписи].toId()};
           if ((propValue!=null)&&(propValue.stringValue==headFioValue)) {
            FindeditemUid = itemUid;
            aggrItems.remove(aggrItems.findIndexOf {it==itemUid});
           }
          }
          if (FindeditemUid == null) {
           FindeditemUid = support.enrichCreateItem("6628f5b7-5297-48f6-a897-a3f11f16bf70");
           itemValueUpdates.set(FindeditemUid, @[Лицо, имеющее право подписи].toId());
          }
          //Получим построитель запроса на изменение значений позиции с адресом
          itemAggValueUpdates = support.enrichUpdateItem(FindeditemUid);
          //new SimpleDateFormat("yyyy-MM-dd").parse(headDateValue)
          if ((headDateValue!=null)&&(headDateValue!=""))
           itemAggValueUpdates.set(headDateValue, @[Дата внесения
```

```
изменений ЕГРЮЛ].toId());
            else
              itemAggValueUpdates.clear(@[Дата внесения изменений
ЕГРЮЛ].toId());
            if ((headInnflValue!=null)&&(headInnflValue!=""))
              itemAggValueUpdates.set(headInnflValue, @[ИНН/Налоговый
номер в стране регистрации].toId());
            else
              itemAggValueUpdates.clear(@[ИНН/Налоговый номер в стране
регистрации].toId());
            if ((headPositionValue!=null)&&(headPositionValue!=""))
              itemAggValueUpdates.set(headPositionValue, @[Должность].toId());
            else
              itemAggValueUpdates.clear(@[Должность].toId());
           //Выполним изменение позиции с лицом имеющим право подписи
           itemAggValueUpdates.invoke();
           }
         }
         //Удалим оставшиеся позиции, отсутствующие в списке
         for(String itemUid in aggrItems) {
           support.enrichDeleteItem(itemUid);
         }
         aggrItems.clear();

         //ОБРАБОТАЕМ: Управляющие компании
         aggrItems = scriptData.getValues(@[Управляющие компании]);
         for(String itemUid in aggrItems) {
           propValues = support.getItemPropertyValues(itemUid);
           aggrItemsValues.put(itemUid, propValues);
         }
```

```
            int ulMCCount = rootNode.UL.managementCompanies.size();
            //return rootNode.UL.managementCompanies[0];//.isEmpty()
            for(int i = 0; i < ulMCCount; i++) {
              if (rootNode.UL.managementCompanies[i]!=null) {
                mcDateValue = rootNode.UL.managementCompanies[i].date;
                mcInnValue = rootNode.UL.managementCompanies[i].inn;
                mcNameValue = rootNode.UL.managementCompanies[i].name;
                mcOgrnValue = rootNode.UL.managementCompanies[i].ogrn;
                FindeditemUid = null;
                for(String itemUid in aggrItems) {
                  propValues = aggrItemsValues.get(itemUid);
                  if ((mcInnValue == null)||(mcInnValue == ""))  {
                    propValue = propValues.find {element ->
element.propertyId==@[Наименование контрагента].toId()};
                    if  ((propValue!=null)&&(propValue.stringValue==mcNameValue))
{
                      FindeditemUid = itemUid;
                      aggrItems.remove(aggrItems.findIndexOf {it==itemUid});
                    }
                  }
                  else {
                    propValue = propValues.find {element ->
element.propertyId==@[ИНН/Налоговый номер в стране регистрации].toId()};
                    if ((propValue!=null)&&(propValue.stringValue==mcInnValue)) {
                      FindeditemUid = itemUid;
                      aggrItems.remove(aggrItems.findIndexOf {it==itemUid});
                    }
                  }
                }
                if (FindeditemUid == null) {
```

```
            FindeditemUid = support.enrichCreateItem("da926515-7cc9-4381-9ce7-abc70a24315e");
            itemValueUpdates.set(FindeditemUid, @[Управляющие компании].toId());
        }
        //Получим построитель запроса на изменение значений позиции с адресом
        itemAggValueUpdates = support.enrichUpdateItem(FindeditemUid);
        //new SimpleDateFormat("yyyy-MM-dd").parse(headDateValue)
        if ((mcDateValue!=null)&&(mcDateValue!=""))
            itemAggValueUpdates.set(mcDateValue, @[Дата внесения изменений ЕГРЮЛ].toId());
        else
            itemAggValueUpdates.clear(@[Дата внесения изменений ЕГРЮЛ].toId());
        if ((mcInnValue!=null)&&(mcInnValue!=""))
            itemAggValueUpdates.set(mcInnValue, @[ИНН/Налоговый номер в стране регистрации].toId());
        else
            itemAggValueUpdates.clear(@[ИНН/Налоговый номер в стране регистрации].toId());
        if ((mcNameValue!=null)&&(mcNameValue!=""))
            itemAggValueUpdates.set(mcNameValue, @[Наименование контрагента].toId());
        else
            itemAggValueUpdates.clear(@[Наименование контрагента].toId());
        if ((mcOgrnValue!=null)&&(mcOgrnValue!=""))
            itemAggValueUpdates.set(mcOgrnValue, @[ОГРН/Регистрационный номер в стране регистрации].toId());
        else
```

```
            itemAggValueUpdates.clear(@[ОГРН/Регистрационный номер в стране регистрации].toId());
            //Выполним изменение позиции с управляющей компанией
            itemAggValueUpdates.invoke();
          }
        }
        //Удалим оставшиеся позиции, отсутствующие в списке
        for(String itemUid in aggrItems) {
          support.enrichDeleteItem(itemUid);
        }
        aggrItems.clear();
          //ОБРАБОТАЕМ: Историю КПП
        List<String> KPPs = [];
        aggrItems = scriptData.getValues(@[История КПП]);
        for(String itemUid in aggrItems) {
          propValues = support.getItemPropertyValues(itemUid);
          aggrItemsValues.put(itemUid, propValues);
        }
        int ulKppCount = rootNode.UL.history.kpps.size();
        for(int i = 0; i < ulKppCount; i++) {
          if (rootNode.UL.history.kpps[i]!=null) {
            kppDateValue = rootNode.UL.history.kpps[i].date;
            kppValue = rootNode.UL.history.kpps[i].kpp;
            FindeditemUid = null;
            for(String itemUid in aggrItems) {
              propValues = aggrItemsValues.get(itemUid);
              propValue = propValues.find {element -> element.propertyId==@[КПП].toId()};
              if ((propValue!=null)&&(propValue.stringValue==kppValue)) {
                FindeditemUid = itemUid;
```

```
                aggrItems.remove(aggrItems.findIndexOf {it==itemUid});
              }
            }
            if (FindeditemUid == null) {
              FindeditemUid = support.enrichCreateItem("f04a053e-04a0-441e-908b-830fbf695ca4");
              itemValueUpdates.set(FindeditemUid, @[История КПП].toId());
            }
            else return FindeditemUid;
            //Получим построитель запроса на изменение значений позиции с адресом
            itemAggValueUpdates = support.enrichUpdateItem(FindeditemUid);
            //new SimpleDateFormat("yyyy-MM-dd").parse(headDateValue)
            if ((kppDateValue!=null)&&(kppDateValue!=""))
              itemAggValueUpdates.set(kppDateValue, @[Дата внесения изменений ЕГРЮЛ].toId());
            else
              itemAggValueUpdates.clear(@[Дата внесения изменений ЕГРЮЛ].toId());
            if ((kppValue!=null)&&(kppValue!=""))
              itemAggValueUpdates.set(kppValue, @[КПП].toId());
            else
              itemAggValueUpdates.clear(@[КПП].toId());
            //Выполним изменение позиции с управляющей компанией
            itemAggValueUpdates.invoke();
          }
        }
        //Удалим оставшиеся позиции, отсутствующие в списке
        for(String itemUid in aggrItems) {
          support.enrichDeleteItem(itemUid);
```

```
        }
        aggrItems.clear();

        //ОБРАБОТАЕМ: Историю наименований контрагента
        aggrItems = scriptData.getValues(@[История наименований]);
        for(String itemUid in aggrItems) {
          propValues = support.getItemPropertyValues(itemUid);
          aggrItemsValues.put(itemUid, propValues);
        }
        int ulNmCount = rootNode.UL.history.legalNames.size();
        for(int i = 0; i < ulNmCount; i++) {
          if (rootNode.UL.history.legalNames[i]!=null) {
            nmDateValue = rootNode.UL.history.legalNames[i].date;
            nmFullValue = rootNode.UL.history.legalNames[i].full;
            nmReadableValue = rootNode.UL.history.legalNames[i].readable;
            nmShortValue = rootNode.UL.history.legalNames[i].short;
            FindeditemUid = null;
            for (String itemUid in aggrItems) {
              propValues = aggrItemsValues.get(itemUid);
              propValue = propValues.find {element ->
element.propertyId==@[Дата внесения изменений ЕГРЮЛ].toId()};
              if ((propValue!=null)&&(propValue.stringValue==nmDateValue)) {
                FindeditemUid = itemUid;
                aggrItems.remove(aggrItems.findIndexOf {it==itemUid});
              }
            }
            if (FindeditemUid == null) {
              FindeditemUid = support.enrichCreateItem("037c00c4-b8ae-4baa-
b30f-64395fd5bdba");
              itemValueUpdates.set(FindeditemUid, @[История
```

```
наименований].toId());
            }
            //Получим построитель запроса на изменение значений позиции с адресом
            itemAggValueUpdates = support.enrichUpdateItem(FindeditemUid);
            //new SimpleDateFormat("yyyy-MM-dd").parse(headDateValue)
            if ((nmDateValue!=null)&&(nmDateValue!=""))
              itemAggValueUpdates.set(nmDateValue, @[Дата внесения изменений ЕГРЮЛ].toId());
            else
              itemAggValueUpdates.clear(@[Дата внесения изменений ЕГРЮЛ].toId());
            if ((nmFullValue!=null)&&(nmFullValue!=""))
              itemAggValueUpdates.set(nmFullValue, @[Полное наименование контрагента].toId());
            else
              itemAggValueUpdates.clear(@[Полное наименование контрагента].toId());
            if ((nmReadableValue!=null)&&(nmReadableValue!=""))
              itemAggValueUpdates.set(nmReadableValue, @[Сокращённое наименование контрагента].toId());
            else
              itemAggValueUpdates.clear(@[Сокращённое наименование контрагента].toId());
            if ((nmShortValue!=null)&&(nmShortValue!=""))
              itemAggValueUpdates.set(nmShortValue, @[Наименование контрагента].toId());
            else
              itemAggValueUpdates.clear(@[Наименование контрагента].toId());
            //Выполним изменение позиции с управляющей компанией
```

```
            itemAggValueUpdates.invoke();
          }
        }
        //Удалим оставшиеся позиции, отсутствующие в списке
        for(String itemUid in aggrItems) {
          support.enrichDeleteItem(itemUid);
        }
        aggrItems.clear();

        //Выполним изменение основной позиции
        itemValueUpdates.invoke();

        return null;
    }
```

3.33.3 Использование скриптов Groovy при выполнении импорта

Для расширения штатных возможностей импорта в подсистеме импорта и репликации предусмотрена возможность разработки скриптов Groovy. Подробную информацию о возможностях данного языка, смотрите здесь:

- https://docs.groovy-lang.org/docs/latest/html/documentation/
- https://www.tutorialspoint.com/groovy/index.htm

Скрипты Groovy расширяют возможности импорта данных из внешних систем и позволяют решать следующие задачи:

- Выполнять преобразование структуры сообщения импорта;
- Выполнять обогащение сообщения импорта данными из внешних сервисов;
- Выполнять проверку данных и отклонять сообщений импорта не прошедших проверку;
- Выполнять преобразование значений свойств в сообщении импорта (маппинг значений);

- Выполнять сопоставление с позициями справочников;
- Указывать группы, в которых требуется создавать новые позиции основного или вспомогательных справочников.
- В подсистеме импорта, разработка скриптов Groovy предусмотрена:
- для атрибута обменного класса – задается через кнопку “Формула импорта” для атрибута. Позволяет реализовать преобразование значения (маппинг значений), поступившего из внешней системы. Для атрибутов обменного класса, сопоставленных с агрегацией, позволяет реализовать программное сопоставление с позицией Semantic MDM по присланным вложенным атрибутам. Например, по вложенному атрибуту “Обозначение документа” (например, “ГОСТ 1050-88”), выполнить поиск позиции в справочнике документов и записать в импортируемый объект идентификатор найденной позиции;
- для обменного класса в целом – задается в свойстве обменного класса «Формула импорта». Позволяет выполнить обогащение импортируемого объекта данными из внешней системы, выполнить преобразование структуры объекта, добавить/удалить атрибуты и т.д.

Последовательность выполнения скриптов следующая: сначала будут выполнены все скрипты назначенные на атрибуты обменного класса (в случайном порядке), затем будет выполнен основной скрипт, назначенный на обменный класс.

В любом из скриптов, можно выполнить отклонение сообщения импорта с указанием причины отклонения, если одно или несколько значений атрибутов не проходит программные проверки. Для отклонения сообщения импорта, достаточно в функции скрипта Groovy (в формуле импорта) вернуть через **return** причину отклонения. Например, если ИНН присланного контрагента не проходит проверку на контрольное число, то в функции импорта должна вернуть причину отклонения -

return ‘ИНН присланного контрагента не прошел проверку на контрольное число’,

в противном случае, если не требуется выполнять отклонение сообщения, то функция должна вернуть **NULL**:

```
return null
```

3.33.3.1 Структура сообщения импорта

Структура сообщения импорта, определяется настройкой атрибутов в обменном классе. Каждый атрибут обменного класса (**Property**) имеет обязательно наименование, соответствующее наименованию атрибута во внешней системе и сопоставление (Рисунок 514):

- с атрибутом Semantic MDM,
- с переходным ключом внешней системы,
- без сопоставления.

В последнем случае предполагается, что данный атрибут будет дополнительно обработан скриптами Groovy и он будет программно сопоставлен с атрибутом Semantic MDM или на его основе будут добавлены новые дополнительные атрибуты в импортируемый объект, сопоставленные с атрибутами Semantic MDM. Атрибуты обменного класса могут образовывать иерархию, если в сопоставлении использовались атрибуты Semantic MDM типа «агрегация» или «композиция». В такие атрибуты могут быть добавлены вложенные атрибуты обменного класса, сопоставленные с вложенными в агрегацию/композицию атрибутами Semantic MDM.

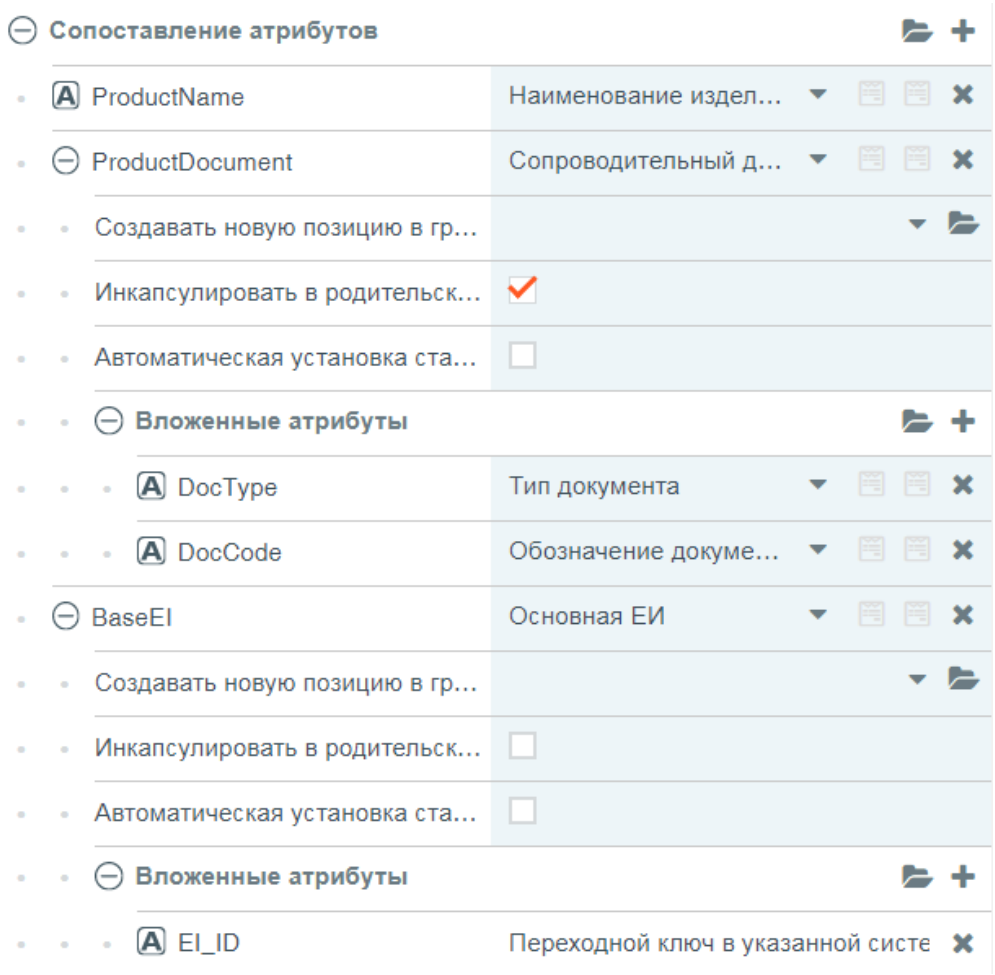


Рисунок 514 – Сопоставление атрибутов в обменном классе

Для сопоставленных с единичной агрегацией/композицией атрибутов обменного класса есть возможность включить опцию **Инкапсулировать в родительский класс**, чтобы внешняя система могла сразу передать значение для вложенного атрибута, через линейный список атрибутов. Но в независимости от значения этой опции, иерархия свойств в импортируемом объекте (**ProcessObject**), на момент выполнения Groovy-скрипта, будет восстановлена. Пример приведения структуры показан на Рисунок 515.

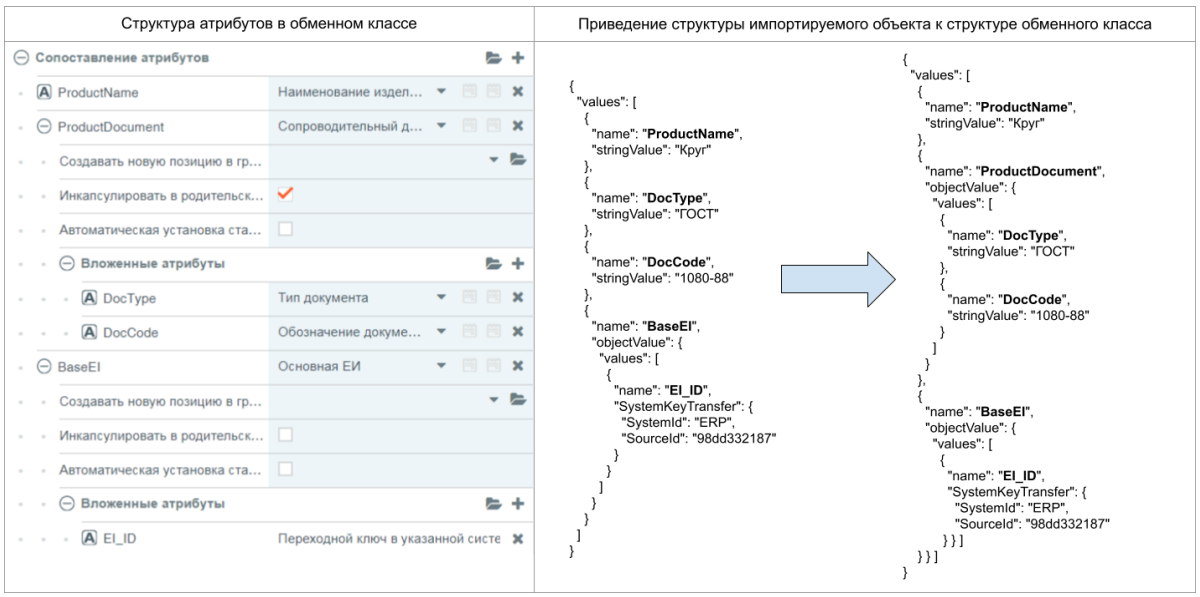


Рисунок 515 – Структура атрибутов в обменном классе

После проверки сообщения импорта на предмет целостности данных и отсутствия не декларированных в обменном классе атрибутов, подсистема импорта создаст на основе обменного класса импортируемый объект (класс **ProcessedRootObject**), заполнит его атрибуты значениями (класс **ProcessedValue**) из сообщения импорта, после чего выполнит скрипты Groovy, для атрибутов обменного класса (Формула импорта на атрибут) и затем скрипт Groovy для обменного класса (Формула импорта на обменный класс).

Ниже представлен пример соответствия структуры обменного класса структуре импортируемого объекта (**ProcessedRootObject**) с иерархией атрибутов (**ProcessedValue**) и вложенных через агрегацию объектов (**ProcessedObject**) (Рисунок 516). Класс **ProcessedRootObject** является наследником класса **ProcessedObject**.

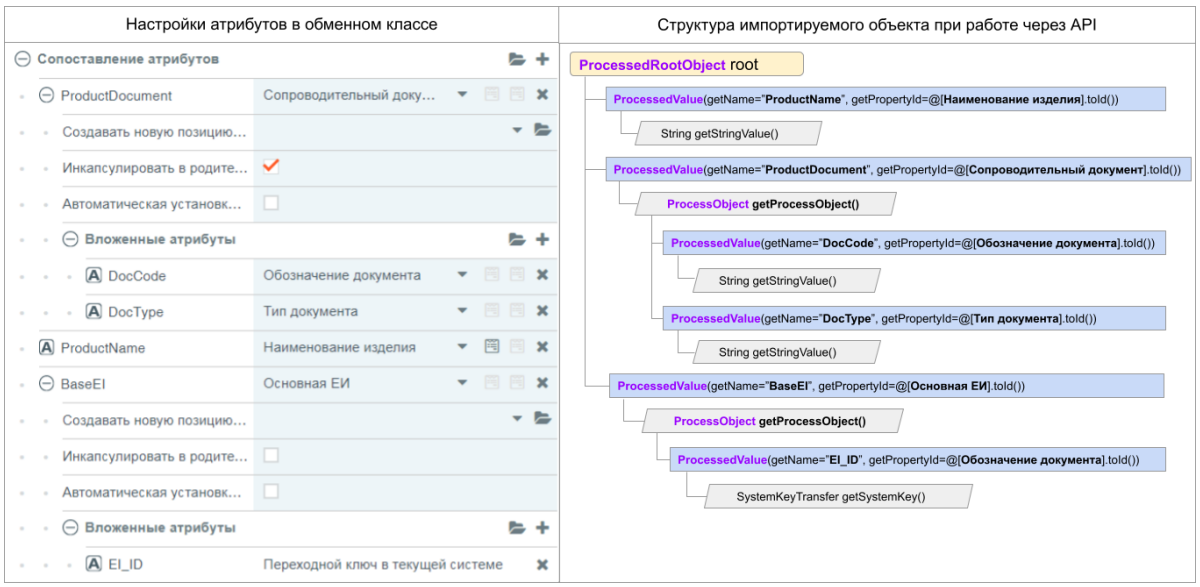


Рисунок 516 – Структура импортируемого объекта

3.33.3.2 Сопоставление импортируемого объекта с позицией Semantic MDM

Сопоставление импортируемого объекта и вложенных в него объектов может быть выполнено различными способами:

- **по GUID позиции Semantic MDM** – может быть передан в сообщении импорта, либо установлен через Groovy-скрипт с помощью метода **ProcessedObject.setMdmId**;
- **по идентификатору объекта во внешней системе** – при наличии с ним переходного ключа в Semantic MDM. Если в сообщении импорта передавался идентификатор объекта для третей внешней системы и для него в Semantic MDM имеется переходной ключ, то сопоставление также будет выполнено;
- **через поиск дубликатов Semantic MDM с включенной опцией «Использовать для сопоставления при импорте»** – для поиска используются поиски дубликатов, настроенные в группе, в которую будет импортирована позиция. Настройка группы для импорта позиций выполняется в свойстве «**Создавать новую позицию в группе**» обменного класса, а также в атрибуте обменного класса,

сопоставленного с агрегацией/композицией. Указать группу для импорта можно в Groovy-скрипт с помощью метода **ProcessedObject.setMdmNodeId**.

При выполнении импорта, сопоставление с позицией Semantic MDM выполняется сначала по GUID позиции Semantic MDM, если он отсутствует то по переходному ключу, в противном случае через настроенные поиски дубликатов. Если позицию не удалось сопоставить ни с одной позицией Semantic MDM, то будет создана новая позиция в группе, указанной в свойстве «**Создавать новую позицию в группе»**.

3.33.3.3 Методы для работы с импортируемым объектом

Методы ProcessedRootObject(ProcessObject)

- **boolean IsDeleted()** – получить состояние флага удаления объекта во внешней системе (true – объект удален, false – объект существует);
- **String getExchangeClassId()** – получить идентификатор обменного класса, для импортируемого объекта;
- **String getExternalSystemId()** – получить идентификатор внешней системы, для импортируемого объекта.

Методы ProcessObject

- **String getSourceId()** – получить идентификатор импортируемого объекта во внешней системе. Идентификатор объекта во внешней системе есть всегда и является обязательным для загрузки сообщения импорта.
- **String getMdmId()** – получить uuid позиции Semantic MDM, с которой требуется обеспечить наличие переходного ключа. Uuid позиции может быть передан внешней системой или записан через скрипт импорта Groovy. В большинстве случаев, внешняя система не присылает uuid позиции Semantic MDM и данное поле содержит **null**.
- **void setMdmId(String uuid)** – записать uuid позиции Semantic MDM, с которой требуется обеспечить наличие переходного ключа. Данный

метод востребован, если позиция Semantic MDM была подобрана с помощью скрипта Groovy и требуется записать с ней сопоставление.

- **getMdmNodeId()** – получить uuid группы, в которой будет создана позиция Semantic MDM, в случае отсутствия переходного ключа и отсутствия uuid позиции Semantic MDM в сообщении импорта (**getMdmId()**). Uuid группы настраивается в обменном классе, через свойство "**Создавать новую позицию в группе**".
- **setMdmNodeId()** – записать uuid группы, в которой будет создана позиция при отсутствии переходного ключа и отсутствия сопоставления с позицией Semantic MDM.
- **String getOriginalSourceId()** – получить идентификатор объекта во внешней системе, который стал оригиналом для текущего импортируемого объекта дубликата. При выполнении импорта, позиция Semantic MDM связанная через переходной ключ с текущей импортируемой позицией (**getSourceId**), будет признана дубликатом позиции Semantic MDM связанной идентификатором оригинала (**getOriginalSourceId**). Дедубликация будет выполнена только при условии, что позиция дубликат Semantic MDM имеет статус отличный от статуса «*Нормализована*», в противном случае будет только удален переходной ключ дубликата в соответствии с опцией «**Режим удаления переходного ключа**», выбранной в обменном классе.
- **ItemStatus getStatus()** – получить статус, который должен быть установлен для позиции Semantic MDM, после выполнения импорта. Если функция вернет **null**, то статус не будет изменен, после выполнения импорта. Данный статус может быть либо передан в сообщении импорта из внешней системы, либо установлен программно в скрипте Groovy с помощью функции **setStatus**. Возможные значения: **ItemStatus.Abnormal** (Не нормализована), **ItemStatus.Standardized** (Нормализована), **ItemStatus.Blocked** (Заблокирована), **ItemStatus.Removed** (Удалена);

- **setStatus(ItemStatus status)** – записать статус, который должен быть установлен для позиции Semantic MDM, после выполнения импорта.
- **String getAutoNormalize()** – получить флаг автоматической установки статуса **ItemStatus.Standardized** (Нормализована), после выполнения импорта. Если после выполнения импорта, позиция не будет соответствовать хотя бы одному обязательному ограничению, то статус позиции будет сброшен на **ItemStatus.Abnormal** (Не нормализована) и для сообщения импорта будет установлен статус «*Отклонена*».
- **setAutoNormalize(boolean autonorm)** – записать флаг автоматической установки статуса **Standardized** (Нормализована), после выполнения импорта.
- **List<ProcessedValue> getValues()** – получить список атрибутов со значениями, которые были переданы внешней системой. Если значение для какого-либо атрибута обменного класса не передавалось внешней системой, то полученный список не будет содержать данный атрибут. Допускается редактирование полученного списка (**List<ProcessedValue>**) через добавление нового или удаление существующего атрибута со значением.

Методы List<ProcessedValue>

Скриптовой язык Groovy, предоставляет возможность с помощью анонимных функций (Closure) выполнить поиск атрибута (**ProcessedValue**) в списке (**List<ProcessedValue>**) по любому его свойству с помощью метода **find()**. Так, например, можно найти атрибут в списке как по наименованию атрибута для внешней системы (**getName()**), так и по идентификатору атрибута Semantic MDM (**getPropertyId()**).

С помощью добавления/удаления атрибутов из списка, возвращаемого функцией **getValues()**, можно выполнять изменение состава атрибутов импортируемого объекта. Такая возможность необходима для того, чтобы привести состав атрибутов импортируемого объекта к составу атрибутов

группы (**ProcessedObject.getMdmNodeId**) Semantic MDM, в которую будет выполняться импорт.

Получение атрибута из списка List<ProcessedValue>:

Пример получения атрибута по наименованию для внешней системы:

```
ProcessedValue FindedVal = root.getValues().find {element -> element.getName()=="CA_TYPE"};
if (FindedVal != null)
{ /*... сделать что-то если атрибут нашли ...*/}
else
{ /*... сделать что-то если атрибут НЕ нашли ... */}
```

Пример получения атрибута по идентификатору атрибута Semantic MDM:

```
ProcessedValue FindedVal = root.getValues().find {element -> element.getPropertyId()==@[Тип контрагента].toId()};
if (FindedVal != null)
{ /*... сделать что-то если атрибут нашли ...*/ }
else
{ /*... сделать что-то если атрибут НЕ нашли ... */ }
```

Добавление атрибута в список List<ProcessedValue>:

Пример добавления строкового атрибута “Тип контрагента” со значением “ЮЛ”:

```
List<ProcessedValue> Values = root.getValues();
ProcessedValue NewValue = new ProcessedValue(propertyId: @[Тип контрагента].toId(), stringValue: "ЮЛ");
root.getValues().add(NewValue);
```

Пример добавления атрибута “Сопроводительный документ” со ссылкой на позицию с известным UUID:

```
List<ProcessedValue> Values = root.getValues();
```

```
ProcessedValue NewValue = new ProcessedValue(propertyId: @[Сопроводительный документ].toId(), processObject: new ProcessObject(mdmId: "6f667c55-8629-11ea-9c9c-f079596e5df8"));
root.getValues().add(NewValue);
```

Пример добавления атрибута “Сопроводительный документ” со ссылкой на позицию с известным идентификатором объекта в текущей внешней системе, из которой выполняется импорт:

```
ProcessedValue NewValue = new ProcessedValue(propertyId: @[Сопроводительный документ].toId(), processObject: new ProcessObject(sourceId: "6f667c55-8629-11ea-9c9c-f079596e5df8"));
root.getValues().add(NewValue);
```

Удаление атрибута из списка List<ProcessedValue>:

Пример удаления атрибута «Тип контрагента»:

```
//Найдем атрибут по идентификатору атрибута Semantic MDM
List<ProcessedValue> ListValues = root.getValues();
int FindedIdx = ListValues.findIndexOf {it.getName()=="CA_TYPE"};
if (FindedIdx >= 0) {
  ListValues.remove(FindedIdx);
}
```

Пример удаления всех атрибутов у импортируемого объекта:

```
//Найдем атрибут по идентификатору атрибута Semantic MDM
root.getValues().clear();
```

Методы ProcessedValue:

Чтение информации об атрибуте обменного класса из ProcessedValue

- **String getName()** – получить наименование атрибута для внешней системы;
- **Long getPropertyId()** – получить идентификатор атрибута Semantic MDM, с которым сопоставлен атрибут внешней системы;

- **ProcessObject getProcessObject()** – получить обрабатываемый объект (**ProcessObject**), которому принадлежит данное обрабатываемое значение атрибута (**ProcessValue**);
- **Date getChangeDate()** – получить дату и время изменения значения атрибута во внешней системе. Если внешняя система не прислала эту информацию, то будет записано время загрузки сообщения импорта в Semantic MDM, в случае, если это свойство изменилось относительно предыдущего сообщения импорта.

Чтение/запись единичных значений в ProcessedValue:

Предполагается, что при разработке скрипта разработчику известно к какому типу данных принадлежит значение, которое должно храниться в **ProcessValue**, и является оно множественным или единичным. Исходя из этой информации, разработчик должен использовать соответствующие методы для чтения и записи значений в атрибут.

- **String getStringValue()** – получить строковое значение атрибута для строкового типа данных. Если тип данных «Число с плав. Запятой», то будет получено строковое представление переданного числа, например: «10», «10 мм», «10 – 50», «10 - 50 мм», «12(-0.01;0.05) мм»;
- **setStringValue(String value)** – записать строковое значение в атрибут для строкового типа данных. Для типа данных: «Число с плав. Запятой» позволяет записать строковое представление числа с плав. запятой, например: «10», «10 мм», «10 – 50», «10 - 50 мм», «12(-0.01;0.05) мм»;. Если значение записано без указания ЕИ, то система будет считать что было передано значение в базовой единице измерения;
- **Date getDateValue()** - получить значение атрибута типа «Дата»;
- **setDateValue(Date value)** – записать дату (в формате UNIXTIME) в значение атрибута. Для приведения строковой даты к формату UNIXTIME, следует использовать тип **SimpleDateFromat**. Для этого требуется импортировать библиотеку: **import java.text.SimpleDateFormat**. Пример записи даты в **ProcessValue PVal**:

    - PVal.setDateValue(new SimpleDateFormat("yyyy-mm-dd").convert("2020-09-01"));
- **Boolean getBoolValue()** – получить булево значение атрибута;
- **setBoolValue(Bool value)** – записать булево значение в атрибут;
- **Integer getIntValue()** - получить числовое значение атрибута;
- **setIntValue(Integer value)** – записать числовое значение в атрибут;
- **ExchangeComplexDouble getComplexDoubleValue()** – используется системой Semantic MDM для внутреннего обмена числом с плавающей запятой. Для интеграции с внешними системами рекомендуется использовать: **getStringValue/setStringValue**;
- **setComplexDoubleValue(List<ExchangeComplexDouble> value)** – используется системой Semantic MDM для внутреннего обмена числом с плавающей запятой. Для интеграции с внешними системами рекомендуется использовать: **getStringValue/setStringValue**. Данный метод записывает новое значение числа с плав. запятой (**ExchangeComplexDouble**) в атрибут;
- **ProcessObject getProcessObject()** – получить вложенный объект из атрибута, который обладает всеми методами, описанными для класса **ProcessObject**;
- **setProcessObject(ProcessObject value)** – записать созданный объект (**ProcessObject**) в атрибут;
- **SystemKeyTransfer getSystemKey()** – получить значение переходного ключа, который прислали в атрибуте. Позволяет внешней системе передавать дополнительные идентификаторы на свои объекты во вспомогательных справочниках или сопоставление с объектами других систем. Получить идентификатор внешней системы и объекты можно с помощью методов:
    - **String getSystemId** – получить строковый идентификатор внешней системы, указанный в подсистеме импорта и репликации при регистрации внешней системы;

- **String getSourceId** – получить строковый идентификатор объекта внешней системы.

Чтение/запись множественных значений в ProcessedValue:

- **List<String> getStringValues()** – получить список строковых значений атрибута для строкового типа данных. Если тип данных «Число с плав. Запятой», то будет получен список строковых представлений переданных чисел.
- **setStringValues(List<String> values)** – записать список строковых значений в атрибут для строкового типа данных. Для типа данных: «Число с плав. Запятой» позволяет записать список строковых представлений чисел с плав. запятой.
- **List<Date> getDateValues()** - получить список значений атрибута типа "Дата".
- **setDateValues(List<Date> values)** – записать список значений типа дата в атрибут.
- **List<Integer> getIntValues()** - получить список числовых значений атрибута.
- **setIntValues(List<Integer> values)** – записать список числовых значений в атрибут.
- **List<ExchangeComplexDouble> getComplexDoubleValues()** – получить список значений чисел с плав. запятой из атрибута.
- **setComplexDoubleValues(List<ExchangeComplexDouble> values)** – записать список значений чисел с плав. запятой в в атрибут
- **List<ProcessObject> getProcessObjects()** – получить список вложенных объектов из атрибута.
- **setProcessObjects(ProcessObject values)** – записать список вложенных объектов в атрибут.

Методы ExchangeComplexDouble:

- **String getMeasurementUnit** – получить гуид единицы измерения Semantic MDM, в которой она должна отображаться в Semantic MDM;

- **ValueDoubleTransfer getValue** – получить объект класса **ValueDoubleTransfer** со следующими методами:
  - **double value** – содержит основное значение атрибута в базовой единице измерения, если из внешней системы передается простое число с плав. запятой или число с отклонениями. Отклонения для числа с плав. запятой будут храниться в min и max соответственно. Для диапазона, должно содержать центральное значение диапазона, к которому при прибавлении значения из min, получаем нижнюю границу отклонения, а при прибавлении значения из max, верхнее отклонение:
  - **double min** – получить значение нижнего отклонения;
  - **double max** – получить значение верхнего отклонения;
  - **encodedString** – строковое кодированное значение, содержит значения value, min и max в той ЕИ, в которой задал значение пользователь. Значение в **encodedString** кодируется следующим образом: **value**=<среднее число >; **min**=<нижнее отклонение, всегда меньше нуля>; **max**=<верхнее отклонение, всегда больше нуля>; **measure**=<числовой идентификатор ЕИ>; **precision**=<количество знаков после запятой для округления>; **type**=<тип ЕИ: «NORMAL», «RANGE», «DEFLECTIONS»>. Примеры кодированной строки **encodedString**:
    - **200 мм**: “value=200;min=0;max=0;measure=313;precision=5;type=NORMAL”
    - **300 – 400 мм**: “value=0;min=300;max=400;measure=313;precision=5;type=RANGE”
    - **10(-0.1;0.15) мм**: “value=10;min=-0.1;max=0.15;measure=313;precision=5;type=DEFLECTIONS”

3.33.3.4 Методы для чтения данных из Semantic MDM

Выполнение поиска позиций:

Возможны различные сценарии импорта, при которых необходимо выполнить поиск позиции по заданным атрибутам в справочнике *Semantic MDM*. Например, найти в классификаторе группу по коду или найти материал по марке и стандарту на материал и далее прописать сопоставление с найденной позицией в импортируемый объект.

Для реализации этих сценариев, в скрипт Groovy добавлены функции выполнения поиска позиции в заданной группе или в заданном классификаторе:

- **SearchParamsBuilder findInNode(String nodeUid)** – выполнить поиск в заданной группе (**nodeUid**) и вернуть первые 100 позиций;
- **SearchParamsBuilder findInNodes(List<String> nodeUids)** – выполнить поиск в заданном списке групп (**nodeUids**) и вернуть первый 100 позиций;
- **SearchParamsBuilder findInClassifier(String nodeUid)** – выполнить поиск в заданной классификационной группе (**nodeUid**) классификатора и вернуть первые 100 позиций. Данный метод востребован, при необходимости выполнить поиск позиции, среди позиций, классифицированных группой отдельного классификатора, например, *ОКВЭД2*, *ОКПД2*, *ЕКПС* и т.д..

Данные функции возвращают построитель критериев поиска (**SearchParamsBuilder**) в котором можно с помощью функций **byValue(String value, String propertyId)** и **byStatus(ItemStatus status)** добавить параметры поиска и далее выполнить поиск с помощью функции **Invoke()**, которая вернет список Uuid позиций (**List<String>**), найденных в *Semantic MDM*. Каждая из функций: **byValue** и **byStatus** возвращает построитель запросов **SearchParamsBuilder**, дополненный критерием по значению атрибута (**byValue**) или статуса (**byStatus**) соответственно, что позволяет путем последовательного вызова этих функций сформировать требуемый поисковый

запрос и выполнить его. Примеры вызова представлены ниже.

Пример поиска страны в классификаторе ОКСМ по наименованию и статусу «Нормализована» :

```
List<String> itemUids = support.findInNode("55805ab0-15cd-4523-9fc2-f926fcf3b376").byValue("РОССИЯ" , @[Наименование ОКСМ].toPath()).byStatus(ItemStatus.Standardized).invoke();
```

Пример поиска в справочнике материалов по марке материала, стандарту на материал и статусу «Нормализована»:

```
List<String> itemUids = support.findInNode("2003b902-7bee-49d5-9401-02fb79e1d6e9").byValue("Ст3кп", @[Марка материала].toPath()).byValue("ГОСТ 380-94", @[Стандарт на материал][Сокращенное наименование].toPath()).byStatus(ItemStatus.Standardized).invoke();
```

3.33.3.4.1 Чтение данных о позиции

В сценариях импорта, могут встречаться задачи получения текущего статуса позиции, получения идентификатора внешней системы или идентификатора *Semantic MDM* из переходного ключа, а также получения информации о группе. Пример сценария импорта: получить по идентификатору внешней системы идентификатор позиции *Semantic MDM*, проверить статус связанной позиции в *Semantic MDM* и если он соответствует статусу «*Заблокирована*», «*Удалена*», «*Дубликат*», то отменить импорт и вернуть ошибку, в противном случае, выполнить импорт по штатному алгоритму.

Для решения этих задач, в скрипте Groovy, в специальной переменной **support** предусмотрены следующие функции:

- **ItemStatus getItemStatus(String itemUid)** – получить статус позиции по GUID (**ItemUid**) позиции. Функция вернет одно из следующих значений:
  - **NULL** - если позиция с переданным **itemUid** не существует
  - **ItemStatus.Abnormal** – соответствует статусу «*Не нормализована*»;

- **ItemStatus.Standardized** – соответствует статусу «*Нормализована*»;
- **ItemStatus.Blocked** – соответствует статусу «*Заблокирована*»;
- **ItemStatus.Removed** – соответствует статусу «*Удалёна*»;
- **ItemStatus.Duplicate** – соответстует статусу «*Дубликат*».

- **String getMDMId(String sourceId, String systemId)** – получить GUID позиции по идентификатору объекта (**sourceId**) во внешней системе и идентификатору внешней системы (**systemId**). Вернет **NULL** если переходной ключ отсутствует.
- **String getSourceId(String itemUid, String systemId)** – получить идентификатор объекта во внешней системе по GUID позиции (**itemUid**) и идентификатору внешней системы (**systemId**). Если переходной ключ отсутствует, то вернет **NULL**.
- **String getItemNodeUid(String itemUid)** – получить GUID группы, которой принадлежит позиция с указанным GUID (**itemUid**). Если позиция с переданным GUID не существует, то будет возвращена ошибка.

Пример получения GUID позиции для импортируемого объекта:

```
String itemUid = support.getMDMId(root.getSourceId(), root.getExternalSystemId())
```

3.33.3.4.2 Чтение значений атрибутов позиции

Для чтения значений атрибутов позиции, необходимо использовать следующий метод:

- **List<PropertyValueDto> getItemPropertyValues(String itemUid)** – получить список значений атрибутов позиции с GUID **itemUid**. Если атрибут содержит множественные значения, то для каждого значения будет возвращен свой **PropertyValueDto**, с одинаковым **PropertyValueDto.propertyId**.

  Свойства **PropertyValueDto**:

  - **String itemUid** – идентификатор позиции;

- **long propertyId** – числовой идентификатор атрибута;
- **String stringValue** – содержит строковое значение атрибута, если тип данных: **String** или **Text**. Содержит GUID позиции, если тип данных **Aggregation** или **Composition**;
- **ExchangeComplexDouble doubleValue** – содержит число с плавающей запятой, если тип данных: **Double**;
- **long integerValue** – содержит целое число, если тип данных **Int**;
- **ExchangeFile fileValue** – содержит наименование файла и хэш содержимого. Если тип данных **Image**, то наименование файла будет пустым.

Свойства **ExchangeFile**:

- **name** – наименование файла, для изображения не указывается;
- **bodyHash** – хэш тела файла;

- **Boolean booleanValue** – содержит булево значение, если тип данных **Bool**;
- **Date dateValue** – содержит дату, если тип данных **DateTime**;
- **String nameValue** – содержит наименование контекста для контекстных значений атрибута;

- **List<PropertyValueDto> getItemsPropertyValueList(List<String> itemUids, List<long> PropertyIds)** – получить список значений для атрибутов указанных в **PropertyIds** и позиций указанных в **itemUids**. Если **PropertyIds** содержит пустой список, то будут возвращены значения всех атрибутов. Для каждого **PropertyValueDto** в свойстве **itemUid** будет указан guid позиции, к которому относится возвращенное значение атрибута. Если атрибут содержит множественные значения, то для каждого значения будет возвращен свой **PropertyValueDto**, с одинаковым **PropertyValueDto**.**propertyId** и одинаковым **PropertyValueDto.itemUid**.

Метод **getItemPropertyValues** возвращает список всех значений

атрибутов для заданной позиции (**itemUid**). Для чтения значений конкретного атрибута из полученного списка, необходимо знать идентификатор и тип данных атрибута, а также множественное у него значение или нет. Для получения этой информации в скрипте, необходимо использовать метод **getPropertyDeclarationsByNodeUid**.

Ниже представлены примеры чтения значений атрибутов.

Пример чтения значений атрибутов: «Наименование изделия», «Диаметр», «Материал->Марка материала»:

```
List<PropertyValueDto> itemValues = support.getItemPropertyValues("a1ddfe35-6fca-4f25-b874-604d082d9d3c");
//Получим значение атрибута "Наименование изделия"
PropertyValueDto attrValue = itemValues.find {element -> element. propertyId==@[Наименование изделия].toId()};
if (attrValue!=null)
{String prodName = attrValue.stringValue}
//Получим значение атрибута "Диаметр" в миллиметрах
attrValue = itemValues.find {element -> element. propertyId==@[Диаметр].toId()};
If (attrValue!=null)
//Получим диаметр в миллиметрах
{Double prodDiam = doubleValue.getValue().value/1000}
//Получим значение атрибута "Материал->Марка материала" в миллиметрах
attrValue = itemValues.find {element -> element. propertyId==@[Материал].toId()};
if (attrValue!=null)
{
prodMatUid = attrValue.stringValue;
List<PropertyValueDto> MaterialValues = support.getItemPropertyValues(prodMatUid);
```

attrValue = itemValues.find {element -> element. propertyId==@[Марка материала].toId()};

if (attrValue!=null)

{String prodMatMark = attrValue.stringValue}

}

Пример получения списка гуидов позиций из множественного атрибута «Логистические ЕИ»:

List<PropertyValueDto> itemValues = support.getItemPropertyValues("a1ddfe35-6fca-4f25-b874-604d082d9d3c");

List<PropertyValueDto> attrValues = itemValues.findAll {element -> element. propertyId==@[Логистические ЕИ].toId()};

List<String> itemGuids = attrValues.collect {element -> return element.stringValue };

3.33.3.4.3 Чтение данных по атрибутам из Semantic MDM

• **PropertyInfoDto getPropertyInfoByUid(String propertyUid)** – получить информацию об атрибуте по GUID атрибута *Semantic MDM*. Если атрибут с заданным **PropertyUid** не существует, то функция вернет **NULL**.

Свойства **PropertyInfoDto**:

- **long id** – числовой идентификатор атрибута;
- **String uuid** – GUID атрибута;
- **String eOTDid** – идентификатор OTD;
- **dataType** – тип данных атрибута:
  - **PropertyDataType.String** – строка;
  - **PropertyDataType.Double** – число с плав. запятой;
  - **PropertyDataType.Int** – целое число;
  - **PropertyDataType.Bool** – булев;
  - **PropertyDataType.Image** – картинка;
  - **PropertyDataType.Text** – текст;
  - **PropertyDataType.DateTime** – дата;
  - **PropertyDataType.Colour** – цвет;

- **PropertyDataType.File** – файл;
- **PropertyDataType.Formula** – формула;
- **PropertyDataType.Aggregation** – агрегация;
- **PropertyDataType.Composition** – композиция;

- **Map<String, String> names** – возвращается карта “Идентификатор языка” – “Наименование атрибута”. Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **Map<String, String> aggregateNames** – для атрибута связи возвращается карта “Идентификатор языка” – “Обратное наименование атрибута”. Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
  - **String comment** – описание атрибута;
  - **String sourceDocument** – значение свойства «Источник (документ)»;
  - **boolean isMultiple** – флаг **Массив**;
  - **boolean isHierarchical** – флаг **Иерархический**;
  - **boolean isContextual** – флаг контекстного атрибута;
- **List<PropertyInfoDto> getPropertyInfoByUidList(List<String> propertyUids)** – получить информацию по заданному списку атрибутов. Если один из атрибутов с заданным **PropertyUid** не существует, то функция не вернет информации по данному атрибуту.
- **PropertyInfoDto getPropertyInfoByOtdId(String propertyOtdId)** – получить информацию об атрибуте по идентификатору OTD атрибута Semantic MDM. Если атрибут с заданным **propertyOtdId** не существует, то функция вернет **NULL**.
- **List<PropertyInfoDto> getPropertyInfoByOtdIdList (List<String> propertyOtdIds)** – получить информацию по заданному списку атрибутов. Если один из атрибутов с заданным **PropertyOtdId** не существует, то

функция не вернет информации по данному атрибуту.

- **PropertyInfoDto getPropertyInfoById(Long propertyId)** – получить информацию об атрибуте по числовому идентификатору атрибута Semantic MDM. Если атрибут с заданным **PropertyId** не существует, то функция вернет ошибку.
- **List<PropertyInfoDto> getPropertyInfoByIdList (List< Long> propertyIds)** – получить информацию по заданному списку атрибутов. Если один из атрибутов с заданным **PropertyId** не существует, то функция вернет ошибку.
- **List<PropertyDeclarationInfoDto> getPropertyDeclarationsByNodeUid(String nodeUid, boolean resolveMultilinguals)** – получить список деклараций для группы с указанным GUID (**nodeUid**). Флаг **resolveMultilinguals** определяет необходимость получения данных о мультиязычных наименованиях в **PropertyInfoDto.Names**, **PropertyInfoDto.aggregateNames**, **PropertyDeclarationInfoDto.categories**.

  Свойства **PropertyDeclarationInfoDto:**

  - **List<String> scopeNodes** – список GUID групп, на позиции которых может ссылаться агрегация/композиция (**PropertyDeclarationInfoDto.dataType** равен **Aggregation** или **Composition**)
  - **Map<String, String> categorie**s – возвращается карта “Идентификатор языка” – “Наименование категории”. Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
    - **long id** – числовой идентификатор атрибута;
    - **String uuid** – GUID атрибута;
    - **String eOTDid** – идентификатор OTD;
    - **PropertyDataType dataType** – тип данных атрибута:
      - **PropertyDataType.String** – строка;
      - **PropertyDataType.Double** – число с плав. запятой;

- **PropertyDataType.Int** – целое число;
- **PropertyDataType.Bool** – булев;
- **PropertyDataType.Image** – картинка;
- **PropertyDataType.Text** – текст;
- **PropertyDataType.DateTime** – дата;
- **PropertyDataType.Colour** – цвет;
- **PropertyDataType.File** – файл;
- **PropertyDataType.Formula** – формула;
- **PropertyDataType.Aggregation** – агрегация;
- **PropertyDataType.Composition** – композиция;

- **Map<String, String> Names** – возвращается карта «Идентификатор языка» – «Наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **Map<String, String> aggregateNames** – для атрибута связи возвращается карта «Идентификатор языка» – «Обратное наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
  - **String comment**– описание атрибута;
  - **String sourceDocument** – массив значений «Источник (документ)»;
  - **boolean isMultiple** – флаг **Массив**;
  - **boolean isHierarchica**l – флаг **Иерархический**.

3.33.3.4.4 Чтение данных по группам из Semantic MDM

- **ClassifiableElementInfoDto getNodeInfo(String NodeUid, Boolean fetchParents)** – получить информацию о группе по ее GUID (**NodeUid**) Semantic MDM. Флаг **fetchParents** позволяет вернуть информацию по всем родительским группам. Если группа с заданным **NodeUid** не существует, то функция вернет **NULL**

Свойства **ClassifiableElementInfoDto**:

- **Long id** – числовой идентификатор группы;
- **String uid** – GUID группы;
- **String displayName** – наименование группы;
- **List<ClassifiableElementDto> parents** – список родительских групп, первой идет корневая группа, последней идет родительская группа для указанной группы (**NodeUid**).

3.33.3.5 Методы для работы с XML и JSON форматом

В сценариях импорта, могут встречаться задачи получения данных от внешней системы в формате XML или JSON в произвольной структуре, на основе которых требуется дополнить объект импорта необходимыми атрибутами и значениями, после чего выполнить импорт.

Получение данных из внешней системы в формате XML или JSON, можно организовать через атрибут обменного класса, из которого через Groovy-скрипт можно считать данные и далее добавить нужные атрибуты со значениями в импортируемый объект (**processObject**).

Для работы с XML и JSON форматами, в Groovy-скрипт добавлена поддержка следующих библиотек:

- **groovy.xml.XmlSlurper** – для работы с XML: https://docs.groovy-lang.org/latest/html/gapi/index.html?overview-summary.html
- **groovy.json.JsonSlurper** – для работы с JSON: https://docs.groovy-lang.org/latest/html/gapi/index.html?groovy/json/JsonSlurper

Для работы с ними в Groovy-скрипт, требуется в первой строке скрипта выполнить импорт (**import**) соответствующей библиотеки.

Пример чтения данных из XML:

```
import groovy.xml.XmlSlurper


String getSampleDSEFullName() {
StringBuilder XMLText = new StringBuilder();
```

```
//Сформируем пример XML с описанием объекта ДСЕ из внешней системы
XMLText.append('<ExternalObject>')
    .append('<FullName>Штуцер 123.01.01.40</FullName>')
    .append('<Attributes>')
    .append('<Attribute name="Наименование">Штуцер</Attribute>')
    .append('<Attribute name="Обозначение">123.01.01.40</Attribute>')
    .append('<Attribute name="Базовая ЕИ"><Object><Attributes><Attribute name="Обозначение ЕИ">шт.</Attribute></Attributes></Object></Attribute>')
    .append('</Attributes>')
    .append('</ExternalObject>');
//Выполним парсинг XML
def rootNode = new XmlSlurper().parseText(XMLText.toString());
//Извлечем значения атрибутов из присланного в XML объекта
String FullName = rootNode.'FullName'.text();
String DSECode = rootNode.'Attributes'.'*'.find {attribute -> attribute.'@name'=="Обозначение"}.text();
String DSEName = rootNode.'Attributes'.'*'.find {attribute -> attribute.'@name'=="Наименование"}.text();
String EICode = rootNode.'Attributes'.'*'.find {attribute -> attribute.'@name'=="Базовая ЕИ"}.'Object'.'Attributes'.'Attribute'[0].text();
//Вернем наименование по формуле: "<Наименование> <Обозначение> (<Обозначение ЕИ>)"
return DSEName + ' '+DSECode + ' ('+EICode+')';
}
```

Пример чтения данных из JSON:

```
import groovy.json.JsonSlurper
```

```
String getSampleDSEFullName () {
StringBuilder XMLText = new StringBuilder();
//Сформируем пример JSON с описанием объекта ДСЕ из внешней системы
XMLText.append('{')
    .append('   "FullName": "Штуцер 123.01.01.40",')
    .append('   "Attributes": [')
    .append('     {')
    .append('       "name": "Наименование",')
    .append('       "value": "Штуцер"')
    .append('     },')
    .append('     {')
    .append('       "name": "Обозначение",')
    .append('       "value": "123.01.01.40"')
    .append('     },')
    .append('     {')
    .append('       "name": "Базовая ЕИ",')
    .append('       "Object": {')
    .append('         "Attributes": [')
    .append('           {')
    .append('             "name": "Обозначение ЕИ",')
    .append('             "value": "шт."')
    .append('           }')
    .append('         ]')
    .append('       }')
    .append('   ]')
    .append('}');
//Выполним парсинг JSON
```

```
def rootNode = new JsonSlurper().parseText(XMLText.toString());
//Извлечем значения атрибутов из присланного в JSON объекта
String FullName = rootNode.'FullName';
String DSECode = rootNode.'Attributes'.find {it.'name'=="Обозначение"}.'value';
String DSEName = rootNode.'Attributes'.find {it.'name'=="Наименование"}.'value';
String EICode = rootNode.'Attributes'.find {it.'name'=="Базовая ЕИ"}.'Object'.'Attributes'[0].'value';
//Вернем наименование по формуле: "<Наименование> <Обозначение> (<Обозначение ЕИ>)"
return DSEName + ' '+DSECode + ' ('+EICode+')';
```

3.33.3.6 Методы для подключения к внешним сервисам

В сценариях импорта, могут встречаться задачи обогащения данными из внешних сервисов. Например, при получении контрагента, необходимо получить по ИНН информацию и ЕГРЮЛ и обогатить данными импортируемый объект.

Для подключения к внешним сервисам, в Groovy скрипт предусмотрена возможность использования **HttpURLConnection**. Ниже представлен пример подключения к сервису *Контур.Фокус*, получения информации о контрагенте по его ИНН в формате JSON и последующим парсингом JSON через JsonSlurper.

```
HttpURLConnection connection = new URL("https://focus-api.kontur.ru/api3/req?key=3208d29d15c507395db770d0e65f3711e40374df&ogrn=&inn=6663003127").openConnection() as HttpURLConnection;
def rootNode = new JsonSlurper().parseText(connection.inputStream.text);
```

3.33.3.7 Пример решения задачи сопоставления с позицией вспомогательного справочника

Данный пример демонстрирует возможность выполнения программного сопоставления с классификационной группой по коду классификатора ОКПД2, а также сопоставления по марке материала с материалом из вспомогательного справочника материалов.

Пример настройки обменного класса для импорта номенклатуры:

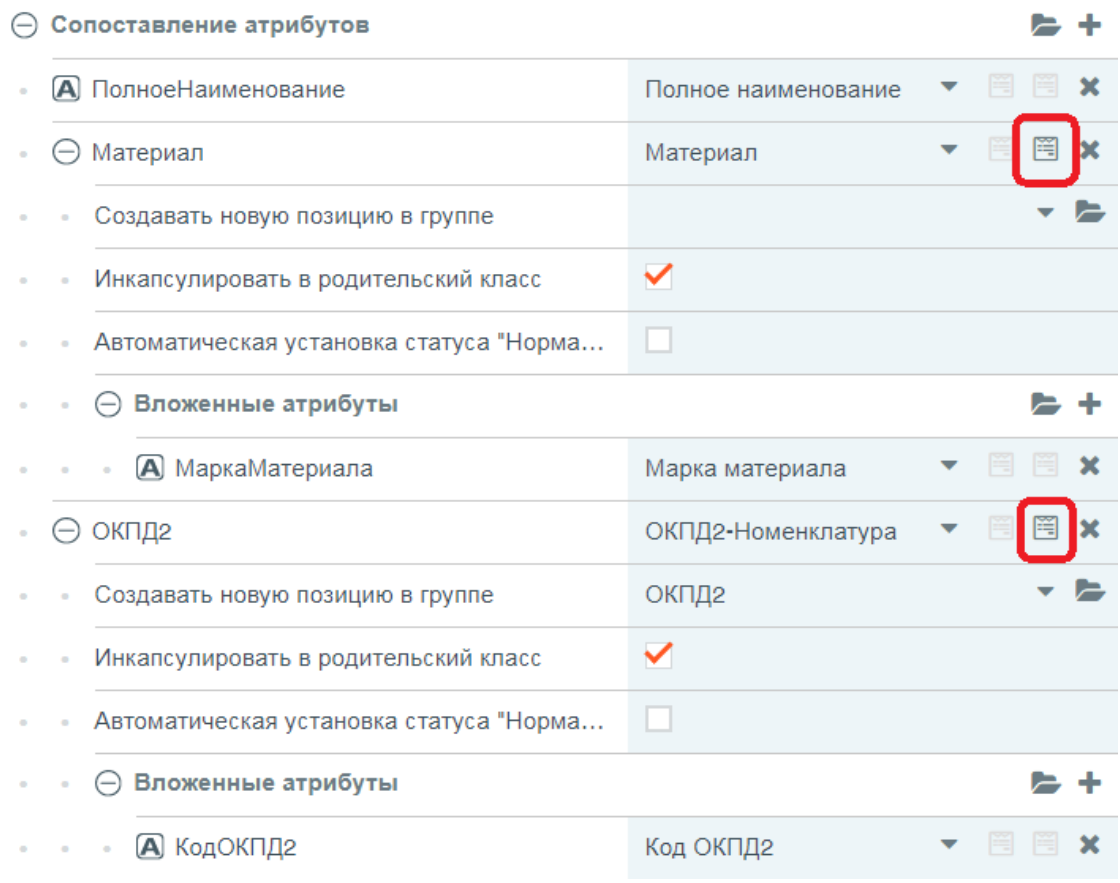


Рисунок 517 - Кнопка открытия Редактора формул

Для выполнения программного сопоставления, были разработаны два скрипта Groovy, один для атрибута «Материал», второй для атрибута «ОКПД2». Для открытия редактора написания скриптов необходимо нажать на кнопку открытия Редактора формул (Рисунок 517).

Формула импорта для атрибута «*Материал*»:

```
String transformValue(ProcessObject root, ProcessedValue value) {
```

```
//Получим значение вложенного атрибута
ProcessedValue attrVal = value.getProcessObject().getValues().find {element -> element.getName()=="МаркаМатериала"};
//Удалим все вложенные значения в атрибут "Материал", чтобы новый объект марки не был создан
value.getProcessObject().getValues().clear();

//Если получили значение с маркой материала
if (attrVal!=null)
{ //Если значение марки не пустое
 if (attrVal.getStringValue()!=null)
 {
  //Получим искомое значение марки материала и выполним поиск в справочнике
  String SearchVal = attrVal.getStringValue();
  List<String> itemUids = support.findInNode('2003b902-7bee-49d5-9401-02fb79e1d6e9')
                                .byValue(SearchVal, @[Марка материала].toPath())
                                .byStatus(ItemStatus.Standardized).invoke();
  //если найдена одна единственная марка, о запишем с ней сопоставление
  if (itemUids.size()==1)
  {
   value.getProcessObject().setMdmId(itemUids[0]);
  }
  else //в противном случае отклоним сообщение
  {
   if (itemUids.size() == 0)
    return 'В справочнике Материалов не найдено ни одной марки материала "'+SearchVal+'"';
```

```
            else
              return 'В справочнике Материалов найдено более одной марки материала '"+SearchVal+"';
            }
          }
          else
          {
            return 'Не указана марка материала'
          }
        }
        else
        {
          return 'Не указана марка материала'
        }

        return null;
      }
```

Формула импорта для атрибута «*ОКПД2*»:

```
      String transformValue(ProcessObject root, ProcessedValue value) {
        //Получим значение вложенного атрибута по наименования заданному для внешней системы
        ProcessedValue attrVal = value.getProcessObject().getValues().find {element -> element.getName()=="КодОКПД2"};
        //Удалим все вложенные значения в атрибут "ОКПД2", чтобы новый объект классификатора ОКПД2 не был создан
        value.getProcessObject().getValues().clear();

        //Если получили значение с кодом ОКПД2
        if (attrVal!=null)
        { //Если код ОКПД2 не пустой
```

```
            if (attrVal.getStringValue()!=null)
            {
             //Получим искомое значение кода ОКПД2 и выполним поиск в классификаторе ОКПД2
             String SearchVal = attrVal.getStringValue();
             List<String> itemUids = support.findInClassifier('5d7f1d1f-a94c-4ebc-938a-42accaab7658')
                                        .byValue(SearchVal, @[Код ОКПД2].toPath())
                                        .byStatus(ItemStatus.Standardized).invoke();
            //если была найдена одна единственная группа ОКПД2 в Semantic MDM по этому коду, то запишем с ней сопоставление
             if (itemUids.size()==1)
             {
              value.getProcessObject().setMdmId(itemUids[0]);
             }
             else //в противном случае отклоним сообщение, т.к. если была задана группа ОКПД2, то она должна быть в классификаторе
             {
              if (itemUids.size() == 0)
               return 'В классификаторе ОКПД2 не найдено ни одной группы с кодом "'+SearchVal+'";
              else
               return 'В классификаторе ОКПД2 найдено более одной группы с кодом "'+SearchVal+'";
             }
            }

           return null;
        }
```

3.33.3.8 Пример программного определения группы для создания позиции через импорт

Данный пример демонстрирует возможность программного определения системы, из которой поступило обрабатываемое сообщение импорта и указание группы, в которую требуется выполнить импорт данных из этой внешней системы. Данный сценарий востребован, когда необходимо разложить данные из внешних систем по разным группам, чтобы позиции в этих группах обрабатывали закрепленные за этими группами эксперты НСИ.

```
String transformRoot(ProcessedRootObject root) {

    //Получим идентификатор системы, для которой выполняется импорт
    String systemId = scriptData.getProcessedRootObject().getExternalSystemId();

    //В зависимости от идентификатора системы определим группу, в которой должны создаваться новые позиции
    switch (systemId) {
    case '1C_ERP': root.setMdmNodeId('b3f660f2-5d37-4cfa-83c2-90233cdaeed7'); break;
    case '1C_BUH': root.setMdmNodeId('23b2b08f-7c74-4728-928b-9a0758e9c106'); break;
    case 'PARUS': root.setMdmNodeId('3ce56be1-9a3a-4a7d-8b9c-dff11a32aa92'); break;
    }

    return null;

}
```

3.33.3.9 Пример обогащения данными импортируемого объекта из внешней системы

Данный пример наглядно демонстрирует решение следующих задач:

- преобразование структуры импортируемого объекта;
- обогащение импортируемого объекта данными из внешних сервисов;
- указание группы, в которой должны быть создана основная позиция;
- выполнять сопоставление позиции;
- отклонять сообщение импорта по заданным условиям с указанием причины отклонения.

Сценарий импорта: На вход поступает сообщение импорта из внешней системы, в котором передано полное наименование объекта справочника «*Номенклатура*». Требуется по полученному наименованию объекта, с помощью сервиса машинного обучения (ML) предсказать группу, в которой должна быть создана позиция и значения атрибутов, в том числе атрибутов вспомогательных справочников и классификаторов. После получения этих данных из сервиса машинного обучения (ML), требуется найти требуемые позиции во вспомогательных справочниках, после чего выполнить обогащение импортируемого объекта полученными данными из сервиса машинного обучения. Если импортируемый объект уже связан с позицией Semantic MDM через переходной ключ, то система должна проверить что позиция принадлежит той же группе, что была предсказана сервисом машинного обучения, если позиция не принадлежит предсказанной группе, то сообщение импорта должно быть отклонено. Сообщение импорта должно быть также отклонено, если не найдена нормализованная позиция во вспомогательном справочнике.

Для решения данной задачи, потребуется следующее:

```
import groovy.json.JsonSlurper

String transformRoot(ProcessedRootObject root) {
```

```
String EI_uid=null;
String DoubleVal=null;

//Получим полное наименование из импортируемого объекта
ProcessedValue FindedVal = root.getValues().find {element -> element.
getName()=="FULL_NAME"};
//Если не нашли атрибут с полным наименованием, то отклоняем
сообщение
if (FindedVal == null)
  return "Сообщение импорта не содержит наименование
объекта(FULL_NAME)";
String FullName = FindedVal.getStringValue();
//Удалим атрибут с полным наименованием, т.к. он будет
преобразован в набор атрибутов
root.getValues().remove(FindedVal);

// Получить из сервиса машинного обучения предсказанные значения
атрибутов, на основе полного наименование объекта (FULL_NAME)
HttpURLConnection connection = new URL("http://ml.sdi-
solution.ru/extract1?full_name=" +

ru.sdi.m2.resources.util.StringUtil.urlEncode(FullName)).openConnection() as
HttpURLConnection;

//Выполним парсинг JSON, с результатом предсказания
def rootNode = new
JsonSlurper().parseText(connection.inputStream.text);

//Получим GUID группы, в которой требуется создать позицию
//GUID группы предсказывается атрибутом attr_name=="groupId".
```

```
Идентификатор данного атрибута attr_id='31415926'
        String NodeUid = rootNode.extracted_attrs.find {it.attr_name=="groupId"}.attr_value;
        //Проверим существование группы
        ClassifiableElementInfoDto NodeInfo = support.getNodeInfo(NodeUid,false);
        if (NodeInfo==null)
          return "Предсказанная машинным обучением группы не существует";
        root.setMdmNodeId(NodeUid);

        //При наличии переходного ключа, проверить соответствует ли группа требуемой
        String mdmId = support.getMDMId(root.getSourceId(),root.getExternalSystemId());
        if (mdmId!=null)
        {//получить uid группы и сравнить ее с предсказанной
          String CurNodeUid = support.getItemNodeUid(mdmId);
          if (NodeUid!=CurNodeUid)
            return "Выявлено несоответствие текущей группы("+CurNodeUid+") позиции("+mdmId+") и предсказанной группы("+NodeUid+")";
        }

        //Получим список продекларированных атрибутов в предсказанной группе
        List<PropertyDeclarationInfoDto> nodeDeclarations = support.getPropertyDeclarationsByNodeUid(NodeUid,false);

        //Пройдемся в цикле по извлеченным атрибутам
```

```
            for (int i = 1; i < rootNode.extracted_attrs.size(); i++) {
              String PropPath = rootNode.extracted_attrs[i].attr_id;
              String PropValue = rootNode.extracted_attrs[i].attr_value;

              //Будем рассматривать только атрибуты с непустым значением
              //Атрибут "groupId" с идентификатором '31415926' игнорируем, т.к. обработали его ранее
              if ((PropPath!='31415926')&&((PropValue!=null)&&(PropValue!="")))
              {
               //Если атрибут связь
               int delimPos=PropPath.indexOf('->');
               if (delimPos>0)
               { //Получим гуид агрегации(атрибут связи) из пути
                 String AggrPropUid = PropPath.substring(0,delimPos);
                 PropertyDeclarationInfoDto attrDeclaration = nodeDeclarations.find {element -> element.uuid == AggrPropUid};
                 //Если агрегация с таким гуид не существует, то выдаем ошибку
                 if (attrDeclaration == null)
                  throw new Exception('не удалось найти декларацию атрибута связи с гуид "'+AggrPropUid+'" в группе "'+NodeUid+'"');
                 //Проверим, обрабатывалась ли ранее эта агрегация. Если обрабатывалась, то она уже будет в списке атрибутов импортируемого объекта
                 FindedVal = root.getValues().find {element -> element.getPropertyId()==attrDeclaration.id};
                 if (FindedVal==null)
                 { //Найдем все атрибуты, для обрабатываемой агрегации
                  def allAggrs = rootNode.extracted_attrs.findAll {it.attr_id.contains(AggrPropUid)}
                  //Получим область поиска агрегации
                  List<String> aggrScope = attrDeclaration.scopeNodes;
```

```
            if (attrDeclaration.scopeNodes.size()==0)
              return 'Область для атрибута "'+allAggrs[0].attr_name.subString(0,allAggrs[0].attr_name.indexOf('->'))+'" не задана в группе "'+root.getMdmNodeId()+"";
            String aggrItemUid = getAggregationItemUid(rootNode, AggrPropUid, aggrScope);
            //Добавим в импортируемый объект атрибут агригации со ссылкой на найденную агрегируемую позицию
            if (aggrItemUid!=null)
              root.getValues().add(new ProcessedValue(propertyId: attrDeclaration.id, processObject: new ProcessObject(mdmId: aggrItemUid)))
            else
              return 'Не удалось найти агрегируемую позицию "'+PropValue+'" для атрибута "'+allAggrs[0].attr_name.substring(0,allAggrs[0].attr_name.indexOf('->'))+"";
          }
        }
        //Если атрибут простой
        else
        {
          //Проверим наличие атрибута, если атрибут не декларирован, то выдаем ошибку
          PropertyDeclarationInfoDto attrDeclaration = nodeDeclarations.find {element -> element.uuid == PropPath};
          if (attrDeclaration==null)
            throw new Exception('не удалось найти атрибут с гуид "'+PropPath+"");

          //Добавим новый атрибут в импортируемый объект, если для него определено значение
```

```
            if ((PropValue!=null)&&(attrDeclaration!=null))
              switch (attrDeclaration.dataType) {
                case PropertyDataType.String:
                  //Добавим строковое значение атрибута в импортируемый объект
                  root.getValues().add(new ProcessedValue(propertyId: attrDeclaration.id, stringValue: PropValue));
                  break;
                case PropertyDataType.Int:
                  //Добавим числовое значение атрибута в импортируемый объект
                  root.getValues().add(new ProcessedValue(propertyId: attrDeclaration.id, intValue: PropValue.toInteger()));
                  break;
                case PropertyDataType.Double:
                  //Число с плав. запятой можно записать как строку без указания ЕИ, тогда будет считаться что передается значение в базовой ЕИ
                  //Число с плав. запятой можно записать с указанием обозначения ЕИ через пробел
                  root.getValues().add(new ProcessedValue(propertyId: attrDeclaration.id, stringValue: PropValue));
                  break;
              }
            }
          }
        }

        return null;

    }
```

```
//Найти агрегированную позицию по вложенным атрибутам и вернуть ее гуид
//rootNode - JSON с ответом подсистемы машинного обучения
//aggrPath - путь к агрегации из идентификаторов, для которой нужно найти агрегируемую позицию
//aggrAttrId - числовой идентификатор атрибута
String getAggregationItemUid(def rootNode, String aggrPath, List<String> aggrScope) {
 //Проверим что область агрегации где будем искать не пуста
 if (aggrScope.size()==0)
  throw new Exception('не определена область для агрегации "'+aggrPath+'"');

 //Получим список вложенных в агрегацию атрибутов
 def aggrAttrs = rootNode.extracted_attrs.findAll {it.attr_id.contains(aggrPath)};
 //Здесь будем хранить список обработанных вложенных агрегаций
 List<String> aggrProcessed = [];
 //Соберем поисковый запрос
 SearchParamsBuilder SearchBuilder=support.findInNodes(aggrScope);
 for (int i = 0; i < aggrAttrs.size(); i++) {
  //Получим полный путь к вложенному атрибуту
  String attrPath = aggrAttrs[i].attr_id;
  //Удалим из путя во вложенный атрибут, путь до агрегации
  String NestedAttr = attrPath.minus(aggrPath+'->');
  if ((aggrAttrs[i].attr_value != '')&&(aggrAttrs[i].attr_value != null)) {
   //Если во вложенном атрибуте есть еще вложенные атрибуты, то обработаем их
   if (NestedAttr.indexOf('->')>0)
   {
```

```
            String nestedAggrPropUid = Value.subString(0,NestedAttr.indexOf('->'));
            if (!aggrProcessed.contains(nestedAggrPropUid))
            {
              //Получим числовой идентификатор свойства
              Long nestedAggrPropId = support.getPropertyId(nestedAggrPropUid);
              //Получим область поиска агрегации
              List<String> nestedAggrScope = support.getPropertyScopeInNode(nestedAggrPropId,root.getMdmNodeId());
              //Получим агрегируемую позицию, найденную по вложенным атрибутам
              String AggregationItemUid=getAggregationItemUid(rootNode, aggrPath+'->'+nestedAggrPropUid, nestedAggrScope);
              //Добавляем критерий поиска
              if (AggregationItemUid!=null)
                SearchBuilder = SearchBuilder.byValue(AggregationItemUid, nestedAggrPropId);
              else
                throw new Exception('не удалось найти позицию для агрегации '"+aggrAttrs[i].nestedAggrPropUid+"');
            }
          }
        //Если вложенный атрибут является простым атрибутом и значение не пустое, то добавим его в критерии поиска
          else
          {
            PropertyInfoDto searchedProp = support.getPropertyInfoByUid(NestedAttr);
            String searchedVal = aggrAttrs[i].attr_value.toString();
            //Добавляем критерий поиска
```

```
            SearchBuilder = SearchBuilder.byValue(searchedVal,
searchedProp.id.toString());
          }
        }

        //Выполним поиск по сформированным критериям
        List<String> itemUids =
SearchBuilder.byStatus(ItemStatus.Standardized).invoke();

        //Если нашлась одна единственная позиция, то возвращаем ее
        if (itemUids.size()==1)
          return itemUids[0];
      }
      //Вернем пустоту, если ничего не нашли
      return null;
    }
```

3.33.4 Использование скриптов Groovy при выполнении экспорта

Для расширения штатных возможностей экспорта в подсистеме импорта и репликации предусмотрена возможность разработки скриптов Groovy. Подробную информацию о возможностях данного языка, смотрите здесь:

- https://docs.groovy-lang.org/docs/latest/html/documentation/
- https://www.tutorialspoint.com/groovy/index.htm

Скрипты Groovy расширяют возможности экспорта данных во внешние системы и позволяют решать следующие задачи:

- Выполнять преобразование структуры сообщения экспорта;
- Выполнять обогащение сообщения экспорта вычисляемыми атрибутами;
- Выполнять преобразование значений свойств в сообщении экспорта (маппинг значений).

В подсистеме экспорта, разработка скриптов Groovy предусмотрена:

- для атрибута обменного класса – задается через кнопку ***Формула экспорта*** для атрибута. Позволяет реализовать преобразование значения (маппинг значений), к терминам используемым во внешней системе.
- для обменного класса в целом – задается в свойстве обменного класса «**Формула экспорта**». Позволяет выполнить обогащение экспортируемого сообщения дополнительными вычисляемым данными, выполнить преобразование структуры сообщения, добавить/удалить атрибуты и т.д.

Последовательность выполнения скриптов следующая: сначала будут выполнены все скрипты назначенные на атрибуты обменного класса (в случайном порядке), затем будет выполнен основной скрипт, назначенный на обменный класс.

Стратегия использования скриптов Groovy в экспорте: скрипты Groovy должны использоваться в исключительных случаях, когда необходимо выполнить вычисление или маппинг значений. Если в скрипте Groovy для вычисления необходимы значения из единичных атрибутов, то данные требуется получать через ссылку на атрибут @[<имя атрибута>]. Ссылка на атрибут формируется автоматически, при перетаскивании атрибута из списка доступных атрибутов слева в окно с Groovy-скрипт. Если для вычисления необходимы значения из множественных атрибутов, то рекомендуется настроить в обменном классе выгрузку всех необходимых данных из множественных атрибутов. Далее, в скрипте Groovy необходимо считать эти подготовленные данные **ExportValue** из **ExportObject**, провести необходимые вычисления, после чего записать результат в нужный атрибут (**ExportValue**). Лишние атрибуты (**ExportValue**), которые нужны были для вычисления, после этого можно удалить.

3.33.4.1 Структура сообщения экспорта

Структура сообщения экспорта, определяется настройкой атрибутов в обменном классе. Каждый атрибут обменного класса (**Property**) имеет обязательно наименование, соответствующее наименованию атрибута во внешней системе и сопоставление:

- с атрибутом Semantic MDM,
- с переходным ключом внешней системы,
- без сопоставления.

В последнем случае предполагается, что данный атрибут будет вычислен через скрипт Groovy. Атрибуты обменного класса могут образовывать иерархию, если в сопоставлении использовались атрибуты Semantic MDM типа агрегация или композиция. В такие атрибуты могут быть добавлены вложенные атрибуты обменного класса, сопоставленные с вложенными в агрегацию/композицию атрибутами Semantic MDM.

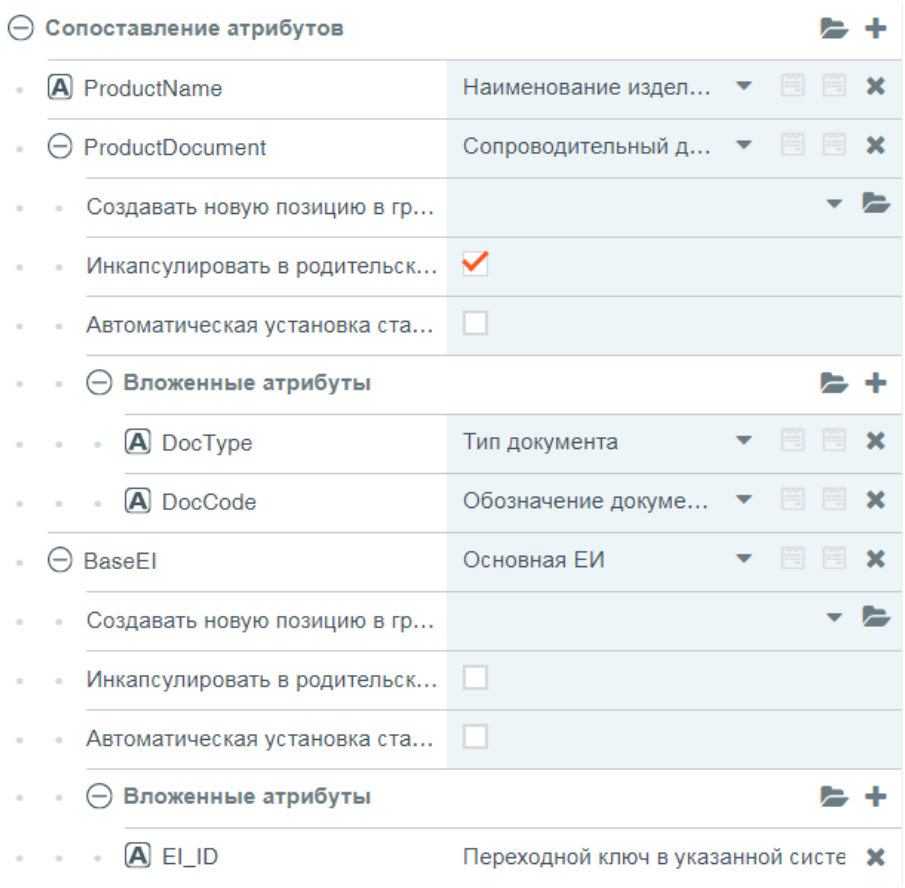


Рисунок 518 – Включение опции
«Инкапсулировать в родительский класс»

Для сопоставленных с единичной агрегацией/композицией атрибутов обменного класса есть возможность включить опцию «**Инкапсулировать в родительский класс**» (Рисунок 518), чтобы во внешнюю систему можно было сразу передать вложенный атрибут в линейном списке, без иерархии через агрегацию. Пример приведения структуры на рисунке ниже.

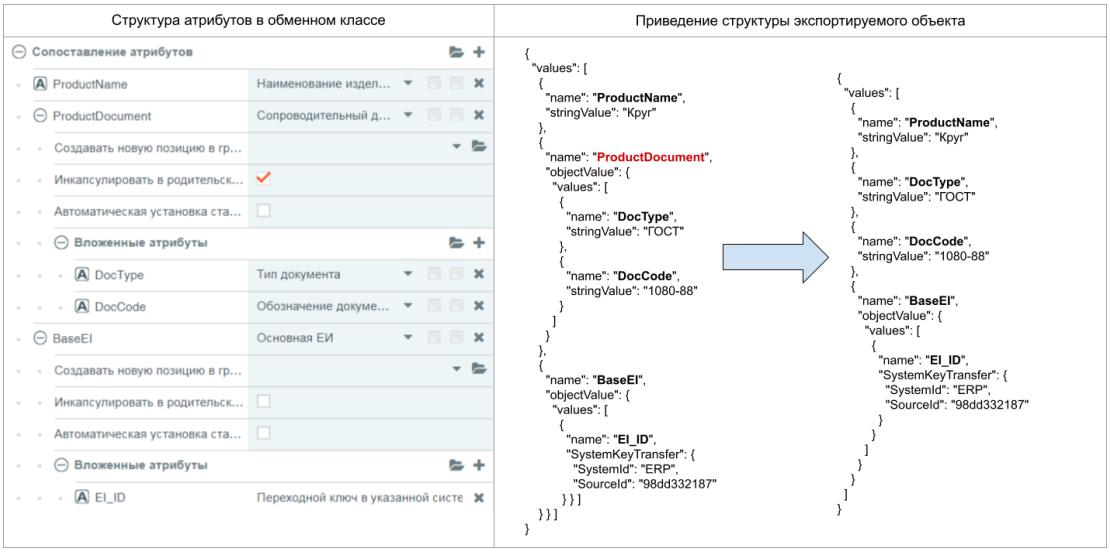


Рисунок 519 – Структура экспортируемого объекта

Перенос инкапсулируемых атрибутов на уровень выше, вложенных в агрегацию, выполняется после выполнения всех скриптов Groovy.

below представлен пример соответствия структуры обменного класса структуре экспортируемого объекта (**ExportRootObject**) с иерархией атрибутов (**ExportValue**) и вложенных через агрегацию объектов (**ExportObject**). Класс **ExportRootObject** является наследником класса **ExportObject**.

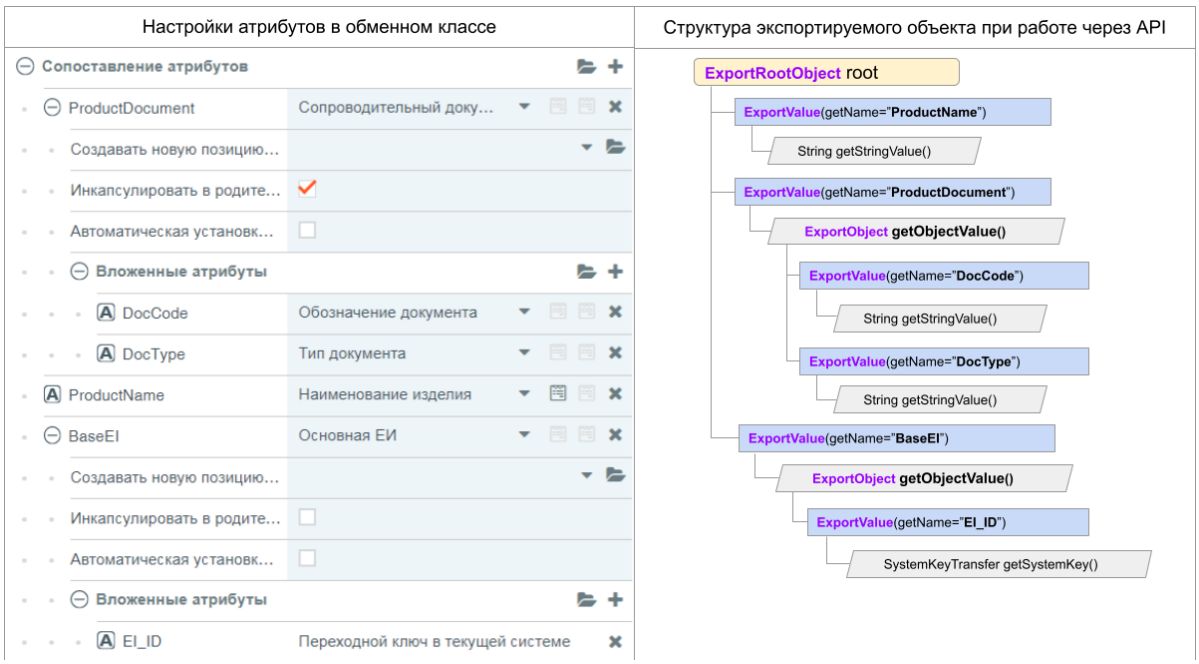


Рисунок 520 – Структура экспортируемого объекта при работе через API

Методы для работы с экспортируемым объектом:

Методы ExportRootObject (exportObject):

- **boolean getConformsExportRestriction ()** – удовлетворяет условиям экспорта (true – удовлетворяет / false – не удовлетворяет);
- **String getExchangeClassId()** – получить идентификатор обменного класса, для экспортируемого объекта;
- **String getExternalSystemId()** – получить идентификатор внешней системы, в которую экспортируется объект.

Методы ExportObject:

- **List<ExportSourceId> getSourceIds()** – получить идентификаторы внешней системы, с которыми связана экспортируемая позиция. Если для системы, в которую экспортируется позиция, в опции «Формировать сообщение на экспорт» установлен режим «для каждой позиции», то список будет содержать все идентификаторы объектов внешней системы, в которую выполняется экспорт. Если для системы в которую экспортируется позиция, в опции **«Формировать сообщение на экспорт»** установлен режим «для каждого переходного ключа», то сообщение экспорта будет сформировано для каждого

идентификатора внешней системы, в которую выполняется экспорт, при этом для каждого идентификатора дубликата (с флагом **isDuplicate()** =true), список **getSourceIds()** будет дополнен идентификатором объекта оригинала (с флагом **isDuplicate()** =false).

Свойства **ExportSourceId**:

- **String getValue()** – идентификатор объекта во внешней системе.
- **Boolean isDuplicate()** – признак дубликата (true – дубликат / false - оригинал).
- **setSourceIds(List<ExportSourceId>)** – записать в экспортируемый объект информацию о связанных с ним идентификаторах внешних объектов. Данный метод может быть востребован для инициализации нового **ExportObject**, созданного через оператор new в Groovy скрипте.
- **String getMdmId()** – получить uuid экспортируемой позиции Semantic MDM.
- **setMdmId()** – записать uuid позиции Semantic MDM. Данный метод может быть востребован только для инициализации нового **ExportObject**, созданного в Groovy скрипте через оператор new.
- **getMdmNodeId()** – получить uuid группы, которой принадлежит экспортируемая позиция.
- **ItemStatus getStatus()** – получить статус экспортируемой позиции. Возможные значения:
- **ItemStatus.Abnormal** (Не нормализована),
- **ItemStatus.Standardized** (Нормализована),
- **ItemStatus.Blocked** (Заблокирована),
- **ItemStatus.Removed** (Удалёна)
- **ItemStatus.Duplicate** (Дубликат)
- **List<ExportValue> getValues()** – получить список экспортируемых значений атрибутов. Допускается редактирование полученного списка (**List<ExportValue>**) через добавление нового или удаление

существующего атрибута со значением.

Методы List<ExportValue>:

Скриптовой язык Groovy, предоставляет возможность с помощью анонимных функций (**Closure**) выполнить поиск атрибута (**ExportValue**) в списке (**List<ExportValue>**) по любому его свойству с помощью метода **find()** или **findAll()**. Так, например, можно найти атрибут в списке по его наименованию для внешней системы (**getName()**) или по любому другому свойству, доступному в **ExportValue**. Подробнее о методах класса **ExportValue** в пункте п. 3.33.3.4.2.

С помощью добавления / удаления атрибутов из списка, возвращаемого функцией **getValues()**, можно выполнять изменение состава атрибутов экспортируемого объекта. Такая возможность необходима для того, чтобы привести состав атрибутов экспортируемого объекта к составу атрибутов справочника внешней системы.

Получение атрибута из списка List<ExportValue>:

Пример получения атрибута по наименованию для внешней системы:

```
ExportValue FindedVal = exportObject.getValues().find {element -> element.getName()=="CA_TYPE"};
if (FindedVal != null)
{ /*... сделать что-то если атрибут нашли ...*/}
else
{ /*... сделать что-то если атрибут НЕ нашли ... */}
```

Добавление атрибута в список List<ProcessedValue>:

Пример добавления строкового атрибута «CA_TYPE» со значением «ЮЛ»:

```
List<ExportValue> Values = exportObject.getValues();
ExportValue NewValue = new ExportValue(name: "CA_TYPE", stringValue: "ЮЛ");
```

```
exportObject.getValues().add(NewValue);
```

Пример добавления атрибута «DOCUMENT» с вложенным в него атрибутом «CODE»:

```
//Создадим вложенный объект с атрибутом CODE
ExportObject ValObj = new ExportObject();
ValObj.getValues().add(new ExportValue(name: "CODE", stringValue: "ГОСТ 1050-88"));
//Добавим новый атрибут DOCUMENT с созданным ранее объектом
exportObject.getValues().add(new ExportValue(name:"DOCUMENT", objectValue: ValObj));
```

Пример добавления атрибута «DOCUMENT» со ссылкой на позицию с известным идентификатором объекта во внешней системе:

```
//Создадим новый вложенный объект
ExportObject ValObj = new ExportObject();
//Пропишем в него ссылку на объект внешней системы
ValObj.setSourceIds([new ExportSourceId(value: "6f667c55-8629-11ea-9c9c-f079596e5df8", duplicate: false)]);
//Добавим новый атрибут DOCUMENT с созданным ранее объектом
exportObject.getValues().add(new ExportValue(name:"DOCUMENT", objectValue: ValObj));
```

Пример добавления атрибута «EXTERNAL_KEY» с переходным ключом:

```
SystemKeyTransfer key = new SystemKeyTransfer(systemId:"TEAMCENTER", sourceId: "gjt8rvZnjbRp1fcAZ")
exportObject.getValues().add(new ExportValue(name:"EXTERNAL_KEY", systemKey: key));
```

Удаление атрибута из списка List<ProcessedValue>:

Пример удаления атрибута «CA_TYPE»:

```
//Найдем атрибут по идентификатору атрибута Semantic MDM
List<ProcessedValue> ListValues = exportObject.getValues();
//Получим индекс атрибута
int FindedIdx = ListValues.findIndexOf {it.getName()=="CA_TYPE"};
if (FindedIdx >= 0) {
  ListValues.remove(FindedIdx);
}
```

Пример удаления всех атрибутов у экспортируемого объекта:

```
//Найдем атрибут по идентификатору атрибута Semantic MDM
exportObject.getValues().clear();
```

Методы ExportValue:

Чтение/запись имени атрибута обменного класса из ExportValue:

- **String getName()** – получить наименование атрибута для внешней системы
- **setName(String name)** –записать наименование атрибута для внешней системы. Востребовано только при создании **ExportValue** через оператор **new**.

Чтение/запись единичных значений в ExportValue:

- **String getStringValue()** – получить строковое значение атрибута для строкового типа данных;
- **setStringValue(String value)** – записать строковое значение в атрибут для строкового типа данных;
- **Date getDateValue()** - получить значение атрибута типа «Дата»;
- **setDateValue(Date value)** – записать дату в значение атрибута;
- **Boolean getBoolValue()**- получить булево значение атрибута;
- **setBoolValue(Bool value)** – записать булево значение в атрибут;
- **Integer getIntValue()** - получить числовое значение атрибута;
- **setIntValue(Integer value)** – записать числовое значение в атрибут;
- **ExchangeComplexDouble getComplexDoubleValue()** – получить

значение для атрибута с числом с плав. запятой;

- **setComplexDoubleValue(ExchangeComplexDouble value)** – записать значение числа с плав. запятой (**ExchangeComplexDouble**) в атрибут;
- **ExportObject getObjectValue()** – получить вложенный объект из атрибута, который обладает всеми методами, описанными для класса **ExportObject**;
- **setObjectValue(ExportObject value)** – записать созданный объект (**ExportObject**) в атрибут;
- **SystemKeyTransfer getSystemKey()** – получить переходной ключ с объектом внешней системы;

Свойства **SystemKeyTransfer**:

- **String getSystemId** – получить строковый идентификатор внешней системы, указанный при регистрации в подсистеме импорта и репликации;
- **String getSourceId** – получить строковый идентификатор объекта внешней системы;
- **setSystemKey(SystemKeyTransfer value)** – записать переходной ключ с объектом внешней системы;
- **ExchangeFile getFileValue()** – получить информацию о хеше и имени экспортируемого файла;

Свойства **ExchangeFile:**

- **name** – наименование файла, для изображения не указывается;
- **bodyHash** – хэш тела файла;
- **setFileValue(ExchangeFile value)** – записать информацию о хеше и имени экспортируемого файла. Получить хеш и имя файла из позиции Semantic MDM можно с помощью функции: **getItemPropertyValues**.

Чтение/запись множественных значений в ExportValue:

- **List<String> getStringValues()** – получить список строковых значений атрибута для строкового типа данных.
- **setStringValues(List<String> values)** – записать список строковых

значений в атрибут для строкового типа данных.

- **List<Date> getDateValues()** - получить список значений атрибута типа Дата.
- **setDateValues(List<Date> values)** – записать список значений типа дата в атрибут.
- **List<Integer> getIntValues()** - получить список числовых значений атрибута.
- **setIntValues(List<Integer> values)** – записать список числовых значений в атрибут.
- **List<ExchangeComplexDouble> getComplexDoubleValue()** – получить список значений чисел с плав. запятой из атрибута.
- **setComplexDoubleValues(List<ExchangeComplexDouble> values)** – записать список значений чисел с плав. запятой в в атрибут
- **List<ExportObject> getObjectValues()** – получить список вложенных объектов из атрибута.
- **setObjectValues(List<ExportObject> values)** – записать список вложенных объектов в атрибут.
- **List<ExchangeFile> getFileValues()** – получить информацию о хеше и имени экспортируемого файла
- **setFileValues(List<ExchangeFile> values)** – записать новый список файлов с указанием хеша и имени экспортируемого файла для каждого элемента списка. Получить хеш и имя файла из позиции *Semantic MDM* можно с помощью функции: **getItemPropertyValues**.

Методы ExchangeComplexDouble:

- **String getMeasurementUnit** – получить гуид единицы измерения Semantic MDM, в которой она должна отображаться в Semantic MDM;
- **ValueDoubleTransfer getValue** – получить объект класса **ValueDoubleTransfer** со следующими методами:
- **double value** – содержит основное значение атрибута в базовой единице измерения, если из внешней системы передается простое

число с плав. запятой или число с отклонениями. Отклонения для числа с плав. запятой будут храниться в min и max соответственно. Для диапазона, должно содержать центральное значение диапазона, к которому при прибавлении значения из min, получаем нижнюю границу отклонения, а при прибавлении значения из max, верхнее отклонение;

- **double min** – получить значение нижнего отклонения;
- **double max** – получить значение верхнего отклонения;
- **encodedString** – строковое кодированное значение, содержит значения **value**, **min** и **max** в той ЕИ, в которой задал значение пользователь. Значение в **encodedString** кодируется следующим образом: **value**=<среднее число >; **min**=<нижнее отклонение, всегда меньше нуля>; **max**=<верхнее отклонение, всегда больше нуля>; **measure**=<числовой идентификатор ЕИ>; **precision**=<количество знаков после запятой для округления>; **type**=<тип ЕИ: “NORMAL”, “RANGE”, “DEFLECTIONS”>. Примеры кодированной строки **encodedString**:

  • **200 мм**: “value=200;min=0;max=0;measure=313;precision=5;type=NORMAL”

  • **300 – 400 мм**: “value=0;min=300;max=400;measure=313;precision=5;type=RANGE”

  • **10(-0.1;0.15) мм**: “value=10;min=-0.1;max=0.15;measure=313;precision=5;type=DEFLECTIONS”

3.33.4.2 Методы для чтения данных из Semantic MDM

Основным способом чтения значений атрибутов текущей экспортируемой позиции, является чтение через конструкцию **@[имя атрибута]**. Для чтения значения вложенных в агрегации атрибутов,

необходимо последовательно перечислить имена атрибутов, например:

- **@[имя атрибута связи][имя вложенного атрибута]**
- **@[имя атрибута связи][имя вложенного атрибута связи][имя вложенного атрибута]**

Перечисленные конструкции вернут одно значение атрибута, если все перечисленные в конструкции атрибуты хранят единичное значение. Если хотя бы один из перечисленных в конструкции атрибутов хранит массив значений, то данная конструкция вернет **List** значений. Для каждого из типов данных атрибутов, будет возвращен свой тип данных:

- **строка**: String;
- **целое число**: Integer;
- **число с плав. запятой**: объект класса **ValueDouble**;
- класс **ValueDouble**:
  - **Double getValue()** – получить центральное значение;
  - **Double getMin()** – получить нижнее отклонение, всегда отрицательное ;
  - **Double getMax()** – получить верхнее отклонение, всегда положительное;
  - **String getEncodedString()** – строковое кодированное значение, содержит значения value, min и max в той ЕИ, в которой задал значение пользователь. Значение в **encodedString** кодируется следующим образом: **value**=<среднее число >; **min**=<нижнее отклонение, всегда меньше нуля>; **max**=<верхнее отклонение, всегда больше нуля>; **measure**=<числовой идентификатор ЕИ>; **precision**=<количество знаков после запятой для округления>; **type**=<тип ЕИ: “NORMAL”, “RANGE”, “DEFLECTIONS”>. Примеры кодированной строки **encodedString**:
    - **200 мм**: “value=200;min=0;max=0;measure=313;precision=5;type=NORMAL”;

- **300 – 400 мм**: “value=0;min=300;max=400;measure=313;precision=5;type=RANGE”;
- **10(-0.1;0.15) мм**: “value=10;min=-0.1;max=0.15;measure=313;precision=5;type=DEFLECTIONS”;

- **дата**: Date;
- **логический тип**: Boolean;
- **цвет**: integer, где первый, второй и третий байт кодируют цвет RGB;
- **файл**: имя файла;
- **атрибут связи**: String с guid позиции.

Методы чтения данных из *Semantic MDM*, описанные ниже, рекомендуется использовать в **исключительных случаях**, когда штатные средства настройки атрибутов обменных классов и использование ссылок на атрибуты (@[<наименование атрибута>]) в скриптах Groovy не позволяют прочитать необходимые данные из *Semantic MDM* для формирования сообщения экспорта.

3.33.4.2.1 Чтение данных о позиции

- **ItemStatus getItemStatus(String itemUid)** – получить статус позиции по GUID (**ItemUid**) позиции. Функция вернет одно из следующих значений:
  - **NULL** - если позиция с переданным **itemUid** не существует
  - **ItemStatus.Abnormal** – соответствует статусу «*Не нормализована*»,
  - **ItemStatus.Standardized** – соответствует статусу «*Нормализована*»,
  - **ItemStatus.Blocked** – соответствует статусу «*Заблокирована*»,
  - **ItemStatus.Removed** – соответствует статусу «*Удалёна*»,
  - I**temStatus.Duplicate** – соответствует статусу «*Дубликат*».
- **String getMDMId(String sourceId, String systemId)** – получить GUID позиции по идентификатору объекта (**sourceId**) во внешней системе и идентификатору внешней системы (**systemId**). Вернет **NULL** если

переходной ключ отсутствует.

- **String getSourceId(String itemUid, String systemId)** – получить идентификатор объекта во внешней системе по GUID позиции (**itemUid**) и идентификатору внешней системы (**systemId**). Если переходной ключ отсутствует, то вернет **NULL**.
- **String getItemNodeUid(String itemUid)** – получить GUID группы, которой принадлежит позиция с указанным GUID (**itemUid**). Если позиция с переданным GUID не существует, то будет возвращена ошибка.

3.33.4.2.2 Чтение значений атрибутов позиции

Основным способом чтения значений атрибутов текущей позиции, является чтение через конструкцию **@[имя атрибута]**. Для чтения значения вложенных в агрегации атрибутов, необходимо последовательно перечислить имена атрибутов, например:

- **@[имя атрибута связи][имя вложенного атрибута];**
- **@[имя атрибута связи][имя вложенного атрибута связи][имя вложенного атрибута];**
- **@[#composer][имя вложенного атрибута];**
- **@[#composer][имя вложенного атрибута связи][имя вложенного атрибута]**.

Перечисленные конструкции вернут одно значение атрибута, если все перечисленные в конструкции атрибуты хранят единичное значение. Если хотя бы один из перечисленных в конструкции атрибутов хранит массив значений, то данная конструкция вернет **List** значений.

Кроме получения списка значений (**List**) для вложенного атрибута, есть возможность получить полный путь до этого значения через метод: **getMapValues()**, который вернет **Map<String, Object>**, где **String** будет содержать GUID агрегированной позиции, а **Object** будет содержать либо значение заданного атрибута, либо еще один **Map<String, Object>**, если путь к вложенному атрибуту состоит из более чем одного атрибута связи.

Примеры возвращаемых значений:

- **@[имя множественного атрибута связи][имя вложенного строкового атрибута].getMapValues()** – данная конструкция вернет **Map<String, String>**;
- **@[имя множественного атрибута связи] [имя единичного атрибута связи] [имя вложенного строкового атрибута].getMapValues()** – данная конструкция вернет **Map<String, Map<String, String>>**;
- **@[имя множественного атрибута связи] [имя единичного атрибута связи] [имя вложенного множественного строкового атрибута].getMapValues()** – данная конструкция вернет **Map<String, Map<String, List<String>>>**.

Для каждого из типов данных атрибутов, будет возвращен свой тип данных:

- **строка**: String
- **целое число**: Integer
- **число с плав. запятой**: **valueDouble** - методы для работы с **ValueDouble**
- **дата**: Date
- **логический тип**: Boolean
- **цвет**: Long. – число в формате Blue, Green, Red, Alpha, где Blue – страший байт, а Alpha – младший байт. Пример красного цвета в шестнадцатиричном строковом представлении: “241CEDFF”, где: Blue=24, Green=1C, Red=ED, Alpha=FF.
    - Преобразование цвета из шестнадцатеричного строкового представления в число: Long.parseLong("241CEDFF",16);
    - Преобразование числа в строковое шестнадцатеричное представление: Long.toString(605875711,16);
- **файл**: String - имя файла
- **атрибут связи**: String - с guid позиции

Описанные ниже функции рекомендуется использовать в **исключительных случаях**, при невозможности изначально определить

полный перечень экспортируемых атрибутов в обменном классе. Примером такого исключения, может быть интеграция с внешними системами по справочнику номенклатуры, в котором необходимо передать информацию по списку атрибутов справочника, который не может быть определен заранее и должен вычисляться динамически, в зависимости от различных условий. Например, выполнить экспорт значений только для тех атрибутов, которые входят в определенную категорию атрибутов и/или участвуют в вычислении полного наименования.

- **List<PropertyValueDto> getItemPropertyValues(String itemUid)** – получить список значений атрибутов позиции с гуидом **itemUid**. Если атрибут содержит множественные значения, то для каждого значения будет возвращен свой **PropertyValueDto**, с одинаковым **PropertyValueDto.propertyId**.

  Свойства **PropertyValueDto**:

  - **String itemUid** – идентификатор позиции
  - **long propertyId** – числовой идентификатор атрибута
  - **String stringValue** – содержит строковое значение атрибута, если тип данных: **String** или **Text**. Содержит гуид позиции, если тип данных **Aggregation** или **Composition**
  - **ExchangeComplexDouble doubleValue** – содержит число с плавающей запятой, если тип данных: **Double**.
  - **long integerValue** – содержит целое число, если тип данных Int
  - **ExchangeFile fileValue** – содержит наименование файла и хэш содержимого. Если тип данных **Image**, то наименование файла будет пустым.

  Свойства **ExchangeFile**:

  - **name** – наименование файла, для изображения не указывается
  - **bodyHash** – хэш тела файла

  - **Boolean booleanValue** – содержит булево значение, если тип

данных **Bool**;

- **Date dateValue** – содержит дату, если тип данных **DateTime**;
- **String nameValue** – содержит наименование контекста для контекстных значений атрибута;

- L**ist<PropertyValueDto> getItemsPropertyValueList(List<String> itemUids, List<long> PropertyIds)** – получить список значений для атрибутов указанных в **PropertyIds** и позиций указанных в **itemUids**. Если **PropertyIds** содержит пустой список, то будут возвращены значения всех атрибутов. Для каждого **PropertyValueDto** в свойстве **itemUid** будет указан guid позиции, к которому относится возвращенное значение атрибута. Если атрибут содержит множественные значения, то для каждого значения будет возвращен свой **PropertyValueDto**, с одинаковым **PropertyValueDto.propertyId** и одинаковым **PropertyValueDto.itemUid**;
- L**ist<ScriptedPropertyValue> getCalculatedPropertyValueDependencies(String itemUid, Long propertyId)** – для указанной позиции (**itemUid**) получить атрибуты и их значения, на основе которых был вычислен указанный вычисляемый атрибут (**propertyId**). Данный метод вернет список **ScriptedPropertyValue**, который будет содержать информацию о значениях атрибутов и путях, к этим атрибутам.

  Свойства **ScriptedPropertyValue**:

  - **Long getPropertyId()** - идентификатор конечного атрибута;
  - **String getPropertyName()** - наименование конечного атрибута;
  - **List<Long> getCalculationPathIds()** – путь из идентификаторов атрибутов, по которому система получила значение для конечного атрибута;
  - **List<String > getCalculationPathUids()** – путь из GUID атрибутов, по которому система получила значение для конечного атрибута;
  - **List<String> getCalculationPathNames()** – путь из наименований

атрибутов, по которому система получила значение для конечного атрибута;

- **String getReadableValue()** – строковое значение свойства.

Метод **getItemPropertyValues** возвращает список всех значений атрибутов для заданной позиции (**itemUid**). Для чтения значений конкретного атрибута из полученного списка, необходимо знать идентификатор и тип данных атрибута, а также множественное у него значение или нет. Для получения этой информации в скрипте, необходимо использовать метод **getPropertyDeclarationsByNodeUid** (Чтение данных по атрибутам).

Ниже представлен пример чтения значений атрибутов.

Пример чтения значений атрибутов: «Наименование изделия», «Диаметр», «Материал->Марка материала»:

```
List<PropertyValueDto> itemValues = support.getItemPropertyValues("a1ddfe35-6fca-4f25-b874-604d082d9d3c");
//Получим значение атрибута "Наименование изделия"
PropertyValueDto attrValue = itemValues.find {element -> element.propertyId()==@[Наименование изделия].toId()};
if (attrValue!=null)
{String prodName = attrValue.stringValue}
//Получим значение атрибута "Диаметр" в миллиметрах
attrValue = itemValues.find {element -> element.propertyId()==@[Диаметр].toId()};
If (attrValue!=null)
//Получим диаметр в миллиметрах
{Double prodDiam = doubleValue.getValue().value/1000}
//Получим значение атрибута "Материал->Марка материала" в миллиметрах
attrValue = itemValues.find {element -> element.propertyId()==@[Материал].toId()};
if (attrValue!=null)
```

```
{
prodMatUid = attrValue.stringValue;
List<PropertyValueDto> MaterialValues = support.getItemPropertyValues(prodMatUid);
attrValue = itemValues.find {element -> element. propertyId()==@[Марка материала].toId()};
if (attrValue!=null)
{String prodMatMark = attrValue.stringValue}
}
```

3.33.4.2.3 Чтение данных по атрибутам из Semantic MDM

Данные методы рекомендуется использовать в исключительных случаях, когда кроме данных, во внешнюю систему необходимо передать информацию о метаданных. Использование данных методов может значительно снизить скорость репликации данных во внешние системы.

- **PropertyInfoDto getPropertyInfoByUid(String propertyUid)** – получить информацию об атрибуте по GUID атрибута Semantic MDM. Если атрибут с заданным **PropertyUid** не существует, то функция вернет **NULL**.

  Свойства **PropertyInfoDto**:

  - **long id** – числовой идентификатор атрибута;
  - **String uuid** – GUID атрибута;
  - **String eOTDid** – идентификатор OTD;
  - **dataType** – тип данных атрибута:
    - **PropertyDataType.String** – строка;
    - **PropertyDataType.Double** – число с плав. запятой;
    - **PropertyDataType.Int** – целое число;
    - **PropertyDataType.Bool** – булев;
    - **PropertyDataType.Image** – картинка;
    - **PropertyDataType.Text** – текст;

- **PropertyDataType.DateTime** – дата;
- **PropertyDataType.Colour** – цвет;
- **PropertyDataType.File** – файл;
- **PropertyDataType.Formula** – формула;
- **PropertyDataType.Aggregation** – агрегация;
- **PropertyDataType.Composition** – композиция;

- **Map<String, String> names** – возвращается карта «Идентификатор языка» – «Наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **Map<String, String> aggregateNames** – для атрибута связи возвращается карта «Идентификатор языка» – «Обратное наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **String comment**– описание атрибута;
- **String sourceDocument** – значение свойства «*Источник (документ)*»;
- **boolean isMultiple** – флаг **Массив**;
- **boolean isHierarchical** – флаг **Иерархический**;
- **boolean isContextual** – флаг контекстного атрибута;

- **List<PropertyInfoDto> getPropertyInfoByUidList(List<String> propertyUids)** – получить информацию по заданному списку атрибутов. Если один из атрибутов с заданным **PropertyUid** не существует, то функция не вернет информации по данному атрибуту;
- **PropertyInfoDto getPropertyInfoByOtdId(String propertyOtdId)** – получить информацию об атрибуте по идентификатору OTD атрибута Semantic MDM. Если атрибут с заданным **propertyOtdId** не существует, то функция вернет **NULL**;
- **List<PropertyInfoDto> getPropertyInfoByOtdIdList (List<String>**

propertyOtdIds) – получить информацию по заданному списку атрибутов. Если один из атрибутов с заданным **PropertyOtdId** не существует, то функция не вернет информации по данному атрибуту;

- **PropertyInfoDto getPropertyInfoById(Long propertyId**) – получить информацию об атрибуте по числовому идентификатору атрибута Semantic MDM. Если атрибут с заданным **PropertyId** не существует, то функция вернет ошибку;
- **List<PropertyInfoDto> getPropertyInfoByIdList (List< Long> propertyIds)** – получить информацию по заданному списку атрибутов. Если один из атрибутов с заданным **PropertyId** не существует, то функция вернет ошибку;
- **List<PropertyDeclarationInfoDto> getPropertyDeclarationsByNodeUid(String nodeUid, boolean resolveMultilinguals)** – получить список деклараций для группы с указанным GUID (**nodeUid**). Флаг **resolveMultilinguals** определяет необходимость получения данных о мультиязычных наименованиях в **PropertyInfoDto.Names**, **PropertyInfoDto.aggregateNames**, **PropertyDeclarationInfoDto.categories**;

  Свойства **PropertyDeclarationInfoDto**:

  - **List<String> scopeNodes** – список GUID групп, на позиции которых может ссылаться агрегация/композиция (**PropertyDeclarationInfoDto.dataType** равен **Aggregation** или **Composition**)
  - **Map<String, String> categories** – возвращается карта “Идентификатор языка” – “Наименование категории”. Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
  - **long id** – числовой идентификатор атрибута;
  - **String uuid** – GUID атрибута;
  - **String eOTDid** – идентификатор OTD;

- **PropertyDataType dataType** – тип данных атрибута:
  - **PropertyDataType.String** – строка;
  - **PropertyDataType.Double** – число с плав. запятой;
  - **PropertyDataType.Int** – целое число;
  - **PropertyDataType.Bool** – булев;
  - **PropertyDataType.Image** – картинка;
  - **PropertyDataType.Text** – текст;
  - **PropertyDataType.DateTime** – дата;
  - **PropertyDataType.Colour** – цвет;
  - **PropertyDataType.File** – файл;
  - **PropertyDataType.Formula** – формула;
  - **PropertyDataType.Aggregation** – агрегация;
  - **PropertyDataType.Composition** – композиция;
- **Map<String, String> Names** – возвращается карта «Идентификатор языка» – «Наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **Map<String, String> aggregateNames** – для атрибута связи возвращается карта «Идентификатор языка» – «Обратное наименование атрибута». Наименование на русском языке всегда должно быть первым. Идентификаторы языка: ru_RU – русский, en_US – английский;
- **String comment**– описание атрибута;
- **String sourceDocument** – массив значений «Источник (документ)»;
- **boolean isMultiple** – флаг Массив;
- **boolean isHierarchical** – флаг Иерархический.

3.33.4.2.4 Чтение данных по группам из Semantic MDM

- **getNodePath(String itemUid) -** получить путь из групп до указанной

позиции. Данная функция сформирует путь по следующему шаблону: <наименование группы>|<GUID группы><символ табуляции><наименование группы>|<GUID группы>… Пример пути до группы: Все справочники|00000000000000000000 -> Основные справочники|27cd5c64-64ef-4559-9695-ac4dc3b6104a -> Единый справочник Номенклатуры|967241ea-d6e6-4b36-a5bc-f041f4b94abc;

• **ClassifiableElementInfoDto getNodeInfo(String NodeUid, Boolean fetchParents)** – получить информацию о группе по ее GUID (**NodeUid**) Semantic MDM. Флаг **fetchParents** позволяет вернуть информацию по всем родительским группам. Если группа с заданным **NodeUid** не существует, то функция вернет **NULL**;

Свойства **ClassifiableElementInfoDto**:

• **Long id** – числовой идентификатор группы;

• **String uid** – GUID группы;

• **String displayName** – наименование группы;

• **List<ClassifiableElementDto> parents** – список родительских групп, первой идет корневая группа, последней идет родительская группа для указанной группы (**NodeUid**).

3.33.4.3 Методы для работы с XML и JSON форматом

В сценариях экспорта, могут встречаться задачи по экспорту данных в формате XML или JSON в произвольной структуре. Для решения данной задачи предполагается, что в обменный класс будет добавлен атрибут, через который будет выполняться передача XML/JSON во внешнюю систему. Формирование и запись XML/JSON текста в этот атрибут, должна осуществляться с помощью Groovy-скрипт, разработанного в «Формула экспорта» для обменного класса.

Для работы с XML и JSON форматами, в Groovy-скрипт добавлена поддержка следующих библиотек:

• groovy.xml.XmlSlurper – для работы с XML: https://docs.groovy-

lang.org/latest/html/gapi/index.html?overview-summary.html

• groovy.json.JsonSlurper – для работы с JSON: https://docs.groovy-lang.org/latest/html/gapi/index.html?groovy/json/JsonSlurper

Для работы с ними в Groovy-скрипт, требуется в первой строке скрипта выполнить импорт (**import**) соответствующей библиотеки.

Пример чтения данных из XML:

```
import groovy.xml.XmlSlurper

String getSampleDSEFullName() {
StringBuilder XMLText = new StringBuilder();
//Сформируем пример XML с описанием объекта ДСЕ из внешней системы
XMLText.append('<ExternalObject>')
    .append('<FullName>Штуцер 123.01.01.40</FullName>')
    .append('<Attributes>')
    .append('<Attribute name="Наименование">Штуцер</Attribute>')
    .append('<Attribute name="Обозначение">123.01.01.40</Attribute>')
    .append('<Attribute name="Базовая ЕИ"><Object><Attributes><Attribute name="Обозначение ЕИ">шт.</Attribute></Attributes></Object></Attribute>')
    .append('</Attributes>')
    .append('</ExternalObject>');
//Выполним парсинг XML
def rootNode = new XmlSlurper().parseText(XMLText.toString());
//Извлечем значения атрибутов из присланного в XML объекта
String FullName = rootNode.'FullName'.text();
String DSECode = rootNode.'Attributes'.'*'.find {attribute -> attribute.'@name'=="Обозначение"}.text();
String DSEName = rootNode.'Attributes'.'*'.find {attribute -> attribute.'@name'=="Наименование"}.text();
```

```
String EICode = rootNode.'Attributes'.'*'.find {attribute -> attribute.'@name'=="Базовая ЕИ"}.'Object'.'Attributes'.'Attribute'[0].text();
//Вернем наименование по формуле: "<Наименование> <Обозначение> (<Обозначение ЕИ>)"
return DSEName + ' '+DSECode + ' ('+EICode+')';
}
```

Пример чтения данных из JSON:

```
import groovy.json.JsonSlurper

String getSampleDSEFullName () {
StringBuilder XMLText = new StringBuilder();
//Сформируем пример JSON с описанием объекта ДСЕ из внешней системы
XMLText.append('{')
    .append('  "FullName": "Штуцер 123.01.01.40",')
    .append('  "Attributes": [')
    .append('    {')
    .append('      "name": "Наименование",')
    .append('      "value": "Штуцер"')
    .append('    },')
    .append('    {')
    .append('      "name": "Обозначение",')
    .append('      "value": "123.01.01.40"')
    .append('    },')
    .append('    {')
    .append('      "name": "Базовая ЕИ",')
    .append('      "Object": {')
    .append('        "Attributes": [')
    .append('          {')
```

```
            .append('                "name": "Обозначение ЕИ",')
            .append('                "value": "шт."')
            .append('              }')
            .append('            ]')
            .append('          }')
            .append('        }')
            .append('    ]')
            .append('}');
        //Выполним парсинг JSON
        def rootNode = new JsonSlurper().parseText(XMLText.toString());
        //Извлечем значения атрибутов из присланного в JSON объекта
        String FullName = rootNode.'FullName';
        String DSECode = rootNode.'Attributes'.find {it.'name'=="Обозначение"}.'value';
        String DSEName = rootNode.'Attributes'.find {it.'name'=="Наименование"}.'value';
        String EICode = rootNode.'Attributes'.find {it.'name'=="Базовая ЕИ"}.'Object'.'Attributes'[0].'value';
        //Вернем наименование по формуле: "<Наименование> <Обозначение> (<Обозначение ЕИ>)"
        return DSEName + ' '+DSECode + ' ('+EICode+')';
        }
```

3.33.4.4 Пример экспорта значений атрибутов, на основе которых было сформировано полное наименование позиции

Данный пример демонстрирует возможность программного определения списка атрибутов и их значений, на основе которых было сформировано полное наименование и далее добавление их в экспортируемый объект по следующим правилам:

- имя атрибута – должно быть сформировано по шаблону: attrId=”<guid

атрибута>”; attrName=<наименование атрибута>. Если использовались атрибуты агрегируемого объекта, то путь к этому атрибуту сформируем через “->”;

- значение атрибута – значение атрибута в строковом представлении.

Для решения данной задачи, “Формула экспорта” обменного класса должна содержать следующий скрипт:

```
void transformExportObject(ExportRootObject exportObject) {

//Добавим атрибуты на основе которых вычисляется наименование позиции
StringBuilder PathAttr = null;
List<ScriptedPropertyValue> AttrValues = support.getCalculatedPropertyValueDependencies(exportObject.getMdmId(),@[Полное наименование].toId());
for (int i = 0; i < AttrValues.size(); i++) {
//Получим список guid и наименований атрибутов, которые использовались для получения значения атрибута.
//Например, если значение атрибута было получено из Материал.Марка, то список будет содержать атрибуты Материал(0), Марка(1)
List<Long> AttrIds = AttrValues[i].getCalculationPathUids();
List<String> AttrNames = AttrValues[i].getCalculationPathNames();

//Сформируем путь из guid и наименований атрибутов
PathAttrIds = new StringBuilder();
PathAttrNames = new StringBuilder();
for (j = 0; j < AttrIds.size(); j++) {
 PathAttrIds.append(AttrIds.get(j).toString());
 PathAttrNames.append(AttrNames.get(j).toString());
  if (j<AttrIds.size()-1) {
   PathAttrIds.append('->');
```

```
                PathAttrNames.append('->');
            }
        }
        //Добавим к объекту экспорта новый атрибут
        exportObject.getValues().add(new ExportValue(name:
'attrId="'+PathAttrIds.toString()+'";attrName="'+PathAttrNames.toString(),
stringValue:AttrValues[i].readableValue ));
        }
    }
```

3.33.4.5 Пример экспорта содержания драг. металлов, рассчитанного на основе содержания драг. металла в марке

На данном примере, наглядно показано, как через groovy-скрипт прочитать необходимые данные по содержанию драг.металлов из марки материала, как прочитать необходимые коэффициенты пересчета из множественной композиции с логистическими ЕИ и как правильно провести вычисления с использование переменных типа **BigDecimal**.

Исходные данные по задаче: В обменном классе, настроен экспорт атрибутов номенклатуры, с информацией о содержании драг.металлов в граммах, полученных лабораторным путем. Ниже предоставлен перечень атрибутов с маппингом на атрибуты внешней системы:

- «Содержание серебра (Ag) лаб.» – «AG»
- «Содержание золота (Au) лаб.» – «AU»
- «Содержание иридия (Ir) лаб.» – «IR»
- «Содержание осмия (Os) лаб.» – «OS»
- «Содержание палладия (Pd) лаб.» – «PD»
- «Содержание платино-иридиевого сплава (ПлИ) лаб.» – «PL_IR»
- «Содержание платины (Pt) лаб.» – «PT»
- «Содержание родия (Rh) лаб.» – «RH»
- «Содержание рутения (Ru) лаб.» – «RU»

Задача: Для тех экспортируемых атрибутов, для которых отсутствует информация о содержании драг.металла, полученного лабораторным путем, требуется получить содержание доли драг.металла в марке материала и рассчитать содержание драг.металла в граммах. Для корректного расчета драг.металла в граммах, необходимо использовать базовую единицу измерения, определенную в атрибуте «Базовая ЕИ». Если базовая ЕИ не относится к мере «*Масса*», то для приведения базовой ЕИ к массе, необходимо использовать коэффициент пересчета из логистической ЕИ. Все доступные логистические ЕИ хранятся в атрибуте с множественной композицией «*Логистические ЕИ*», в которых указана логистическая ЕИ и коэффициент пересчета в базовую ЕИ. Кроме расчета драг.металлов, требуется в экспортируемый объект добавить атрибут «*DRAG_ME*» с флагом наличия драг.металлов в экспортируемом объекте.

Для решения данной задачи, потребуется в **Формулу экспорта** обменного класса записать следующий скрипт:

```
import java.math.RoundingMode

void transformExportObject(ExportRootObject exportObject) {

    ValueDouble valDragWeight;
    ValueDouble valDragPercent;
    Map<String, double> MapDragPercent = [:];
    List<String> DragAttrs = [];
    List<ExportValue> expValues = exportObject.getValues();

    //Добавим в экспорт атрибут с флагом наличия драг.металлов.
    //Значение данного атрибута, запишем по факту наличия одного из анализируемых драг.металлов
    valDragMe = expValues.find {element -> element.getName()=='DRAG_ME'};
```

```
if (valDragMe==null){
  valDragMe = new ExportValue(name: 'DRAG_ME');
  expValues.add(valDragMe);
}

//Получим содержание серебра в марке материала, если не указано содержание, полученное в лабораторных условиях
valDragWeight = @[Содержание серебра (Ag) лаб.];
if ((valDragWeight == null)||(valDragWeight.getValue()==0)) {
   valDragPercent = @[Материал][Содержание серебра (Ag)];
   if (valDragPercent!=null)
    if (valDragPercent.getValue()!=0) {
      MapDragPercent.put('AG',valDragPercent.getValue());
      DragAttrs.add('AG');
      valDragMe.setBoolValue(true);
    }
}
else if (valDragWeight.getValue()!=0)
  valDragMe.setBoolValue(true);

//Получим содержание золота в марке материала, если не указано содержание, полученное в лабораторных условиях
valDragWeight = @[Содержание золота (Au) лаб.];
if ((valDragWeight == null)||(valDragWeight.getValue()==0)) {
   valDragPercent = @[Материал][Содержание золота (Au)];
   if (valDragPercent!=null)
    if (valDragPercent.getValue()!=0) {
      MapDragPercent.put('AU',valDragPercent.getValue());
      DragAttrs.add('AU');
      valDragMe.setBoolValue(true);
```

```
        }
    }
    else if (valDragWeight.getValue()!=0)
      valDragMe.setBoolValue(true);

    //Получим содержание иридия в марке материала, если не указано содержание, полученное в лабораторных условиях
    valDragWeight = @[Содержание иридия (Ir) лаб.];
    if ((valDragWeight == null)||(valDragWeight.getValue()==0)) {
       valDragPercent = @[Материал][Содержание иридия (Ir)];
       if (valDragPercent!=null)
         if (valDragPercent.getValue()!=0) {
           MapDragPercent.put('IR',valDragPercent.getValue());
           DragAttrs.add('IR');
           valDragMe.setBoolValue(true);
         }
    }
    else if (valDragWeight.getValue()!=0)
      valDragMe.setBoolValue(true);

    //Получим содержание осмия в марке материала, если не указано содержание, полученное в лабораторных условиях
    valDragWeight = @[Содержание осмия (Os) лаб.];
    if ((valDragWeight == null)||(valDragWeight.getValue()==0)) {
       valDragPercent = @[Материал][Содержание осмия (Os)];
       if (valDragPercent!=null)
         if (valDragPercent.getValue()!=0) {
           MapDragPercent.put('OS',valDragPercent.getValue());
           DragAttrs.add('OS');
           valDragMe.setBoolValue(true);
```

```
        }
    }
    else if (valDragWeight.getValue()!=0)
     valDragMe.setBoolValue(true);

    //Получим содержание палладия в марке материала, если не указано содержание, полученное в лабораторных условиях
    valDragWeight = @[Содержание палладия (Pd) лаб.];
    if ((valDragWeight == null)||(valDragWeight.getValue()==0)) {
       valDragPercent = @[Материал][Содержание палладия (Pd)];
       if (valDragPercent!=null)
        if (valDragPercent.getValue()!=0) {
          MapDragPercent.put('PD',valDragPercent.getValue());
          DragAttrs.add('PD');
          valDragMe.setBoolValue(true);
        }
    }
    else if (valDragWeight.getValue()!=0)
     valDragMe.setBoolValue(true);

    //Получим содержание ПлИ в марке материала, если не указано содержание, полученное в лабораторных условиях
    valDragWeight = @[Содержание платино-иридиевого сплава (ПлИ) лаб.];
    if ((valDragWeight == null)||(valDragWeight.getValue()==0)) {
       valDragPercent = @[Материал][Содержание платино-иридиевого сплава (ПлИ)];
       if (valDragPercent!=null)
        if (valDragPercent.getValue()!=0) {
          MapDragPercent.put('PL_IR',valDragPercent.getValue());
```

```
            DragAttrs.add('PL_IR');
            valDragMe.setBoolValue(true);
          }
        }
        else if (valDragWeight.getValue()!=0)
          valDragMe.setBoolValue(true);

        //Получим содержание платины в марке материала, если не указано содержание, полученное в лабораторных условиях
        valDragWeight = @[Содержание платины (Pt) лаб.];
        if ((valDragWeight == null)||(valDragWeight.getValue()==0)) {
          valDragPercent = @[Материал][Содержание платины (Pt)];
          if (valDragPercent!=null)
            if (valDragPercent.getValue()!=0) {
              MapDragPercent.put('PT',valDragPercent.getValue());
              DragAttrs.add('PT');
              valDragMe.setBoolValue(true);
            }
        }
        else if (valDragWeight.getValue()!=0)
          valDragMe.setBoolValue(true);

        //Получим содержание родия в марке материала, если не указано содержание, полученное в лабораторных условиях
        valDragWeight = @[Содержание родия (Rh) лаб.];
        if ((valDragWeight == null)||(valDragWeight.getValue()==0)) {
          valDragPercent = @[Материал][Содержание родия (Rh)];
          if (valDragPercent!=null)
            if (valDragPercent.getValue()!=0) {
              MapDragPercent.put('RH',valDragPercent.getValue());
```

```
            DragAttrs.add('RH');
            valDragMe.setBoolValue(true);
          }
        }
        else if (valDragWeight.getValue()!=0)
          valDragMe.setBoolValue(true);

        //Получим содержание рутения в марке материала, если не указано содержание, полученное в лабораторных условиях
        valDragWeight = @[Содержание рутения (Ru) лаб.];
        if ((valDragWeight == null)||(valDragWeight.getValue()==0)) {
          valDragPercent = @[Материал][Содержание рутения (Ru)];
          if (valDragPercent!=null)
            if (valDragPercent.getValue()!=0) {
              MapDragPercent.put('AG',valDragPercent.getValue());
              DragAttrs.add('AG');
              valDragMe.setBoolValue(true);
            }
        }
        else if (valDragWeight.getValue()!=0)
          valDragMe.setBoolValue(true);

        //Если есть драг.металлы, для которых требуется рассчитать массу на основе доли указанной в марке,
        //то проверим базовую ЕИ и если она не относится к массе, то получим коэффициент пересчета в массу,
        //через логистические ЕИ
        BigDecimal RecalcKoef=1;
        String MassaEIGuid=null;
        if (DragAttrs.size()>0) {
```

```
String EI_class = @[Базовая ЕИ][Класс ЕИ][Наименование класса ЕИ];
MassaEIGuid=@[Базовая ЕИ];
if (EI_class!="Масса") {
 //Получим список GUID логистических ЕИ и их коэффициентов, композированных в номенклатуру, и среди них найдем ЕИ, которая относится к массе
 Map<String, Object> logEiGuids = @[Логистические ЕИ][Логистическая ЕИ].getMapValues();
 Map<String, Object> logKoefs = @[Логистические ЕИ][Коэффициент пересчета в базовую ЕИ].getMapValues();
 Set<String> LogisticsEI = logKoefs.keySet();
 for (String value: LogisticsEI) {
  //Получим guid логистической ЕИ
  String EILogGuid = logEiGuids.get(value);
  if ((EILogGuid!=null)&&(EILogGuid!=""))
  { //Нас интересуют только ЕИ Массы: гр, кг, т
   if ((EILogGuid=="5h6GkqQcBRlsBUJqCgvBka")||(EILogGuid=="KknB4s8zFcKeJRTo.6rlFb")||(EILogGuid=="YfCXjsngrBA.0FPpIT0Y7a"))
    {
     //Если коэффициент пересчета заполнен и статус соответствует "Нормализована" , то получим коэффициент
     ValueDouble RecalcEIKoef = logKoefs.get(value);
     if ((RecalcEIKoef!=null)&&(RecalcEIKoef.getValue()>0)&&(support.getItemStatusByUid(value)==ItemStatus.Standardized))
     {
      MassaEIGuid = EILogGuid;
      //Получим коэффициент пересчета из "кг" в базовую ЕИ. По
```

```
сути коэффициент это количество кг в одной базовой ЕИ
                    RecalcKoef = RecalcEIKoef.getValue();
                  }
                 }
                }
               }
              }
             }
            //Пересчитаем через коэф. пересчета содержание всех собранных драг.металлов
            BigDecimal DragPercent;
            ExportValue expDragVal;
            for (extAttrName in DragAttrs) {
              expDragVal = expValues.find {element -> element.getName()==extAttrName};
              if (expDragVal==null) {
                expDragVal = new ExportValue(name: extAttrNam, complexDoubleValue: new ExchangeComplexDouble(value: new ValueDoubleTransfer()));
                expValues.add(expDragVal);
              }
              else if (expDragVal.getComplexDoubleValue() == null) {
                expDragVal.setComplexDoubleValue(new ExchangeComplexDouble(value: new ValueDoubleTransfer()));
              }
              DragPercent = MapDragPercent.get(extAttrName);
              //Вернем долю содержания драг. металла в граммах
              if (MassaEIGuid=="5h6GkqQcBRlsBUJqCgvBka")//граммы
                expDragVal.getComplexDoubleValue().getValue().value = ((RecalcKoef*DragPercent/100G).setScale(7,RoundingMode.HALF_UP));
```

```
            else if (MassaEIGuid=="Knb4s8zFcKeJRTo.6rlFb")//килограммы
             expDragVal.getComplexDoubleValue().getValue().value =
((RecalcKoef*DragPercent/100G *
1000G).setScale(7,RoundingMode.HALF_UP));
            else if (MassaEIGuid=="YfCXjsngrBA.0FPpIT0Y7a")//тонны
             expDragVal.getComplexDoubleValue().getValue().value =
((RecalcKoef*DragPercent/100G *
1000000G).setScale(7,RoundingMode.HALF_UP));
            else
             throw new ru.sdi.m2.system.exception.M2Exception('ошибка
вычисления массы драг. металла. для атрибута:'"+extAttrName+"',
ЕИ:'"+MassaEIGuid+"', Коэф.пересчета:'"+RecalcKoef+"');
          }
        }
```

3.33.5 Использование скриптов Groovy в условиях репликации

Для расширения штатных возможностей условий репликации, в системе реализована возможность добавления в условия репликации скриптов Groovy. Основной задачей данного скрипта, является анализ необходимых данных по экспортируемой позиции и возврат булевого признака соответствия ее условиям экспорта.

3.33.5.1 Методы для чтения данных из Semantic MDM

3.33.5.1.1 Чтение данных о позиции

- **@[#itemGuid]** – получить guid текущей экспортируемой позиции. Данная сигнатура используется без **support**.
- **ItemStatus getItemStatus(String itemUid)** – получить статус позиции по GUID (**ItemUid**) позиции. Функция вернет одно из следующих значений:
  - **NULL** - если позиция с переданным **itemUid** не существует;

• **ItemStatus.Abnormal** – соответствует статусу «Не нормализована»;

• **ItemStatus.Standardized** – соответствует статусу «Нормализована»;

• **ItemStatus.Blocked** – соответствует статусу «Заблокирована»;

• **ItemStatus.Removed** – соответствует статусу «Удалёна»;

• **ItemStatus.Duplicate** – соответствует статусу «Дубликат»;

• **String getMDMId(String sourceId, String systemId)** – получить GUID позиции по идентификатору объекта (**sourceId**) во внешней системе и идентификатору внешней системы (**systemId**). Вернет **NULL** если переходной ключ отсутствует;

• **String getSourceId(String itemUid, String systemId)** – получить идентификатор объекта во внешней системе по GUID позиции (**itemUid**) и идентификатору внешней системы (**systemId**). Если переходной ключ отсутствует, то вернет **NULL**;

• **String getItemNodeUid(String itemUid)** – получить GUID группы, которой принадлежит позиция с указанным GUID (**itemUid**). Если позиция с переданным GUID не существует, то будет возвращена ошибка.

3.34 Специальные view в БД Semantic для разработки отчетов

В этом разделе дано описание реляционных таблиц БД Semantic MDM, применяемых для разработки шаблонов и формирования отчетов в подсистеме **Отчеты**.

3.34.1 rpt_application_template: Шаблоны заявок

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор шаблона заявки	
name	character varying(255)	Наименование шаблона заявки	

applicationtype	character varying(255)	Тип шаблона заявки	
usenodenegostages	boolean	Флаг получения маршрута согласования из групп, в которых были выполнены изменения	
dialogtype	integer	Вид отображения значений атрибутов позиции в заявке: 0 – Отображать только исходные значения 1 – Отображать только новые значения 2 – Отображать исходные и новые значения	
itemattrsusage	integer	Способ настройки атрибутов позиции: 0 – атрибуты позиции не задаются в заявке 1 – атрибуты позиции сформированы динамически, на основе ссылки на группу или позицию 2 – атрибуты позиции настроены статически в шаблоне заявки	
annulmentperiod	bigint	Срок аннулирования в днях	
pendingannulmentperiod	bigint	Срок уведомления об аннулировании в днях	
linkedexchangeclass	bigint	Идентификатор обменного класса, для которого используется данный шаблон заявки	rpt_exchange_class.id
startpageentryid	bigint	Идентификатор справочника или классификатора, заданный в шаблоне заявки	rpt_reference_book.id

3.34.2 rpt_app_property: атрибуты заявки

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор декларации атрибута в шаблоне заявки	
multiple	boolean	Флаг множественности значения атрибута	
attributetype	integer	Тип атрибута: 0 – Атрибут заявки с типом данных «Строка», значение в поле valuestr 1 – Атрибут заявки с типом данных «Текст», значение не доступно для чтения 2 - Атрибут заявки с типом данных «Файл», наименование файла в поле valuestr, тело файла не доступно для чтения 3 – Атрибут заявки с типом данных «Ссылка на группу или позицию». Если прописана ссылка, то значение хранится в	

		поле elementlinkid, если задали наименование вручную, то в elementlinkname 4 – Атрибут заявки с типом данных «Ссылка на атрибут», значение в поле propertyid, наименование атрибута в поле propertyname 5 – Атрибут заявки с типом данных «Ссылка на декларацию», в поле elementlinkid хранится идентификатор группы, в поле propertyid хранится идентификатор атрибута 7 – Атрибут заявки с типом данных «Ссылка на группу», значение хранится в поле elementlinkid, если задали наименование вручную, то в elementlinkname 8 – Атрибут заявки с типом данных «Ссылка на позицию», значение хранится в поле elementlinkid, если задали наименование вручную, то в elementlinkname	
ml_name	character varying(255)	Наименование атрибута в шаблоне заявки	

3.34.3 rpt_app_property_declaration: Декларации атрибутов в шаблоне заявки

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор декларации атрибута в шаблоне заявки	
application	bigint	Идентификатор шаблона заявки	
mlname	character varying(255)	Наименование атрибута в шаблоне заявки	
property	bigint	Идентификатор атрибута заявки, если атрибут был добавлен из классификатора атрибутов заявки	
attributetype	integer	Тип атрибута: 0 – Атрибут заявки с типом данных «Строка», значение в поле valuestr 1 – Атрибут заявки с типом данных «Текст», значение не доступно для чтения 2 - Атрибут заявки с типом данных «Файл», наименование файла в поле valuestr, тело файла не доступно для чтения	

		3 – Атрибут заявки с типом данных «Ссылка на группу или позицию». Если прописана ссылка, то значение хранится в поле elementlinkid, если задали наименование вручную, то в elementlinkname 4 – Атрибут заявки с типом данных «Ссылка на атрибут», значение в поле propertyid, наименование атрибута в поле propertyname 5 – Атрибут заявки с типом данных «Ссылка на декларацию», в поле elementlinkid хранится идентификатор группы, в поле propertyid хранится идентификатор атрибута 6 – Значение атрибута позиции. Ссылка на атрибут позиции хранится в propertyid, значение в зависимости от типа данных атрибута храниться в: ·Логический тип - valuebool ·Дата - valuedate ·Число с плав. запятой - valuedblstr ·Целое число - valueint ·Строка - valuestring ·Атрибут связи – elementlinkid ·Цвет – чтение не поддерживается ·Многострочный текст – чтение не поддерживается ·Изображение – чтение не поддерживается ·Файл – наименование файла в поле valuestr, тело файла не доступно для чтения 7 – Атрибут заявки с типом данных «Ссылка на группу», значение хранится в поле elementlinkid, если задали наименование вручную, то в elementlinkname 8 – Атрибут заявки с типом данных «Ссылка на позицию», значение хранится в поле elementlinkid, если задали наименование вручную, то в elementlinkname 9 – Системный атрибут заявки с типом данных «Статус позиции», значение хранится в поле itemstatus	
propertyid	bigint	Идентификатор атрибута позиции. Назначение данного поля зависит от значения в поле attributetype	rpt_m2_attribute.id rpt_m2_aggregationstereotype.id
propertyname	character varying(255)	Наименование атрибута позиции	
multiple	boolean	Флаг множественности значения	
inputmask	character varying(255)	Маска ввода	
mandatoriness	integer	Требования к вводу данных в атрибут: 0 – необязательный 1 – должен быть заполнен заявителем	

		2 – должен быть заполнен исполнителем	
defaultlink	boolean	Данный флаг устанавливается для атрибута заявки с типом данных «Ссылка на группу», «Ссылка на позицию», «Ссылка на группу/позицию», если в данный атрибут требуется: ·прописать ссылку на группу/позицию, стоя на которой создавали заявку, ·сформировать список динамических атрибутов позиции заявки, исходя из ссылки на позицию/группу прописанную в данный атрибут ·заполнить колонку «Исходные значения» или «Новые значения» при указании ссылки на позицию	
executorsource	boolean	Если данный флаг установлен для атрибута заявки, то значит исполнители заявки будут получены из группы, на которую прописана ссылка в поле elementlinkid. Если в elementlinkid прописана ссылка на позицию, то исполнители будут получены из группы, в которой находится данная позиция.	
periodssource	boolean	Если данный флаг установлен, то сроки обработки заявки будут получены из группы, ссылка на которую прописана в поле elementlinkid. Если в elementlinkid прописана ссылка на позицию, то исполнителей будут получены из группы, в которой находится данная позиция.	
visiblesince	integer	Видимость атрибута: 0 – виден при создании заявки 1 – должен быть виден, начиная с этапа исполнения	
elementlinkid	bigint	Идентификатор позиции или группы. Значение по умолчанию для атрибута с типом «Ссылка на группу», «Ссылка на позицию», «Ссылка на группу или позицию»	rpt_m2_item.id rtp_m2_classificationnode.id
elementlinkname	character varying(500)	Наименование позиции или группы. Заполняется после выбора позиции по умолчанию для атрибута с типом «Ссылка на группу», «Ссылка на позицию», «Ссылка на группу или позицию».	
valuebool	boolean	Значение по умолчанию для атрибута с логическим типом данных	
valuedate	timestamp	Значение по умолчанию для атрибута с типом данных: Дата и время	
valuedblstr	character varying(255)	Строковое представление значения по умолчанию для атрибута с типом данных: Число с плавающей точкой	
natureid	bigint	Идентификатор измеряемой сущности, для	rpt_m2_pr

		атрибута позиции с числом с плавающей запятой	opertygroup.id
desirableunitid	bigint	Идентификатор единицы измерения по умолчанию, которую задал пользователь для числа с плавающей запятой	rpt_m2_measurementunit.id
valueint	integer	Значение по умолчанию для атрибута с типом данных: Целое число	
valuestring	character varying(500)	Значение по умолчанию для атрибута с типом данных: Строковое значение	

3.34.4 rpt_application: журнал заявок

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор заявки	
duplicateOf	bigint	Данная заявка дублирует заявку с указанным id	rpt_application.id
uid	bigint	Номер заявки, без лидирующих нулей	
applicant	character varying(255)	Логин пользователя	rpt_m2_account.login
executor	character varying(255)	Логин назначенного исполнителя	rpt_m2_account.login
datecreated	timestamp	Дата и время создания заявки	
datechanged	timestamp	Дата и время последнего изменения заявки	
status	integer	Статус заявки: 0 – Зарегистрирована 1 – Ожидает обработки 2 – В работе 3 – Ожидает согласования 4 – На согласовании 5 – Возвращена заявителю 6 – Возвращена исполнителю 7 – Отклонена заявителем 8 – Обработана 9 – Ожидает диспетчеризации 10 – Аннулирована	
startpageentryid	bigint	Идентификатор справочника или классификатора, заданный в шаблоне заявки	reference_book.id
startpageentryname	character varying(254)	Наименование справочника или классификатора, заданное в шаблоне заявки	
templateid	bigint	Идентификатор шаблона заявки	application_template.id
name	character varying(500)	Наименование шаблона заявки	

applicationtype	character varying(500)	Тип заявки, заданный в шаблоне заявки	
comment	character varying(500)	Комментарий к заявке, оставленный заявителем	
dialogtype	integer	Вид отображения значений атрибутов позиции в заявке: 0 – Отображать только исходные значения 1 – Отображать только новые значения 2 – Отображать исходные и новые значения	
itemattrsusage	integer	Способ настройки атрибутов позиции: 0 – атрибуты позиции не задаются в заявке 1 – атрибуты позиции сформированы динамически, на основе ссылки на группу или позицию 2 – атрибуты позиции настроены статически в шаблоне заявки	
sourceelementdescriptor	bigint	Дескриптор группы или позиции, стоя на которой была создана заявка	
transactionid	bigint	Идентификатор пакета изменений, связанного с заявкой	rpt_m2_transaction.id
processingdate	timestamp	Дата и время крайнего срока этапа обработки заявки. Данное поле будет пусто, если заявка еще не перешла на этап обработки или этап обработки был успешно пройден.	
expirationstatus	integer	Состояние этапа обработки заявки: 0 – этап обработки не просрочен 1 – прошел срок предварительного уведомления о необходимости завершить обработку 2 – этап обработки просрочен	
annulmentperiod	bigint	Количество рабочих дней, через которые заявка с открытым пакетом изменений аннулируется, если в течении данного периода над заявкой не выполнялось ни одного действия.	
annulmentdate	timestamp	Дата и время планируемого аннулирования заявки с открытым пакетом изменений, если над заявкой не будет выполнено ни одно действия в течении количества дней, указанных в annulmentperiod.	
pendingannulmentdate	timestamp	Дата и время отправки уведомления о скором аннулировании заявки с открытым пакетом изменений, если над заявкой не будет выполнено ни одно действия в течении количества дней, указанных в annulmentperiod.	
pendingannulmentperiod	bigint	Количество рабочих дней, через которые по заявке с открытым пакетом изменений будет отправлено уведомление о скором аннулировании заявки.	
transactionexpiratio	integer	Состояние заявки по срокам аннулирования: 0 – Непросроченная	

nstatus		1 – Ожидает аннулирования 2 – Аннулирована	
group_	bigint	Идентификатор группы заявок, если данная заявка объединена в группу с другими заявками	
priorityid	bigint	Идентификатор приоритета	
priorityname	character varying(254)	Наименование приоритета	
prioritysortorder	bigint	Порядковый номер приоритета для сортировки	

3.34.5 rpt_application_attribute: Значения атрибутов заявок

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор значения атрибута заявки	
application	bigint	Идентификатор заявки	rpt_application.id
name	character varying(500)	Наименование атрибута в заявке	
templateattributeid	bigint	Идентификатор декларации атрибута заявки	rpt_app_property_declaration.id
attributetype	integer	Тип атрибута: 0 – Атрибут заявки с типом данных «Строка», значение в поле valuestr 1 – Атрибут заявки с типом данных «Текст», значение не доступно для чтения 2 - Атрибут заявки с типом данных «Файл», наименование файла в поле valuestr, тело файла не доступно для чтения 3 – Атрибут заявки с типом данных «Ссылка на группу или позицию». Если прописана ссылка, то значение хранится в поле elementlinkid, если задали наименование вручную, то в elementlinkname 4 – Атрибут заявки с типом данных «Ссылка на атрибут», значение в поле propertyid, наименование атрибута в поле propertyname 5 – Атрибут заявки с типом данных «Ссылка на декларацию», в поле elementlinkid хранится идентификатор группы, в поле propertyid хранится идентификатор атрибута 6 – Значение атрибута позиции. Ссылка	

		на атрибут позиции хранится в propertyid, значение в зависимости от типа данных атрибута храниться в: ·Логический тип - valuebool ·Дата - valuedate ·Число с плав. запятой - valuedblstr ·Целое число - valueint ·Строка - valuestring ·Атрибут связи – elementlinkid ·Цвет – чтение не поддерживается ·Многострочный текст – чтение не поддерживается ·Изображение – чтение не поддерживается ·Файл – наименование файла в поле valuestr, тело файла не доступно для чтения 7 – Атрибут заявки с типом данных «Ссылка на группу», значение хранится в поле elementlinkid, если задали наименование вручную, то в elementlinkname 8 – Атрибут заявки с типом данных «Ссылка на позицию», значение хранится в поле elementlinkid, если задали наименование вручную, то в elementlinkname 9 – Системный атрибут заявки с типом данных «Статус позиции», значение хранится в поле itemstatus	
propertyid	bigint	Идентификатор атрибута позиции	rpt_m2_attribute.id rpt_m2_aggregationstereotype.id
propertyname	character varying(255)	Наименование атрибута позиции	
multiple	boolean	Флаг множественности значения	
mandatoriness	integer	Требования к вводу данных в атрибут: 0 – необязательный 1 – должен быть заполнен заявителем 2 – должен быть заполнен исполнителем	
defaultlink	boolean	Устанавливается для атрибута заявки с типом данных «Ссылка на группу», «Ссылка на позицию», «Ссылка на группу/позицию», если в данный атрибут требуется: ·прописать ссылку на группу/позицию, стоя на которой создавали заявку, ·сформировать список динамических атрибутов позиции заявки	

		·заполнить колонку «Исходные значения» или «Новые значения» при указании ссылки на позицию	
executorsource	boolean	Если данный флаг установлен для атрибута заявки, то значит исполнители заявки будут получены из группы, на которую прописана ссылка в поле elementlinkid. Если в elementlinkid прописана ссылка на позицию, то исполнителей будут получены из группы, в которой находится данная позиция.	
periodssource	boolean	Если данный флаг установлен, то сроки обработки заявки будут получены из группы, ссылка на которую прописана в поле elementlinkid. Если в elementlinkid прописана ссылка на позицию, то исполнителей будут получены из группы, в которой находится данная позиция.	
visiblesince	integer	Видимость атрибута: 0 – виден при создании заявки 1 – должен быть виден, начиная с этапа исполнения	
elementlinkid	bigint	Значение атрибута: Идентификатор позиции или группы	rpt_m2_item.id rtp_m2_classificationnode.id
elementlinkname	character varying(500)	Значение атрибута: Наименование позиции или группы	
itemstatus	integer	Статус позиции: 0 – Не нормализована 1 – Нормализована 2 – Заблокирована 3 – Удалена 4 – Дубликат	
valuebool	boolean	Булевое значение атрибута	
valuedate	timestamp	Значение атрибута: Дата и время	
valuedblstr	character varying(255)	Строковое представление значения атрибута с числом с плавающей точкой	
natureid	bigint	Идентификатор измеряемой сущности, для атрибута позиции с числом с плавающей запятой	rpt_m2_propertygroup.id
desirableunitid	bigint	Идентификатор единицы измерения, которую задал пользователь для значения с числом с плавающей запятой	
valueint	integer	Значение атрибута: Целое число	
valuestring	character varying(500)	Значение атрибута: Строковое значение	

elementlinkdeleted	boolean	Булевый флаг установки пометки на удаление позиции, указанной в поле elementlinkid	
branchid	bigint	Идентификатор позиции, для которой в заявке имеются вложенные атрибуты. Если числовое значение положительное, то это идентификатор существующей позиции в справочнике. Если идентификатор отрицательный, то это идентификатор виртуальной позиции, которую заявитель просит создать.	rpt_m2_item.id
leafid	bigint	Идентификатор позиции, для которой задано значение в данном атрибуте. Если значение NULL, то данный атрибут относится к корневой позиции.	

3.34.6 rpt_application_participant: история изменения статусов заявок и их участников

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
Id	bigint	Уникальный числовой идентификатор события со статусом по заявке	
applicationid	bigint	Числовой идентификатор заявки	application.id
datecreated	timestamp	Дата и время перехода заявки на статус, указанного в поле status	
datefinished	timestamp	Дата и время завершения статуса для заявки, указанного в поле status. Если статус не завершен, то в данном поле прописан null.	
status	integer	Статус заявки, на котором заявка находилась в период с datecreated по datefinished: 0 – Зарегистрирована 1 – Ожидает обработки 2 – В работе 3 – Ожидает согласования 4 – На согласовании 5 – Возвращена заявителю 6 – Возвращена исполнителю 7 – Отклонена заявителем 8 – Обработана 9 – Ожидает диспетчеризации 10 – Аннулирована	
nextstatus	integer	Последующий статус заявки, который был установлен заявке, после завершения статуса из поля status:	

		0 – Зарегистрирована 1 – Ожидает обработки 2 – В работе 3 – Ожидает согласования 4 – На согласовании 5 – Возвращена заявителю 6 – Возвращена исполнителю 7 – Отклонена заявителем 8 – Обработана 9 – Ожидает диспетчеризации 10 – Аннулирована Если статус из поля status для заявки еще не завершен, то данное в данном поле храниться null	
account	character varying(255)	Логин пользователя, для которого действовал статус из поля status и который перевел заявку на новый статус из поля nextstatus.	rpt_m2_account.login

3.34.7 rpt_app_status_journal: журнал заявки

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
Id	bigint	Уникальный числовой идентификатор события в журнале заявки	
applicationid	bigint	Числовой идентификатор заявки	application.id
datecreated	timestamp	Дата и время перехода заявки из статуса previousstatus на статус status	
status	integer	Статус, который заявка получила в указанную дату и время(datecreated): 0 – Зарегистрирована 1 – Ожидает обработки 2 – В работе 3 – Ожидает согласования 4 – На согласовании 5 – Возвращена заявителю 6 – Возвращена исполнителю 7 – Отклонена заявителем 8 – Обработана 9 – Ожидает диспетчеризации 10 – Аннулирована	
previousstatus	integer	Предшествующий статус заявки, который у нее был: 0 – Зарегистрирована 1 – Ожидает обработки 2 – В работе 3 – Ожидает согласования 4 – На согласовании	

		5 – Возвращена заявителю 6 – Возвращена исполнителю 7 – Отклонена заявителем 8 – Обработана 9 – Ожидает диспетчеризации 10 – Аннулирована Внимание! Значение статуса может быть идентично значению в поле status, для случае передачи заявки от одного эксперта другому или в случае создания заявки.	
account	character varying(255)	Логин пользователя, который перевел заявку из статуса previousstatus в status	rpt_m2_account.login
previousexpirationstatus	integer	Предшествующий статус заявки, который был просрочен: 0 – Зарегистрирована 1 – Ожидает обработки 2 – В работе 3 – Ожидает согласования 4 – На согласовании 5 – Возвращена заявителю 6 – Возвращена исполнителю 7 – Отклонена заявителем 8 – Обработана 9 – Ожидает диспетчеризации 10 – Аннулирована	
previoustransactionexpirationstatus	integer	Предыдущей статус превышения сроков обработки заявки с открытым пакетом изменений: 0 – Сроки обработки заявки с открытым пакетом изменений не превышен 1 – Отправлено уведомление о превышении сроков обработки заявки с открытым пакетом изменений 2 – Заявка аннулирована, пакет изменений отменен	
transactionexpirationstatus	integer	Текущий статус превышения сроков обработки заявки с открытым пакетом изменений: 0 – Сроки обработки заявки с открытым пакетом изменений не превышен 1 – Отправлено уведомление о превышении сроков обработки заявки с открытым пакетом изменений 2 – Заявка аннулирована, пакет изменений отменен	
stageinfos	jsonb	Информация об этапе согласования в формате JSON	

3.34.8 rpt_reference_book: реестр справочников

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор справочника	
uuid	character varying(255)	GUID справочника	
name	character varying(255)	Наименование справочника	
applicationsprocessingnorm	integer	Норма обработки заявок экспертом на справочник	

3.34.9 rpt_reference_book_content: группы, входящие в справочники

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор	
element	bigint	Идентификатор группы, которая входит в справочник	rtp_m2_classificationnode.id
referencebook	bigint	Идентификатор справочника	rpt_reference_book.id

3.34.10 rpt_m2_account: реестр учетных записей пользователей

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор пользователя	
uuid	character varying(255)	GUID пользователя	
isdisabled	boolean	Флаг блокировки пользователя	
ip	character varying(255)	Последний IP с которого подключался пользователь к системе	
login	character varying(255)	Логин	
first_name	character varying(255)	Имя	
last_name	character varying(255)	Фамилия	
middle_name	character varying(255)	Отчество	
email	character varying(255)		
date_created	timestamp	Дата создания пользователя	
linkedite	bigint	Идентификатор позиции в справочнике,	rpt_m2_item.id

mid		связанный с данной учетной записью	
lastlogindate	timestamp	Дата время последнего входа в систему	
identitysource	integer	Тип пользователя: 0 – пользователь Semantic MDM 1 – гость 2 – пользователь Active Directory	
isapplicant	boolean	True – если пользователь создал хотя бы одну заявку	
isexecutor	boolean	True – если пользователь обработал хотя бы одну заявку	
isnegotiant	boolean	True – если пользователь согласовал хотя бы одну заявку	

3.34.11 rpt_m2_security_group: группы пользователей

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор группы пользователей	
uuid	character varying(255)	GUID группы пользователей	
ldapsynced	boolean	Флаг синхронизации через LDAP	
name	character varying(255)	Наименование группы	
description	character varying(255)	Описание группы пользователей	

3.34.12 rpt_m2_account_group: Вхождение пользователей в группу пользователей

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
acc_id	bigint	Идентификатор пользователя	rpt_m2_account.id
group_id	bigint	Идентификатор группы пользователей	rpt_m2_security_group.id

3.34.13 rpt_m2_hierarchyhandle: заголовок группы или позиции

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный идентификатор заголовка группы или позиции. После изменения заголовка, в результате изменения наименования позиции/группы или статуса позиции, создается	

		новый заголовок, с новым ID. После публикации пакета изменений, актуальным будет новый заголовок с новым ID, в случае отмены пакета изменений, актуальным будет старый заголовок со старым ID	
descriptor	bigint	Уникальный числовой идентификатор группы или позиции, которым оперирует Система. Данные идентификатор не изменяется при изменении заголовка позиции или группы.	
displayname	character varying(1000)	Наименование позиции или группы, зависит от значения флага elementisnode.	
element	bigint	Идентификатор группы или позиции зависит от значения флага elementisnode.	rpt_m2_classificationnode.id rpt_m2_item.id
parent	bigint	Идентификатор группы	rpt_m2_classificationnode.id
elementisnode	boolean	True – элемент является группой, False – элемент является позицией	
transaction_prnt	bigint	Идентификатор транзакции, в результате публикации которой, был изменен заголовок позиции или группы. Для позиции, заголовок изменяется в случае изменения наименования позиции или статуса. Для группы, заголовок изменяется в случае изменения наименования группы, комментария или картинки.	rpt_m2_transaction.id
initialtransid	bigint	Идентификатор транзакции, в результате публикации которой, был создан новый заголовок позиции или группы.	rpt_m2_transaction.id
itemstatus	integer	Если elementidnode=False, то статус позиции: 0 – Не нормализована 1 – Нормализована 2 – Заблокирована 3 – Удалена 4 – Дубликат	

3.34.14 rpt_m2_item: позиции справочников

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор позиции	
uid	character varying(255)	GUID позиции	
nestedleft	numeric(38,0)	Левая граница интервала. Все другие записи данной таблицы, интервалы которых входят в данный диапазон являются дочерними. Все другие записи	

		данной таблицы, в интервал которых входит текущий интервал, являются родительскими.	
nestedright	numeric(38,0)	Правая граница интервала. Все другие записи данной таблицы, интервалы которых входят в данный диапазон являются дочерними. Все другие записи данной таблицы, в интервал которых входит текущий интервал, являются родительскими.	

3.34.15 rtp_m2_classificationnode: группы справочников

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор группы	
uid	character varying(255)	GUID группы	
nestedleft	numeric(38,0)	Левая граница интервала. Все другие записи данной таблицы, интервалы которых входят в данный диапазон являются дочерними. Все другие записи данной таблицы, в интервал которых входит текущий интервал, являются родительскими.	
nestedright	numeric(38,0)	Правая граница интервала. Все другие записи данной таблицы, интервалы которых входят в данный диапазон являются дочерними. Все другие записи данной таблицы, в интервал которых входит текущий интервал, являются родительскими.	

3.34.16 rpt_m2_attributevalue: значения атрибутов позиций

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор позиции. Данный идентификатор изменяется при изменении значения атрибута позиции	
descriptor	bigint	Уникальный числовой идентификатор значения, которым оперирует Система. Данные идентификатор не изменяется при изменении значения атрибута позиции.	
Item_id	bigint	Идентификатор позиции	rpt_m2_item.id
attribute	bigint	Идентификатор атрибута	rpt_m2_attribute.id
valuebool	boolean	Булево значение атрибута	

valuedate	timestamp	Значение атрибута типа Дата	
valueint	integer	Целочисленное значение атрибута	
valuestr	character varying(500)	Строковое значение атрибута	
valuedblstr	character varying(255)	Строковое предоставление параметров значения числа с плав. Запятой	
has_blob_val	boolean	True – есть значение, имя файла хранится в valuestr; False – нет значения	
has_clob_val	boolean	True – есть значение; False – нет значения	
transaction_prnt	bigint	Идентификатор транзакции, в результате публикации которой, значение атрибута было изменено	rpt_m2_transaction.id
initialtransid	bigint	Идентификатор транзакции, в результате публикации которой, значение атрибута было впервые задано	rpt_m2_transaction.id
changedate	timestamp	Дата и время изменения	
account_id	bigint	Идентификатор пользователя	rpt_m2_account.id

3.34.17 rpt_m2_aggregation: значения атрибута связи

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный идентификатор значения атрибута связи. Изменяется после изменения и публикации нового значения атрибута связи.	
descriptor	bigint	Уникальный идентификатор значения атрибута связи, которым оперирует Система. Данный идентификатор не изменяется при изменении значения атрибута связи.	
aggregationstereotype	bigint	Идентификатор атрибута связи, для которого в позиции aggregate определено значение part.	rpt_m2_aggregationstereotype.id
aggregate	bigint	Идентификатор позиции, для которого в атрибуте связи aggregationstereotype определено значение part.	rpt_m2_item.id
part	bigint	Значение (идентификатор позиции) атрибута связи aggregationstereotype для позиции aggregate	rpt_m2_item.id
transaction_prnt	bigint	Идентификатор транзакции, в результате публикации которой, ссылка на позицию была изменена	rpt_m2_transaction.id
initialtransid	bigint	Идентификатор транзакции, в результате публикации которой, ссылка на другую позицию была впервые создана	rpt_m2_transaction.id
changedate	timestamp	Дата и время изменения	
account_id	bigint	Идентификатор пользователя	rpt_m2_account.id

3.34.18 rpt_m2_attributedeclaration: декларации атрибутов в справочниках

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор декларации атрибута. Изменяется, после изменения декларации через пакет изменений.	
descriptor	bigint	Дескриптор декларации атрибута, не меняется после изменения декларации через пакет изменений	
node	bigint	Идентификатор группы, в которой продекларирован атрибут	
attribute	bigint	Идентификатор продекларированного атрибута	
ml_name	character varying(255)	Перекрытое наименование декларации атрибута, которое отображается пользователю. Если наименование не задано, то наименование берётся с атрибута	
nestedleft	bigint	Значение интервала. Позволяет быстро получить все декларации, которые входят в текущую и все подчиненные группы. Если получить декларации для текущей и родительских групп, затем отсортировать полученные декларации по полю nestedleft по возрастанию, то перекрытые декларации будут в полученном списке находится первыми.	
sortorder	numeric(38,27)	Порядок сортировки декларации в карточке позиции. Если объединить с rpt_m2_attributeaggregation, то позволяет после сортировки получить полный список деклараций в нужно порядке.	rpt_m2_attribute.id
category_name	character varying(255)	Наименования категории атрибута	
has_formula	boolean	True – атрибут вычисляемый, False – атрибут не	

		вычисляемый	
is_displayname	boolean	True – атрибут содержит наименование позиции, False – атрибут не содержит наименование позиции	
metaproperty	integer	Содержит связь с атрибутом метаданных справочника пользователей или единиц измерения. Может принимать следующие значения: Null или 0 – нет связи с метаданными Если группа связана со справочником ЕИ: 1 – Наименование ЕИ 2 – Измеряемая сущность 3 – Код обозначения ЕИ 4 – Признак базовой ЕИ 5 – Коэффициент пересчета ЕИ в базовую 6 – Формула пересчета ЕИ в базовую 7 – Формула пересчета ЕИ из базовой Если группа связана со справочником пользователей: 8 – Логин пользователя 10 – Фамилия 11 – Имя 12 – Отчество 13 – Электронная почта 14 – Блокировка пользователя 15 – Дополнительный атрибут, загружаемый из LDAP	

3.34.19 rpt_m2_aggregationdeclaration: декларации атрибутов связи в справочниках

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор декларации атрибута связи. Изменяется, после изменения декларации через пакет изменений.	
descriptor	bigint	Дескриптор декларации атрибута связи, не меняется после изменения декларации через пакет изменений	
node	bigint	Идентификатор группы, в которой продекларирован атрибут	

stereo	bigint	Идентификатор продекларированного атрибута связи	
nestedleft	bigint	Значение интервала. Позволяет быстро получить все декларации, которые входят в текущую и все подчиненные группы. Если получить декларации для текущей и родительских групп, затем отсортировать полученные декларации по полю nestedleft по возрастанию, то перекрытые декларации будут в полученном списке находится первыми.	
sortorder	numeric(38,27)	Порядок сортировки декларации в карточке позиции. Если объединить с rpt_m2_attributedeclaration, то позволяет после сортировки получить полный список деклараций в нужно порядке.	
category_name	character varying(255)	Наименования категории атрибута связи	
has_formula	boolean	True – атрибут вычисляемый, False – атрибут не вычисляемый	

3.34.20 rpt_m2_attribute: простые атрибуты для справочников

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор атрибута	
uuid	character varying(255)	GUID атрибута	
eotdid	character varying(255)	EOTD атрибута	rpt_m2_attribute.id
ml_name	character varying(255)	Наименования атрибута	
adatatype		Тип данных атрибута: 0 – Строка, 1 – Число с плавающей запятой, 2 – Целое число, 3 – Логический тип, 4 – Изображение, 5 – Многострочный текст, 6 – Дата, 7 – Цвет, 8 – Файл, 9 – Формула.	
ismultiple	boolean	Флаг множественного значения	
namedvalues	boolean	Содержит контекстные значения	
nature	bigint	Идентификатор измеряемой сущности	rpt_m2_propertygroup.

			id

3.34.21 rpt_m2_aggregationstereotype: атрибуты связи для справочников

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор атрибута связи	
uuid	character varying(255)	GUID атрибута связи	
eotdid	character varying(255)	EOTD атрибута связи	
ml_name	character varying(255)	Наименование атрибута связи	
ml_aggregatename	character varying(255)	Обратное наименования атрибута связи	
iscomposite	boolean	Флаг композиции	
ismultiple	boolean	Флаг множественного значения	
hierarchical	boolean	Флаг использования связи для построения иерархии	

3.34.22 rpt_m2_propertygroup: измеряемые сущности

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор измеряемой сущности	
uuid	character varying(255)	GUID измеряемой сущности	
ml_name	character varying(255)	Наименование измеряемой сущности	

3.34.23 rpt_m2_measurementunit: единицы измерений

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор единицы измерения	
uuid	character varying(255)	GUID единицы измерения	
ml_code	character varying(255)	Обозначения единицы измерения	
ml_name	character varying(255)	Наименования единицы измерения	
nature	bigint	Идентификатор измеряемой сущности	rpt_m2_prop

			ertygroup.id
factor	double precision	Коэффициент пересчета из базовой	
formula	character varying(255)	Формула пересчета из базовой	
rev_formula	character varying(255)	Формула пересчета в базовую	
linkeditemid	bigint	Идентификатор позиции в справочнике, связанный с данной единицей измерения	

3.34.24rpt_m2_constraint: Ограничения

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор ограничения. Изменяется после изменения ограничения через пакет изменений.	
descriptor	bigint	Уникальный числовой идентификатор, которым оперирует система. Не изменяется после изменения ограничения через пакет изменений.	
uuid	character varying(255)	GUID ограничения	
node	bigint	Идентификатор группы, на которую назначено данное ограничение	rpt_m2_classificationnode.id
ml_name	character varying(255)	Наименования ограничения	
is_mandatory	boolean	Обязательное ограничение	
nestedleft	numeric(38, 0)	Значение интервала. Позволяет быстро получить для группы все ограничения, которые входят в текущую и все подчиненные группы.	
constraint type	integer	Тип ограничения: 0 – ограничительные таблицы 1 – обязательные атрибуты 2 – уникальные атрибуты 3 – функции проверки данных	
ctable	bigint	Идентификатор ограничительной таблицы	rpt_m2_constrainttable.id
constrainedproperty	bigint	Идентификатор атрибута, на который накладывается ограничение уникальности или обязательного ввода: constrainttype in (1,2)	rpt_m2_attribute.id

3.34.25 rpt_m2_constrainttable: Табличные ограничения

Наименование	**Тип данных**	**Описание поля**	**Ссылка на таблицу**

поля			
id	bigint	Уникальный числовой идентификатор ограничительной таблицы.	
metadatadelegate	boolean	True – глобальная таблица ограничений, False – локальная таблица ограничений.	
foritemgeneration	boolean	Использовать таблицу для создания позиций	
uuid	character varying(255)	GUID ограничительной таблицы	

3.34.26 rpt_m2_tablecolumn: Колонки ограничительной таблицы

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор колонки ограничительной таблицы.	
descriptor	bigint	Уникальный числовой идентификатор колонки ограничительной таблицы, которым оперирует Система.	
ownertable	bigint	Идентификатор ограничительной таблицы	rpt_m2_constrainttable.id
meaning	bigint	Идентификатор атрибута, на значения которого колонка ограничительной таблицы накладывает ограничение.	rpt_m2_attribute.id rpt_m2_aggregationstereotype.id
uuid	character varying(255)	GUID колонки ограничительной таблицы	

3.34.27 rpt_constraint_compliance: Соответствие позиций ограничениям

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор соответствия ограничению. Изменяется в случае изменения статуса соответствия.	
descriptor	bigint	Уникальный числовой идентификатор соответствия ограничению, которым оперирует Система. Не изменяется после изменения статуса соответствия.	
constraint_	bigint	Идентификатор ограничения, для которого рассчитан данный статус соответствия	rpt_m2_constraint.id
item	bigint	Идентификатор позиции, для которой вычислено соответствие ограничению. На одну позицию, вычисляется соответствие для всех ограничений, наложенных на нее.	rpt_m2_item.id

rowdescriptor	bigint	Дескриптор строки ограничительной таблицы, которой соответствует позиция.	rpt_m2_tablerow.descriptor
complliancestatus	integer	Статус соответствия ограничению: 0 – Соответствует ограничению 1 – Не соответствует необязательному ограничению 2 – Не соответствует обязательному ограничению	
reason	character varying(255)	Причина несоответствия ограничению, для формульного ограничения	

3.34.28 rpt_transitional_key: таблица переходных ключей

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор переходного ключа. Изменяется в случае изменения переходного ключа.	
descriptor	bigint	Уникальный числовой идентификатор соответствия ограничению, которым оперирует Система. Не изменяется после изменения переходного ключа.	
item	bigint	Идентификатор позиции, для которой определен переходной ключ	rpt_m2_item.id
value	character varying(255)	Идентификатор позиции в системе externalsystem	
externalsystem	bigint	Идентификатор внешней системы	rpt_externalsystem.id
duplicate	boolean	True – дубликат, False – оригинал. Один из переходных ключей для одной системы и одной позиции всегда False.	
transaction_prnt	bigint	Идентификатор транзакции, в результате публикации которой, был изменен переходной ключ	rpt_m2_transaction.id
initialtransid	bigint	Идентификатор транзакции, в результате публикации которой, переходной ключ был впервые создан	rpt_m2_transaction.id

3.34.29 rpt_external_system: внешние системы

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор внешней системы.	
displayname	character varying(255)	Наименование внешней системы для пользователя	
systemid	character varying(255)	Идентификатор внешней системы для интеграционных адаптеров	
company	character varying(255)	Наименование организации, где развернута данная система	

transport	integer	Способ передачи данных: 0 – Messag Queue 1 – REST/SOAP 2 – Репликация в дочерний Semantic MDM 3 – Репликация в центральный Semantic MDM	
hasfailures	boolean	Есть сбои в интеграции	

3.34.30 rpt_exchange_class: обменный класс

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор обменного класса	
displayname	character varying(255)	Наименование обменного класса	
classid	character varying(255)	Идентификатор обменного класса для интеграционного адаптера	
status	integer	Статус обменного класса: 0 – Остановлен 1 – Приостановлен 2 – Выполняется	

3.34.31 rpt_similarity_search_option: варианты поиска дубликатов

Наименование поля	**Тип данных**	**Описание поля**	**Ссылка на таблицу**
id	bigint	Уникальный числовой идентификатор варианта поиска дубликатов	
node	bigint	Идентификатор группы, для которого зарегистрирован вариант поиска дубликатов	rpt_m2_classificationnode.id
displayname	character varying(255)	Наименование варианта поиска дубликатов	
sourcefilter	jsonb	JSON с условиями фильтрации позиций в текущей группе	
targetfilter	jsonb	JSON с условиями фильтрации позиций в проверяемой группе	
targetnodeids	character varying(255)	Список идентификаторов групп, в которой ищем дубликаты	rpt_m2_classificationnode.id
usewhileimportmatching	boolean	Использовать для сопоставления при импорте	

3.34.32 rpt_similarity_check: список позиций, для которых искали дубликаты

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор	
option	bigint	Идентификатор варианта поиска дубликатов	rpt_similarity_search_option.id
account	bigint	Идентификатор пользователя, который выполнил поиск дубликатов	rpt_m2_account.id
sourceelement	bigint	Идентификатор позиции или группы, для которой запускали поиску дубликатов	rpt_m2_item.id rpt_m2_classificationnode.id
checkeditem	bigint	Идентификатор позиции для которой искали дубликаты	rpt_m2_item.id
filteringcomposeid	bigint	Идентификатор позиции владельца композируемых позиций, среди которых искались дубликаты	rpt_m2_item.id
filteringstereoid	bigint	Идентификатор атрибута связи, если поиск дубликатов ограничивался композируемыми	rpt_m2_aggregation.id

3.34.33 rpt_similar_item: список найденных дубликатов

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор	
similaritycheck	bigint	Идентификатор исходной позиции и поиска дубликатов, в рамках которого в этой таблице зарегистрирован дубликат	rpt_similarity_check.id
coincidencerate	integer	Точность совпадения в процентах	
item	bigint	Идентификатор позиции дубликата	rpt_m2_item.id

3.34.34 rpt_m2_transaction: пакеты изменения

Наименование поля	Тип данных	Описание поля	Ссылка на таблицу
id	bigint	Уникальный числовой идентификатор пакета изменений	
uuid	character varying(255)	GUID пакета изменений	
isactive	boolean	True – активный, False – не активный	
name	character varying(255)	Наименование пакета изменений	

startdate	timestamp	Дата и время создания пакета изменений	
lastchangedate	timestamp	Дата и время последнего изменения в пакете изменений	
finishdate	timestamp	Дата и время публикации пакета изменений	
owner	bigint	Идентификатор пользователя, который владеет данным пакетом	rpt_m2_account.id

3.34.35 История по позициям

Для получения исторических данных по созданию, удалению позиции, а также изменению статуса и родительской группы позиции, следует использовать следующие view с объединением через union all:

- **rpt_current_created_items** – позволяет определить в какой группе и с каким статусом была создана позиция. В данной view будут записи, если после создания позиции не выполнялось изменение статуса и/или группы;
- **rpt_obsolete_created_items** – позволяет определить в какой группе и с каким статусом была создана позиция. В данной view будут записи, если после создания позиции проводилось изменение статуса и/или группы;
- **rpt_current_changed_items** – позволяет определить последние изменения, которые были выполненные по статусу или группе, в которой находится данная позиция
- **rpt_obsolete_changed_items** – позволяет определить изменения по статусу или группе, в которые были выполнены ранее. Данная view не включает в себя последнее выполненное изменение из **rpt_current_changed_item**;
- **rpt_obsolete_deleted_items** – позволяет определить когда были удалены позиции.

Для получения консолидированного представления по истории позиций, необходимо выполнить запрос:

```
select *, 1 as action_type -- создание позиции
```

```
  from rpt_current_created_items
union all
select *, 1 as action_type -- создание позиции
  from rpt_obsolete_created_items
union all
select *, 2 as action_type -- изменение позиции
  from rpt_current_changed_items
union all
select *, 2 as action_type -- изменение позиции
  from rpt_obsolete_changed_items
union all
select *, 3 as action_type -- удаление позиции
  from rpt_obsolete_deleted_items
```

Поля для всех перечисленных выше view полностью идентичны и описаны в Таблица 34.

Таблица 34 – Атрибуты истории по значениям

Наименование поля	**Тип данных**	**Описание поля**	**Ссылка на таблицу**
descriptor	bigint	Дескриптор поизции	
item_id	bigint	Идентификатор позиции	rpt_m2_item.id
transactionid	bigint	Идентификатор транзакции	rpt_m2_transaction.id
old_itemstatus	integer	Предыдущий статус позиции: 0 – Не нормализована 1 – Нормализована 2 – Заблокирована 3 – Удалена 4 – Дубликат	
new_itemstatus	integer	Новый статус позиции: 0 – Не нормализована 1 – Нормализована 2 – Заблокирована 3 – Удалена 4 – Дубликат	
old_parentid	bigint	Предыдущий идентификатор группы, в которой находится позиция	rpt_m2_classificationnode.id
new_pare	bigint	Новый идентификатор группы, в которой	rpt_m2_classific

ntid		находится позиция	ationnode.id
changedate	timestamp	Дата и время изменения	
account_id	bigint	Идентификатор пользователя	rpt_m2_account.id

3.34.36 История изменения атрибутов связи

Для получения исторических данных по созданию, удалению и изменению значений атрибутов связи, следует использовать следующие view с объединением через union all:

- **rpt_current_created_aggregation** – позволяет определить, в каком атрибуте связи было создано новое значение (т.е. до этого атрибут связи не имел значения). В данной view будут записи, если новое значение является актуальным для позиции;
- **rpt_obsolete_created_aggregation** – позволяет определить, в каком атрибуте связи было создано новое значение (т.е. до этого атрибут связи не имел значения). В данной view будут записи, если новое значение уже не является актуальным для позиции;
- **rpt_current_changed_aggregation** – позволяет определить, в каком атрибуте связи было изменено значение. В данной view будут записи, если новое значение является актуальным для позиции;
- **rpt_obsolete_changed_aggregation** – позволяет определить, в каком атрибуте связи было изменено значение. В данной view будут записи, если новое значение уже не является актуальным для позиции;
- **rpt_obsolete_deleted_aggregation** – позволяет определить когда были удалены значения из атрибута связи.

Для получения консолидированного представления по истории значений атрибута связи, необходимо выполнить запрос:

```
select *, 1 as action_type -- создание значения
  from rpt_current_created_aggregation
union all
```

```
select *, 1 as action_type -- создание значения
  from rpt_obsolete_created_aggregation
union all
select *, 2 as action_type -- изменение значения
  from rpt_current_changed_aggregation
union all
select *, 2 as action_type -- изменение значения
  from rpt_obsolete_changed_aggregation
union all
select *, 3 as action_type -- удаление значения
  from rpt_obsolete_deleted_aggregation
```

Для получения информации о значениях атрибута связи, которые были после создания позиции, необходимо использовать следующие view: **rpt_current_initial_item_aggregation** и **rpt_obsolete_initial_item_aggregation**.

Поля для всех перечисленных выше view полностью идентичны и описаны в таблице ниже.

Таблица 35 – Атрибуты истории изменения атрибутов связи

Наименование поля	**Тип данных**	**Описание поля**	**Ссылка на таблицу**
descriptor	bigint	Дескриптор позиции	
item_id	bigint	Идентификатор позиции, для которой было изменено значение атрибута связи	rpt_m2_item.id
propertyid	bigint	Идентификатор атрибута связи, для которого было изменено значение	rpt_m2_aggregation.id
transactionid	bigint	Идентификатор транзакции, в рамках которой было изменено значение атрибута связи	rpt_m2_transaction.id
old_part_id	bigint	Старое значение атрибута связи (ссылка на позицию)	rpt_m2_item.id
new_part_id	bigint	Новое значение атрибута связи (ссылка на позицию)	rpt_m2_item.id
changedate	bigint	Дата изменения значения атрибута связи	
account_id	bigint	Идентификатор пользователя, который изменил значение	rpt_m2_account.id

3.34.37 История изменения простых атрибутов

Для получения исторических данных по созданию, удалению и изменению значений простых атрибутов, следует использовать следующие view с объединением через union all:

- **rpt_current_created_values** – позволяет определить, в каком атрибуте было создано новое значение (т.е. до этого атрибут не имел значения). В данной view будут записи, если новое значение является актуальным для позиции.
- **rpt_obsolete_created_values** – позволяет определить, в каком атрибуте было создано новое значение (т.е. до этого атрибут не имел значения). В данной view будут записи, если новое значение уже не является актуальным для позиции.
- **rpt_current_changed_values** – позволяет определить, в каком атрибуте было изменено значение. В данной view будут записи, если новое значение является актуальным для позиции.
- **rpt_obsolete_changed_values** – позволяет определить, в каком атрибуте было изменено значение. В данной view будут записи, если новое значение уже не является актуальным для позиции.
- **rpt_obsolete_deleted_values** – позволяет определить когда были удалены значения из атрибута

Для получения консолидированного представления по истории значений атрибута связи, необходимо выполнить запрос:

```
select *, 1 as action_type -- создание значения
  from rpt_current_created_values
union all
select *, 1 as action_type -- создание значения
  from rpt_obsolete_created_values
union all
select *, 2 as action_type -- изменение значения
  from rpt_current_changed_values
```

union all

select *, 2 as action_type -- изменение значения

from rpt_obsolete_changed_values

union all

select *, 3 as action_type -- удаление значения

from rpt_obsolete_deleted_values

Для получения информации о значениях атрибута, которые были после создания позиции, необходимо использовать следующие view: **rpt_current_initial_item_values** и **rpt_obsolete_initial_item_values.**

Поля для всех перечисленных выше view полностью идентичны и описаны в таблице ниже

Таблица 36. Атрибуты истории изменения простых атрибутов

Наименование поля	**Тип данных**	**Описание поля**	**Ссылка на таблицу**
descriptor	bigint	Дескриптор позиции	
item_id	bigint	Идентификатор позиции, для которой было изменено значение атрибута	rpt_m2_item.id
propertyid	bigint	Идентификатор атрибута связи, для которого было изменено значение	rpt_m2_aggregation.id
transactionid	bigint	Идентификатор транзакции, в рамках которой было изменено значение атрибута	rpt_m2_transaction.id
old_bool	boolean	Старое значение атрибута	
new_bool	boolean	Новое значение атрибута	
old_date	timestamp	Старое значение атрибута	
new_date	timestamp	Новое значение атрибута	
old_int	integer	Старое значение атрибута	
new_int	integer	Новое значение атрибута	
old_str	character varying(500)	Старое значение атрибута	
new_str	character varying(500)	Новое значение атрибута	
old_dblstr	character varying(255)	Старое значение атрибута	
new_dblstr	character varying(	Новое значение атрибута	

	255)		
old_blob_id	bigint	Старое значение атрибута	
new_blob_id	bigint	Новое значение атрибута	
old_clob_id	bigint	Старое значение атрибута	
new_clob_id	bigint	Новое значение атрибута	
changedate	bigint	Дата изменения значения атрибута связи	
account_id	bigint	Идентификатор пользователя, который изменил значение	rpt_m2_account.id

3.34.38 Хранение данных по заявкам

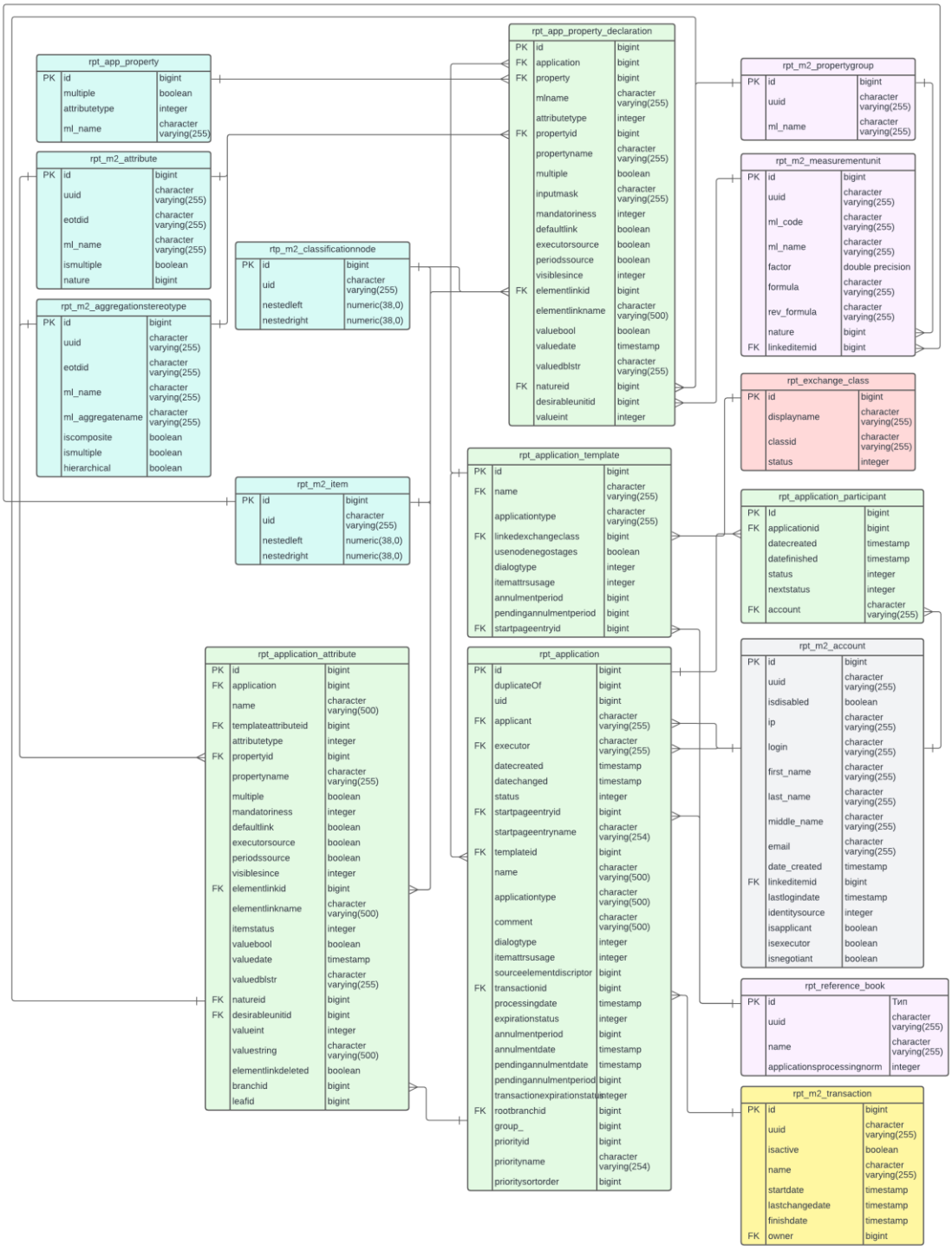


Рисунок 521 – Схема хранения данных по заявкам

3.34.39 Хранение данных по НСИ

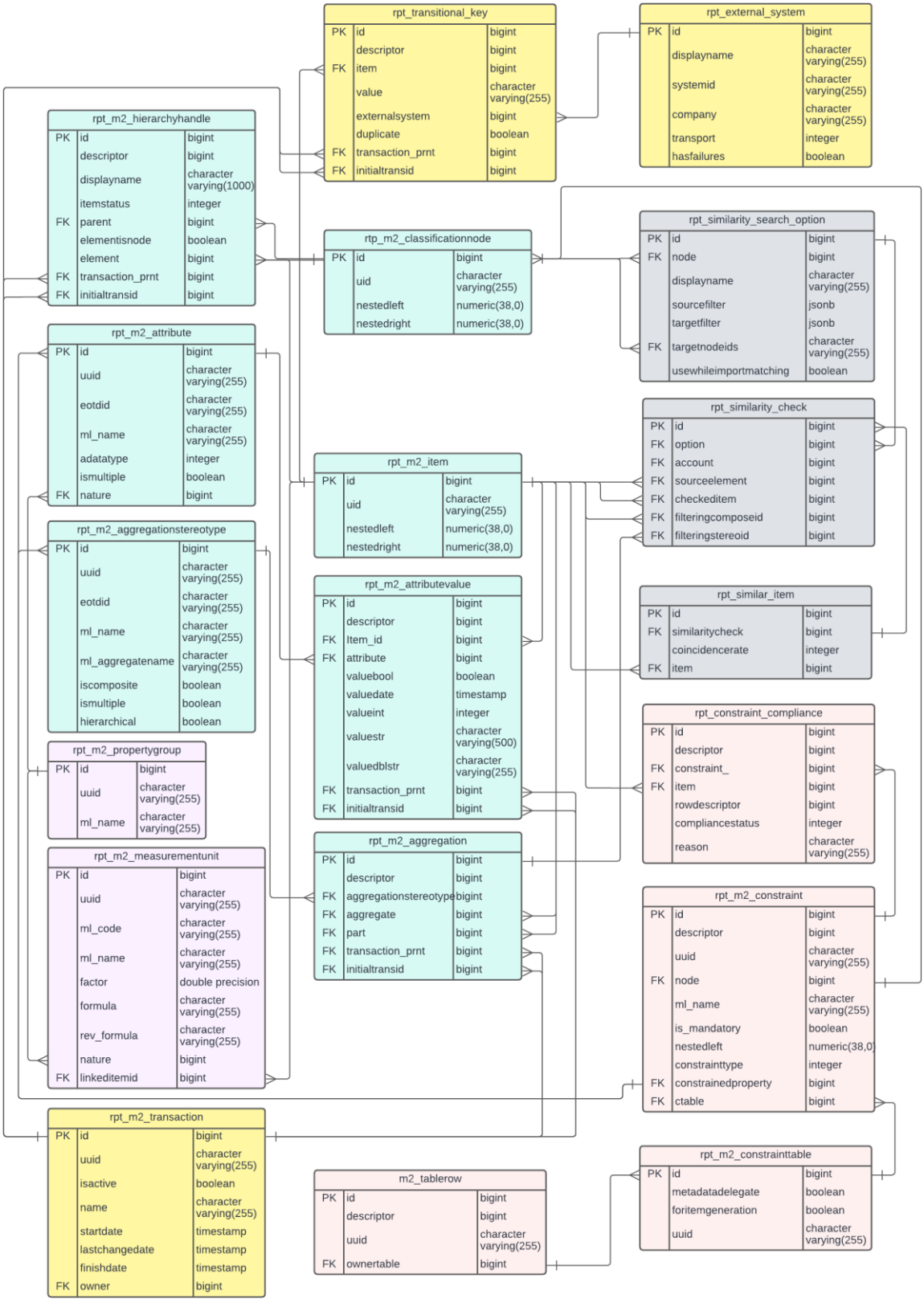


Рисунок 522 – Схема хранения данных по НСИ

4 Аварийные ситуации

Аварийные ситуации, по которым регламентируются требования к показателям надежности системы:

- сбой или отказ оборудования и программного обеспечения по техническим причинам;
- сбои или прекращение электропитания серверного оборудования;
- отказ активного сетевого оборудования (коммуникационного оборудования).

Одновременный отказ основного и резервного (в том числе запасного) элемента СТИ, невыполнение производителем обязательств по технической поддержке или несоответствие фактических показателей элементов СТИ паспортным считаются форс-мажорными обстоятельствами.

Для аварийных ситуаций, связанных с отказом внешних по отношению к *Системе* телекоммуникационных компонентов (каналов связи), требования надёжности не предъявляются.

4.1 Действия при отказах технических средств

В случае отказов технических средств Системы обслуживающий персонал (администратор, специалист по техническому обслуживанию) обязан принять следующие меры к восстановлению их работоспособности:

- выявление точки отказа;
- диагностика неисправности;
- устранение неисправности (исправление программной или технической составляющей).

После восстановления электропитания должна быть обеспечена процедура восстановления *Системы*.

4.2 Действия по восстановлению информации при отказе системы хранения данных или обнаружении ошибок в данных

Для восстановления программ и/или данных при отказе носителей или обнаружении ошибок в данных обслуживающий персонал (администратор, специалист по техническому обслуживанию) должен выполнить следующие действия:

- выполнить остановку *Системы*;
- для восстановления программ произвести инсталляцию программного обеспечения с оригинальных носителей;
- восстановить данные с сервера резервного копирования или резервной копии накопителя.

4.3 Действия при обнаружении несанкционированного вмешательства в данные

В случае обнаружения несанкционированного доступа к данным обслуживающий персонал (администратор, специалист по техническому обслуживанию) обязан принять меры по устранению канала несанкционированного доступа к данным и восстановлению средств защиты от несанкционированного доступа к Системе.

4.3.1.1 Обеспечение безопасности информации

Организационные и технические меры по обеспечению информационной безопасности должны включать:

- антивирусную защиту;
- контроль физического доступа к серверному оборудованию;
- аудит доступа к информационной системе.

На организационном уровне должны быть предусмотрены:

- служба, отвечающая за защиту от НСД, распределение обязанностей;
- комплекс профилактических мер (предупреждение появления вирусов, предупреждение неумышленных действий, ведущих к НСД, и т.д.);

- правила доступа сторонних организаций к ресурсам;
- организация доступа пользователей и персонала к ресурсам;
- регламентирование:
  - удаленного доступа в ИС;
  - использования открытых ресурсов (сети Интернет);
  - использования сертифицированного ПО и т.д.;
- внесение в должностные инструкции персонала соответствующих разделов;
- обучение персонала соблюдению режима защиты от НСД, правильному обращению с информационными ресурсами;
- регламент взаимодействия с представителями сторонних организаций;
- меры по выявлению и предотвращению утечки информации, а также поддержанию надлежащего уровня обеспечения безопасности.

На техническом уровне защиты от НСД должны быть предусмотрены:

- применение технических средств, обеспечивающих защиту от несанкционированного доступа к изменению параметров;
- контроль за целостностью замков и пломб;
- контроль за состоянием информационных и силовых кабелей;
- управление доступом к информации и сервисам, включая требования к разделению обязанностей и ресурсов;
- регистрация значительных событий в журнале для целей повседневного контроля или специальных расследований;
- проверка и обеспечение целостности данных на всех стадиях их обработки;
- защита данных от несанкционированного доступа;
- использования средств шифрования;
- резервное копирование критически важных данных;
- восстановление работы ИС после отказов;
- защита от внесения несанкционированных дополнений и изменений.

4.3.1.2 Поддержание надлежащего уровня обеспечения безопасности

Для поддержания надлежащего уровня обеспечения безопасности *Системы* должны проводиться перечисленные ниже мероприятия:

- поддержание связей с производителями используемого программного обеспечения для своевременного получения обновлений и получения ответов на вопросы, которые могут возникнуть в ходе эксплуатации *Системы*;
- отслеживание новостей, публикуемых в печатных и электронных изданиях, на предмет появления новых возможностей обхода системы безопасности объектов автоматизации;
- периодическое тестирование системы безопасности;
- изучение и внедрение новых технологий, повышающих безопасность *Системы*.