

ООО «Сириус Софт»
109363, город Москва, улица Щосейная, д.90, стр.101
ТЕЛ/ФАКС: +7 (499) 650-85-87, HTTP://SIRIUS-IS.RU
E-MAIL: INFO@SIRIUS-IS.RU, ОКАТО 45290582000
ОГРН 1167746225020, ИНН/КПП 7723434588/772301001

ПК «Сириус-Центр»

Описание функциональных характеристик



СОДЕРЖАНИЕ

1 Общие положения	3
2 Отличительные особенности ПК «Сириус-Центр»	5
3 Основные принципы функционирования ПК «Сириус-Центр».....	7
4 Функциональная структура ПК «Сириус-Центр»	11
5 Подсистема представления данных.....	14
6 Подсистема администрирования и разработки	19
7 Подсистема расчетов.....	21
8 Сервис поддержки SCADA	22
9 Решения по построению систем поддержки принятия диспетчерских решений на базе ПК «Сириус-Центр»	23
10 Особенности построения серверного и пользовательского ПО.....	24
10.1 Стекло технологий	25
10.2 Структура проектов	26

1 Общие положения

ПК «Сириус-Центр» - программный комплекс собственной разработки ООО «Сириус-Софт», использующий в своем ядре компоненты собственного производства и свободное ПО, создавался как отечественный комплекс для поддержки систем оперативно-диспетчерского управления и систем управления производственными процессами.

В настоящее время программный комплекс «Сириус-Центр» - это программная среда, предназначенная для создания комплексных информационно-управляющих систем, содержащих решения для специализированных отраслевых задач, расчетно-аналитических комплексов, систем прогнозирования, статистической обработки данных, систем моделирования и обучения, в том числе для сложных, территориально распределенных технологических объектов.

Основной задачей ПК «Сириус-Центр» является обеспечение прозрачной интеграции различных многоплатформенных слабосвязанных систем автоматизации, работающих на разных операционных системах (Windows, Solaris, Linux и т.д.) и имеющих свои узкоспециализированные базы данных нормативно справочной информации (НСИ), в единое информационное пространство (Рисунок 1) на базе объектной модели предприятия.

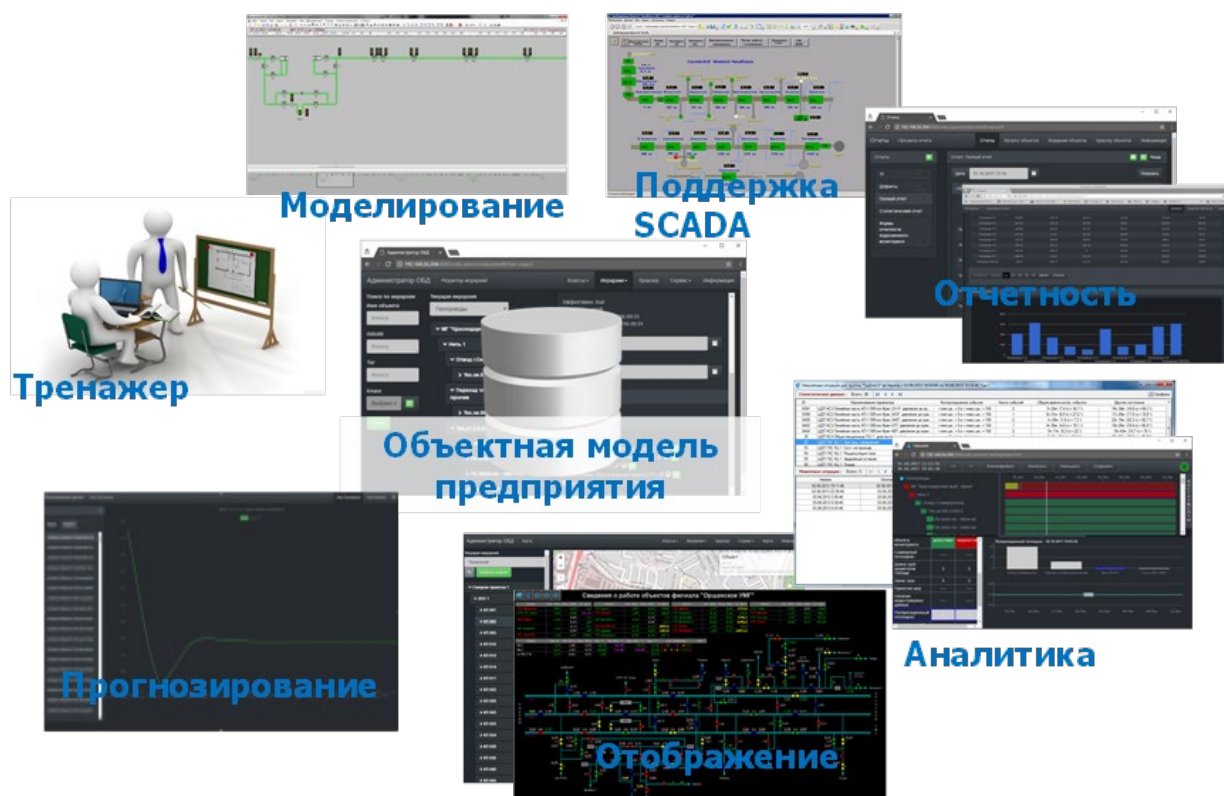


Рисунок 1. Единое информационное пространство

Основные функции, выполняемые комплексом за счет интеграции на базе единой объектной модели:

- Консолидация актуальной информации о состоянии технологических и производственных процессов в едином источнике данных;
- Создание и хранение аналитических срезов данных, режимных листов, суточных сводок;
- Построение интерфейса пользователя системы с любого рабочего места, подключенного к ЛВС предприятия;
- Предоставление информации в удобной для анализа и принятия решений форме, в том числе графическом и табличном виде;
- Проведение расчетов по запросу, расписанию или изменению;
- Формирование отчетов о работе системы;
- Обмен данными со смежными системами;
- Информационная поддержка смежных систем (актуализация объектных моделей НСИ).

Данный документ содержит описание основных функциональных характеристик ПК «Сириус-Центр», назначение и применение данного комплекса, а также особенности построения серверного и пользовательского ПО комплекса.

2 Отличительные особенности ПК «Сириус-Центр»

Основными отличительными особенностями ПК «Сириус-Центр» являются:

- Сквозной процесс разработки системы от этапа проектирования до внедрения и сопровождения на объекте;
- Администрирование и сопровождение проекта выполняется единым инструментальным набором;
- Унифицированное, масштабируемое решение, позволяющее строить вертикально и горизонтально интегрированные системы автоматизации сложных технологических объектов, в т.ч. иерархических и территориально распределенных предприятий;
- Возможно тиражирование путем настройки информационного обеспечения без адаптации программного кода компонент под однотипные предприятия;
- Максимальная открытость и доступность программного кода специалистам эксплуатационных служб для решения задач по сопровождению и функциональному развитию;

Применение ПК «Сириус-Центр» возможно в различных службах предприятия:

- Для диспетчерских служб – широкий набор возможностей по представлению и анализу информации, работа с одним и тем же объектом в разных подсистемах и в различных разрезах;
- Для административно-управленческого персонала – наличие единой точки ввода данных, автоматизация процесса создания и сопровождения проектов различных уровней, снижение затрат на сопровождение разнородных комплексов;
- Для руководства (топ менеджеров) – концентрация информации о технологическом процессе в одном месте, комплексная аналитика и отчеты, качественное решение поставленных бизнес задач, возможность развития и масштабирования системы.

Достоинства ПК «Сириус-Центр» при создании комплексных информационно-управляющих систем:

- Кроссплатформенное решение;
- Модульная архитектура;
- Дружественный интерфейс;
- Простота и доступность при обучении пользователей.
- Поддержка данных из различных источников, любой дискретности, в том числе данных реального времени;



- Универсальность системы в отношении ее применения в различных областях деятельности.
- Гибкий инструментарий для определения правил ведения проекта;
- Решение на базе тонкого клиента.

3 Основные принципы функционирования ПК «Сириус-Центр»

ПК «Сириус-Центр» строится на базе:

- объектной модели предприятия, содержащей метамодель и объектную модель конкретной предметной области;
- слоя API, обеспечивающий доступ к структурам данных;
- специализированных веб-серверов, реализующих предметно ориентированную бизнес-логику;
- тонкого клиента.

Архитектура ПК «Сириус-Центр» представлена ниже (Рисунок 2).

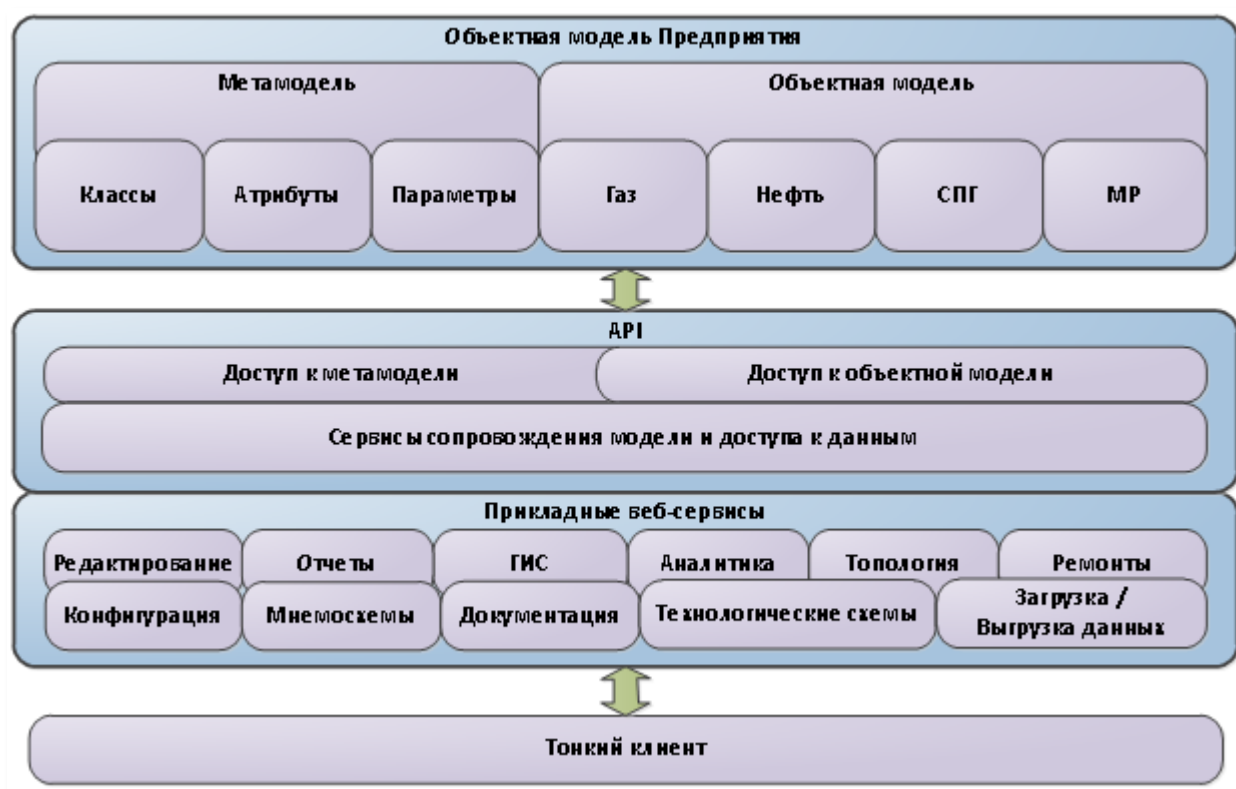


Рисунок 2. Архитектура ПК «Сириус-Центр»

В основе созданного комплекса лежит объектная модель данных, предназначенная для отображения предметной области автоматизируемых предприятий.

Особенностью модели является ее универсальность в части описания любого технологического объекта. Модель является конфигурируемой и расширяемой за счет использования слоя метаданных, что позволяет ее адаптировать под конкретную предметную область без внесения изменений в физическую модель данных или в код инструментальных программ по сопровождению данной модели. Слой метаданных (классы, атрибуты, параметры, иерархии) с одной стороны обеспечивает гибкость, с другой поддерживает механизмы ограничений и правил, сохраняет строгую

структуру информационного наполнения БД и предсказуемое поведение модели в целом.

В общем случае разработанную модель можно представить, как связанное между собой описание метаданных и объектной модели. Метаданные включают в себя следующие базовые сущности:

- Классы;
- Категории атрибутов;
- Категории параметров;
- Наследование классов;
- Иерархии;
- Правила иерархий;
- Категории;
- Аналитики и периоды.

В состав объектной модели для описания технологического объекта входят следующие базовые сущности:

- Объект;
- Каталог объектов;
- Атрибуты объекта;
- Параметры объекта;
- Иерархии объектов.

На базе разработанной модели создана методология цифрового отражения промышленного предприятия в объектную Базу Данных НСИ (БД НСИ), представленная ниже (Рисунок 3 - Рисунок 4). Методология состоит из 4 основных шагов:

1. Определение типовых объектов предметной области, формирование классов, включая возможный набор атрибутов и параметров данного класса, а также допустимые связи между классами;
2. Создание экземпляров классов, отражающих физический или логический объект предприятия;
3. Определение взаимосвязей объектов на базе атрибутов ссылок на объекты или иерархических структур, построенных по правилам, заданным на 1 шаге;
4. Определение состава активных (актуальных) параметров, получаемых от систем автоматизации или смежных систем, возможных аналитик и периодов, исключение параметров, унаследованных от класса.



Рисунок 3. Методология отражения предприятия

Для упрощения проведения первичного наполнения объектной модели могут быть разработаны дополнительные сервисные функции начального импорта структур данных, существующих на предприятии.

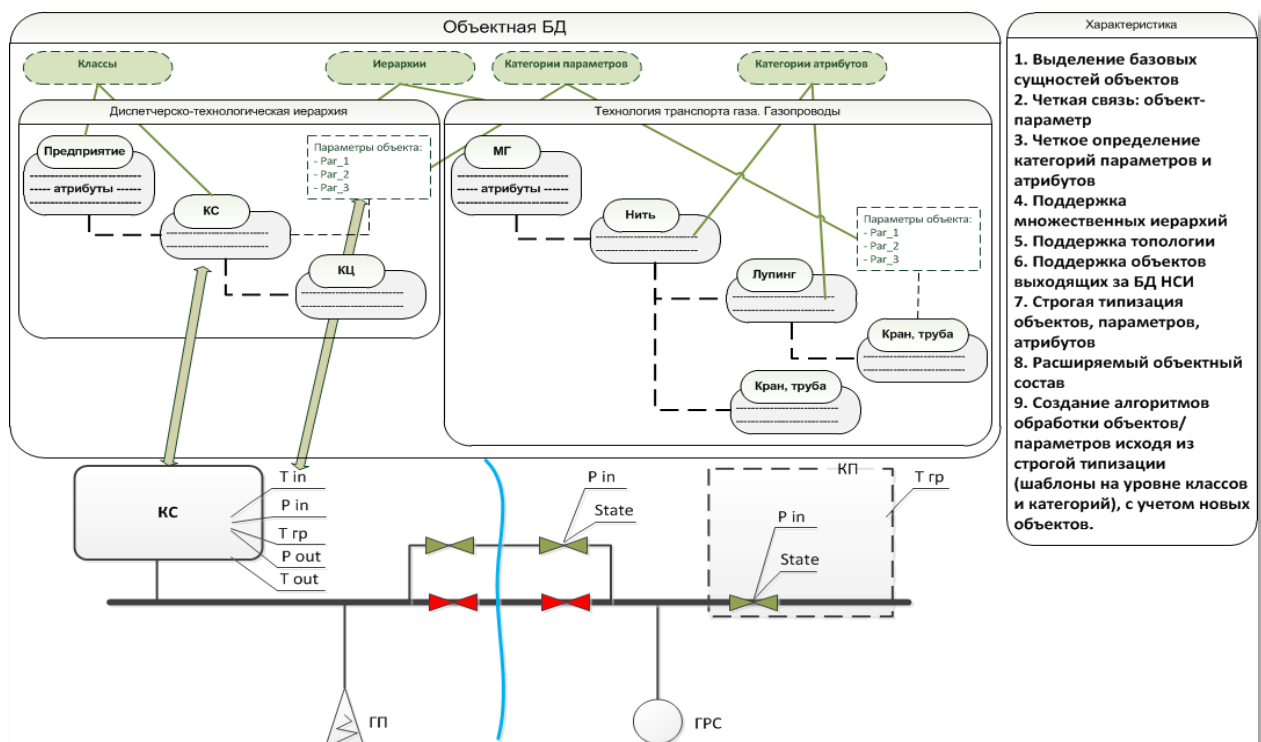


Рисунок 4. Методология отражения предприятия

В основу работы реализованной объектной модели положены следующие соглашения:

- Каждый объект является экземпляром одного и только одного класса;
- Объект создается единожды со всеми свойствами и ограничениями, наложенными на класс, от которого он порожден;
- Объект может находиться в логической или физической взаимосвязи с другими объектами, что поддерживается моделью при помощи иерархий и атрибутов ссылочного типа;
- В штатном режиме ПК «Сириус-Центр» не подразумевает возможность удаления данных, применяется вывод из эксплуатации объекта (сохранение сущности, но внесение признака о ее неактуальности). Данный механизм сохраняется также при работе с атрибутами и иерархическими связями, при этом все исторические отношения сохраняются в БД с целью сохранения целостности при проведении ретроспективного анализа данных;
- При необходимости создания объектов с новыми свойствами, но сохранением поведения в соответствии с заданным классом необходимо создать дочерний класс по отношению к заданному (поддержка наследования);
- При необходимости расширения существующего класса объектов поддерживается каскадное согласованное расширение свойств (категорий параметров и атрибутов).

Для каждого проекта на основе объектной модели строится единое описание предметной области, которым пользуются все приложения, разрабатываемые для данного проекта, в том числе разрабатываемые сторонними организациями. Физически модель строится на базе PostgreSQL, работа с данными осуществляется на базе Java API.

Серверные модули системы взаимодействуют через данное API. К модулям логики относятся задачи, адаптеры для получения данных реального времени, серверные модули интерфейсов пользователя (Отчеты, АРМ и других приложений), модели и т.д.

На основе объектного описания могут быть построены системы контроля и управления, различные информационно-аналитические, расчетные, проектные, тренажерные и другие комплексы предприятия.

4 Функциональная структура ПК «Сириус-Центр»

ПК «Сириус-Центр» состоит из следующих основных функциональных подсистем:

- Объектная модель данных;
- Подсистема хранения;
- Подсистема Представления данных;
- Подсистема Расчетных задач;
- Подсистема Администрирования и разработки;
- Подсистема коммуникаций;
- Подсистема поддержки SCADA.

Функциональная структура показана ниже (Рисунок 5).

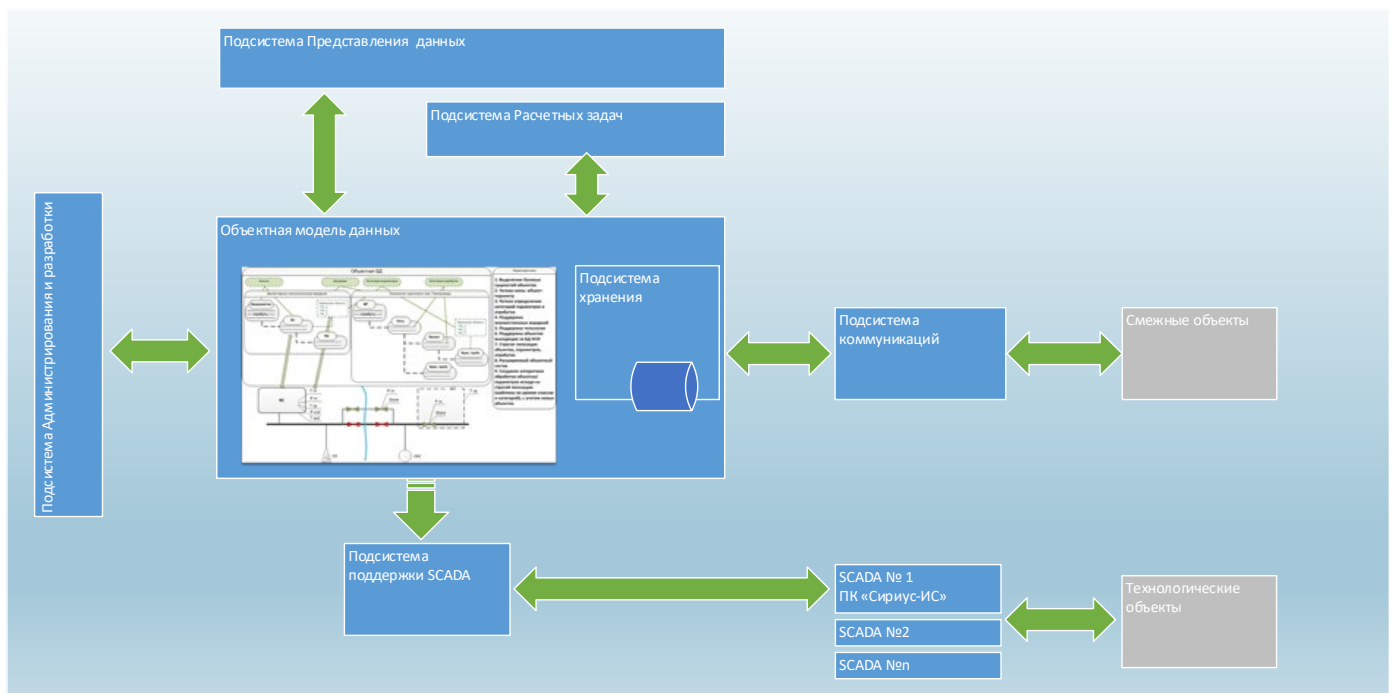


Рисунок 5. Функциональная структура ПК «Сириус-Центр»

Объектная модель данных

Объектная модель данных, предназначена для отображения предметной области автоматизируемых предприятий, принцип работы приведен в разделе 3.

Подсистема хранения

Подсистема хранения предназначена для выполнения следующих функций:

- Ведение истории аналитических рядов;
- Быстрый доступ к данным на запись и чтение при большом объеме сохраняемых данных. Доступ к данным истории и журналов через стандартные SQL запросы или API.

Подсистема Представления данных

Подсистема Представления данных (интерактивные отчеты) предназначена для создания, редактирования и просмотра различных форм отображения аналитической информации в табличном и графическом виде. Позволяет формировать различные табличные представления сложной структуры и форматирования. Так же поддерживается возможность добавлять в представления различные виды диаграмм и графиков.

Элемент (таблица) представления может быть построен в автоматическом режиме, например, на базе определенной иерархии. За счет того, что подсистема построена на базе объектной НСИ, пользователь получает возможность создавать новые и изменять существующие формы представления данных в режиме «онлайн», непосредственно в интерфейсе приложения, не прибегая к программированию и дополнительным средам разработки.

Подсистема Расчетных задач

Подсистема расчетных задач представляет собой комплекс, предназначенный для построения систем мониторинга, поддержки принятия решений и др. с гибко настраиваемым и расширяемым набором модулей, выполняющих аналитические и расчетные задачи.

Основным принципом работы подсистемы являются, единые требования к разработке расчетных модулей, обработке, администрированию, хранению и представлению данных для гибкой настройки, интеграции модулей, и агрегирования информации из различных систем.

Все прикладные задачи рассматриваются как функциональные модули, интегрированные в единое информационное пространство системы, построенное на базе объектной модели. Работа с данными посредством Java API. Подсистема реализует необходимый функционал, обеспечивает целостность данных, контроль доступа для пользователей и приложений. Прикладные задачи реализуются на базе JAVA API или на базе интегрированного интерпретатора JS.

Подсистема Администрирования и разработки

Подсистема администрирования и разработки обеспечивает выполнение функций удаленного сопровождения объектной модели, поддерживается работа как с метамоделью, так и с объектной моделью предприятия, ведения аудита выполняемых действий пользователя. Подсистема администрирования включает в себя сервисные средства сопровождения схемы БД, создания и обновления, копирования и переноса данных.

Подсистема коммуникаций

Коммуникации на основе обмена сообщениями (файлами) используются для передачи сеансовых данных, диспетчерских заданий и сообщений, обмена заявками (планами), отчетными и балансовыми показателями. Обмены осуществляются как в рамках одного предприятия, так и со «смежными» организациями. В отличие от обменов в реальном времени на основе промышленных протоколов передача сообщений может реализовываться по регламенту, с протоколированием шагов подготовки, передачи и получения информации, сопровождаться проверками достоверности и непротиворечивости полученных данных.

Для описания коммуникаций используются шаблоны, каждый из которых описывает обмен определенными данными по определенным регламентам по заданным каналам связи.

Отправка и обработка полученных файлов проводятся автоматически по событию, циклически или вручную по команде. Передаваемые и принимаемые сообщения могут быть в различных форматах – XML, ASCII, Excel и другие. Используется различная среда передачи данных – электронная почта, доступ к файловой системе, SFTP, протокам SOAP, AS2, AS4. Для обеспечения надежности обменов и резервирования для связи могут быть описаны альтернативные коммуникационные каналы. Осуществляется контроль получаемого и передаваемого сообщения по разным критериям, протоколируется ход приема/передачи.

Подсистема поддержки SCADA

Подсистема поддержки БД SCADA позволяет проводить генерацию конфигураций прикладного исполняемого проекта SCADA системы в автоматизированном режиме на основе объектного описания технологического объекта. В конфигурации могут входить БД нормативно-справочной информации (НСИ), БД мнемосхем, БД тревог, БД драйверов ввода/вывода, БД механизмов взаимодействия с внешними системами, БД расчетных задач.

5 Подсистема представления данных

Подсистема представления данных – это интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы.

В ПК «Сириус-Центр» применяется два вида интерфейса:

- Текстовый – командная строка;
- Графический (WIMP - интерфейс (Window - окно, Image - образ, Menu - меню, Pointer - указатель)).

Текстовый вид интерфейса используется специалистами при первоначальной настройке программно-технических средств и при проведении работ по обслуживанию.

В остальных случаях используется графический интерфейс на базе веб-технологий, позволяющий осуществлять просмотр информации из различных подсистем. Графический интерфейс делится на интерфейс пользователя и интерфейс администратора.

Подсистема представления данных служит для автоматизированного формирования отчётов на основе отчетных форм, которые создаются и настраиваются вручную. При обращении к заданному отчету автоматически формируются выборки данных с возможностью ручного ввода (выбора времени формирования) и исправления (значений заданных показателей при необходимости).

Представление данных возможно в табличном и графическом режиме, в виде диаграмм и трендов. Один отчёт может состоять из набора различных представлений. При помощи формы, представленной ниже (Рисунок 6), задаётся внешний вид отчёта и привязка его элементов к данным.



Рисунок 6. Внешний вид отчета

Табличное представление

Представляет из себя таблицу (Рисунок 7, Рисунок 8), в ячейки которой можно вставлять текст, числовые значения, формулы или привязывать значения параметров объектов на заданное время. В качестве метки времени можно использовать непосредственно отметку в часах или сдвиг по времени. Формула может использовать



арифметические выражения с константами, ссылками на другие ячейки таблицы и функции. Набор функций расширяем.

Ячейки таблиц можно объединять и управлять отображением их границ. Размер, цвет и стиль текста можно изменять.

Резервуарный парк Головной нефтеперекачивающей станции																			
	Резервуар 001-P-001			Резервуар 001-P-002			Резервуар 001-P-003			Резервуар 001-P-004			Резервуар 001-P-005			Резервуар 001-P-006			
	Т нефт, °C	Уровень нефт, мм	Уровень "нефть- вода", мм	Т нефт, °C	Уровень нефт, мм	Уровень "нефть- вода", мм	Т нефт, °C	Уровень нефт, мм	Уровень "нефть- вода", мм	Т нефт, °C	Уровень нефт, мм	Уровень "нефть- вода", мм	Т нефт, °C	Уровень нефт, мм	Уровень "нефть- вода", мм	Т нефт, °C	Уровень нефт, мм	Уровень "нефть- вода", мм	
30.05.2019 00:00	0.46	0.1	0.8	0.03	0.1	0.3	0.68	0.9	0	0.82	0.5	0.7	0.57	0.8	0.2	0.53	0.2	0.6	Клапан регулирующий Кг2 - Давление Время: 28.05.2019, 04:58:17 Значение: 0.999
30.05.2019 02:00	0.58	0.9	0	0.42	0	0.4	0.59	0.2	0.6	0.66	0.2	0.1	0.42	0.1	0.99	0.96	0.7	0.1	
30.05.2019 04:00	0.07	0.7	1	0.05	0.2	0	0.04	0.1	0.2	0.41	0.1	0.8	0.82	0.9	0.5	0.55	0.7	0.5	
30.05.2019 06:00	0.94	0.5	0.6	0.6	0.3	0.4	0.83	1	0.5	0.77	0.3	0.4	0.12	0.7	0.8	0.87	0.9	0.5	
30.05.2019 08:00	0.16	0.8	0.2	0.17	0.6	0.5	0.55	0.6	0.5	0.64	0.3	0.7	0.98	0.2	0.9	0.03	0.2	0.4	
30.05.2019 10:00	0.98	0.3	0.6	0.78	0.2	0.8	0.18	0.1	0.1	0.05	0.2	0.9	0.82	0.5	0.8	0.37	0.9	0.7	
30.05.2019 12:00	0.94	0.8	0.2	0.55	0	1	0.55	0.8	0.6	0.65	0.4	0.7	0.06	0.3	0.1	0.54	0.5	0.5	
30.05.2019 14:00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
30.05.2019 16:00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
30.05.2019 18:00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
30.05.2019 20:00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
30.05.2019 22:00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

Рисунок 7. Табличное представление

Отчет: СИКН УПН

Назад

Дата: 31.05.2019 09:55

Показать

СИКН УПН

	Расход	Давление, МПа	Температура, °C	Плотность, кг/м³
31.05.2019 00:00	0.46	0.25	0.8	0.35
31.05.2019 02:00	0.37	0.15	0.66	0.93
31.05.2019 04:00	0.99	0.26	0.67	0.96
31.05.2019 06:00	0.09	0.32	0.15	0.94
31.05.2019 08:00	0.34	0.99	0.38	0.19
31.05.2019 10:00	-	-	-	-
31.05.2019 12:00	-	-	-	-
31.05.2019 14:00	-	-	-	-
31.05.2019 16:00	-	-	-	-
31.05.2019 18:00	-	-	-	-
31.05.2019 20:00	-	-	-	-
31.05.2019 22:00	-	-	-	-

Рисунок 8. Табличное представление

Реализовано табличное представление двух видов: автоматическое и настраиваемое.

Форма автоматического табличного представления задаёт набор привязанных к параметрам колонок и критерии выбора объектов для отчёта. В сформированном отчёте для каждого объекта, соответствующего критериям выбора, создаётся строка, в каждой колонке которой помещается значение соответствующего параметра этого объекта. Такая форма имеет важное преимущество: при добавлении или удалении отображаемых объектов такой вид таблиц не требует корректировки.

Форма настраиваемого табличного представления не имеет такой степени автоматизации, но предоставляет гораздо больше возможностей для настройки макета. Данные и внешний вид каждой ячейки настраиваются отдельно.

Диаграмма

Диаграмма (Рисунок 9, Рисунок 10) представляет из себя графическое отображение дискретного ряда числовых значений с подписями к каждому значению. В качестве данных для отображения диаграмм используются столбцы таблиц отчёта. Для каждой диаграммы задаётся два столбца: из одного извлекаются значения, из другого - подписи. Поддерживаются два вида диаграмм: столбчатая и круговая.

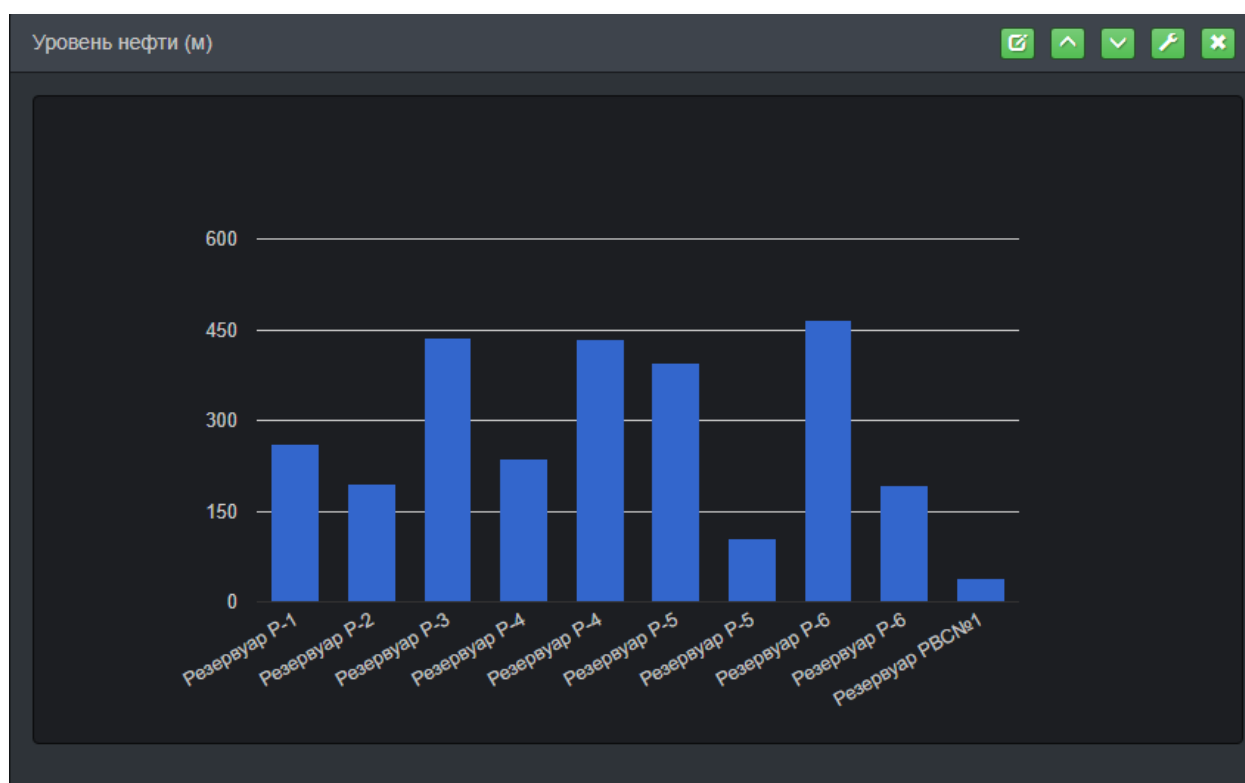


Рисунок 9. Гистограммы

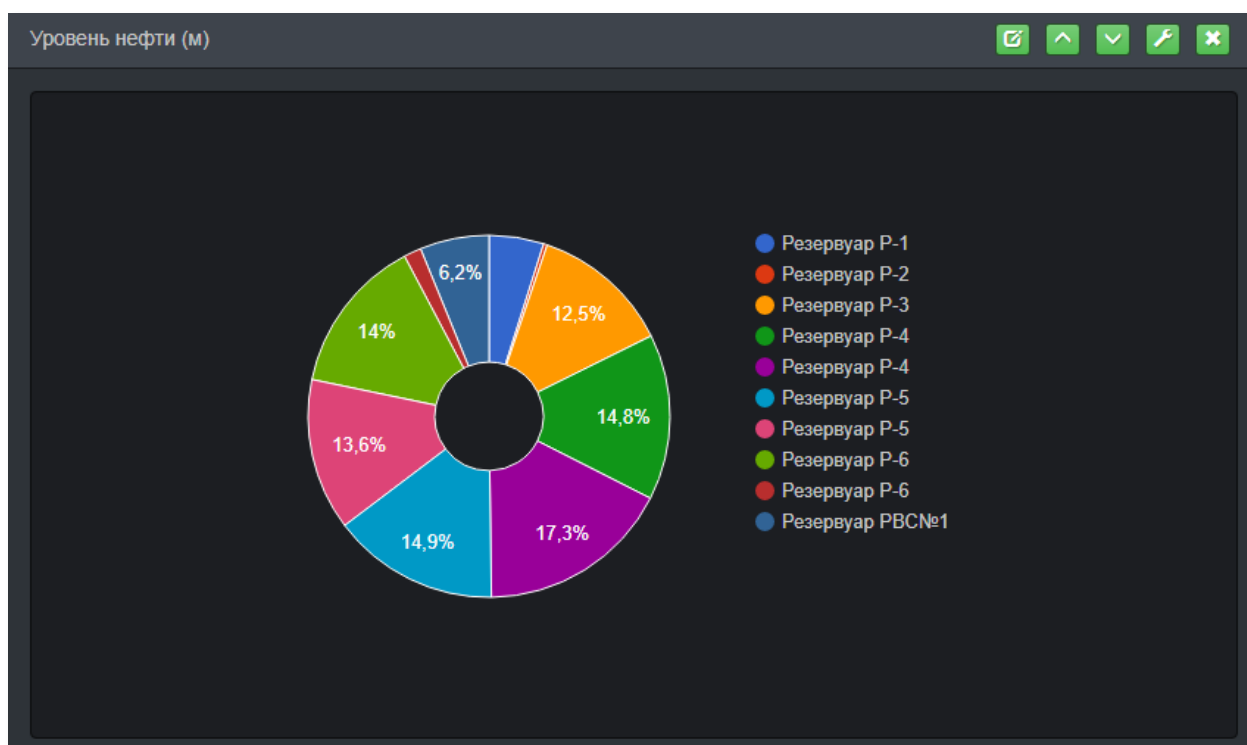


Рисунок 10. Круговые диаграммы

Тренд

Тренд (Рисунок 11) представляет из себя графическое отображение временного ряда числовых значений в течение заданного промежутка времени в виде линий на графике с уставками. В качестве данных используются значения заданного набора параметров объектов. Для каждого параметра объекта строится своя линия. Реализовано два вида линий: «ломаная» и «ступени». Каждой линии можно задать свой цвет.

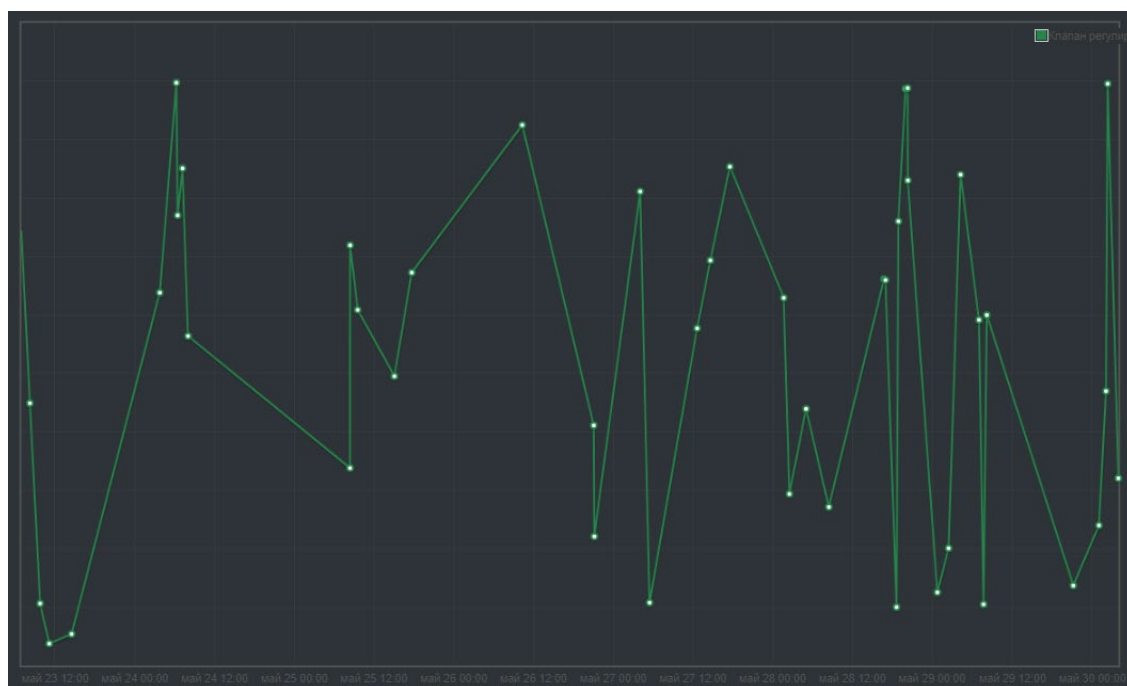


Рисунок 11. Тренд

6 Подсистема администрирования и разработки

Подсистема администрирования и разработки состоит из:

- Администратора;
- сервисного ПО.

Администратор – приложение, предоставляющее пользователю интерфейс для ведения объектной модели и работы со вспомогательным функционалом. В разделе «Метаданные» размещены формы работы с классами, категориями, правилами и другими метаданными. В разделе «Объектная модель» ведется работа с иерархиями, объектами, атрибутами и параметрами.

Рабочая область окна (Рисунок 12) состоит из навигационного меню, области навигации по объектам в виде списка или дерева и области редактирования свойств объектов.

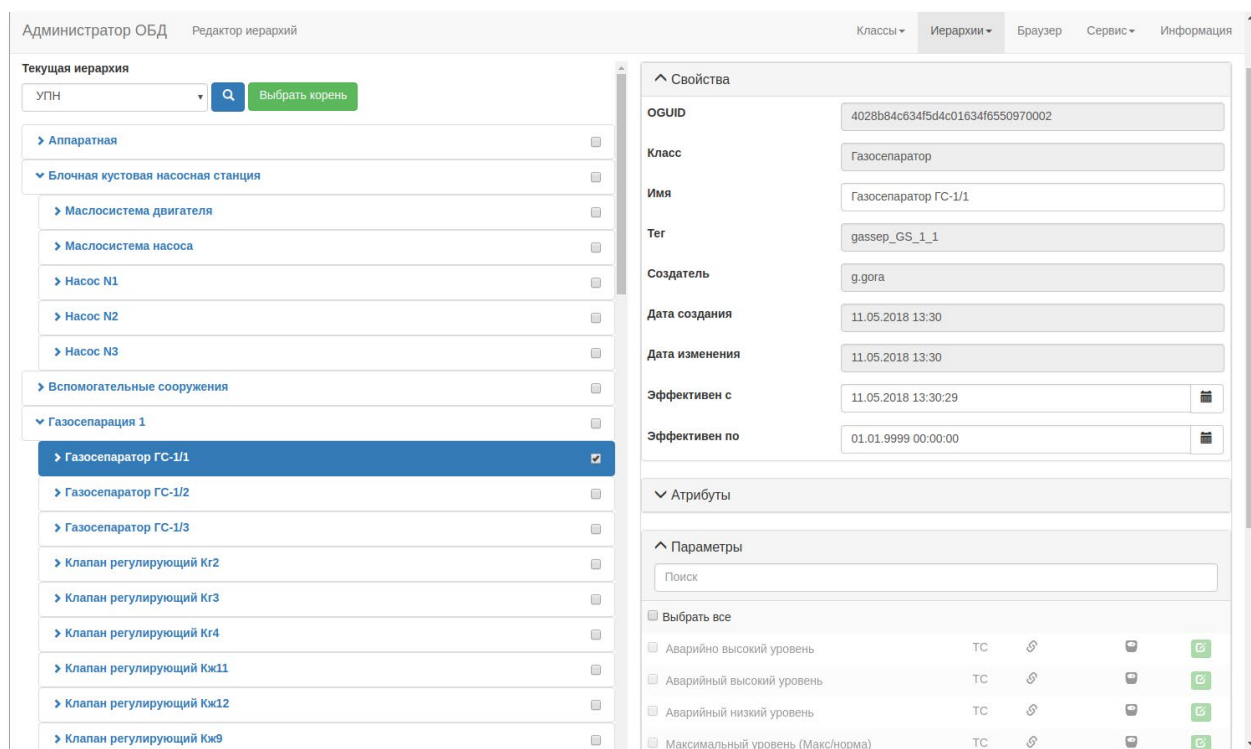


Рисунок 12. Администратор ПК «Сириус-Центр»

Сервисное ПО (tools) предназначено для выполнения сервисных функций по работе с ядром комплекса:

- Создание нового проекта и схемы ОБД;
- Обновления существующей схемы;
- Копирование классов - применяются для переноса общих метаданных при создании схожего с разрабатываемым ранее проектом. Включает в себя перенос классов с зависимыми категориями параметров и атрибутов.

Внешний вид окна Сервисного ПО показан ниже (Рисунок 13).

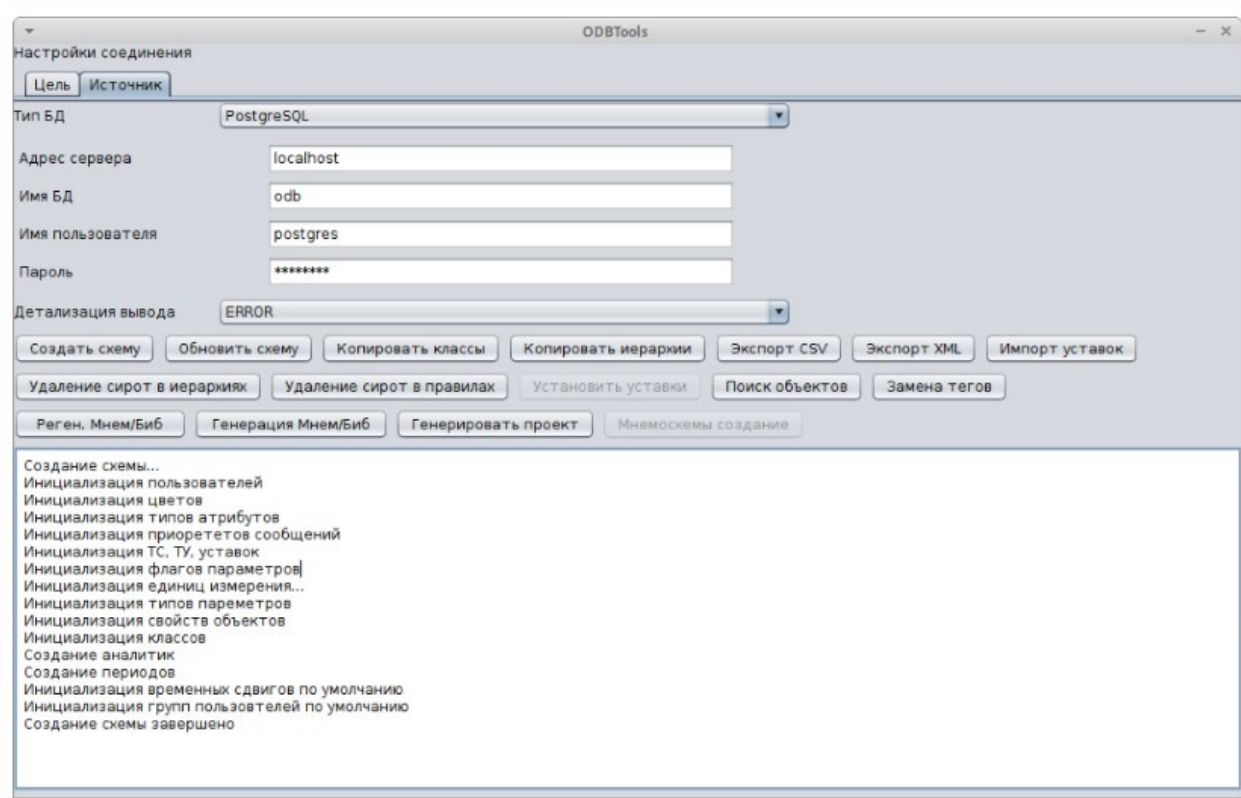


Рисунок 13. Основное окно Сервисного ПО

Создание и обновление схемы частично происходит автоматически средствами Hibernate. Hibernate создает таблицы и связи между их полями по соответствующему описанию ORM в коде. Все таблицы ОБД описаны ORM классами.

7 Подсистема расчетов

Подсистема расчетов - это отдельное приложение, на основе которого создается иерархия расчётных задач, для каждого алгоритма предоставляется его описание: объекты, параметры и атрибуты, с которыми он работает, а также расписание запуска.

Запуск расчёта осуществляется по времени или вручную. Модуль предоставляет информацию обо всех работающих в данный момент алгоритмах – время запуска, время выполнения, время завершения, результат выполнения.

В зависимости от описания алгоритма расчётный модуль осуществляет его выполнение двумя методами:

- Нативный код Java;
- Сценарии JavaScript.

Нативный код Java

Алгоритмы, написанные на Java, представляют собой расширение приложения. Представляются в виде классов, унаследованных от класса `AbstractAlgorithm`. Каждый созданный экземпляр алгоритма выполняется в отдельном потоке, который создаётся стандартными средствами Spring Framework. Ввиду сравнительной сложности этот подход рекомендуется использовать только для самых фундаментальных задач.

Сценарии JavaScript

Сценарии JavaScript хранятся в ОВД и в данный момент редактируются в приложении «Администратор». Выполняются интерпретатором расчётного приложения. Интерпретатор имеет доступ к ряду функций приложения и позволяет пользователю обращаться к ним напрямую из скрипта.

Создание скриптов осуществляется в специализированном редакторе атрибута типа скрипт в объектной БД.

8 Сервис поддержки SCADA

Сервис поддержки SCADA представлен на примере поддержки прикладных проектов ПК «Сириус-ИС». Сервис обеспечивает формирование следующих БД исполняемого проекта:

- БД НСИ;
- БД ОРС;
- БД Логики;
- БД Мнемосхемы.

В качестве иллюстрации приведен пример переноса одного объекта ОБД в адресное пространство исполняемого проекта ПК «Сириус-ИС» (Рисунок 14).

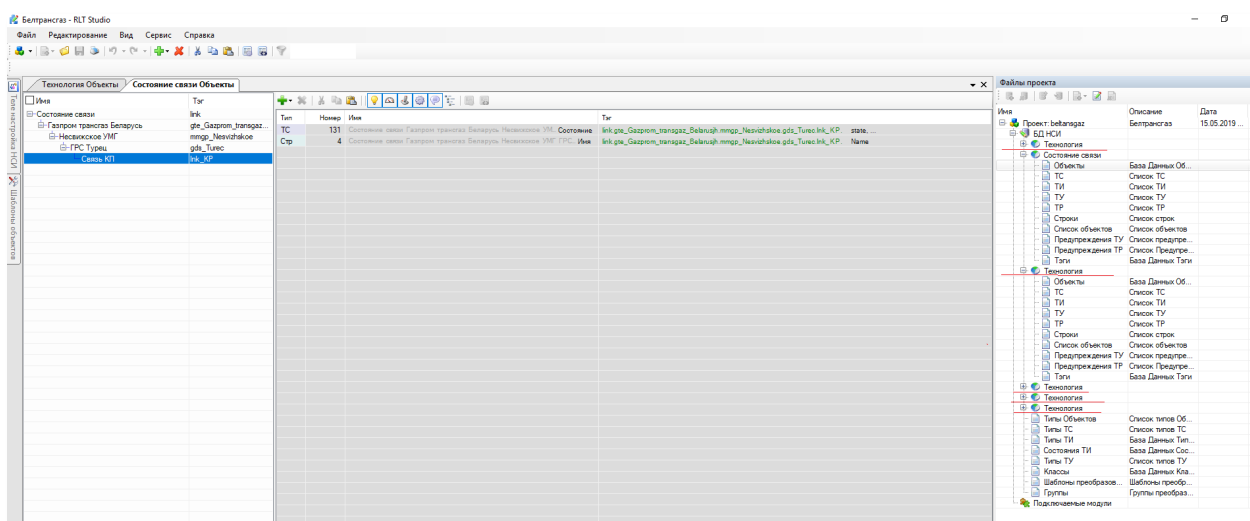


Рисунок 14. Отображение сконвертированной части проекта в RLTStudio

9 Решения по построению систем поддержки принятия диспетчерских решений на базе ПК «Сириус-Центр»

ПК «Сириус-Центр» обеспечивает информационную среду и средства ее сопровождения, позволяющие решать задачи уровня систем поддержки принятия решений. ООО «Сириус Софт» разрабатывает следующие программные комплексы и подсистемы, интегрированные с ПК «Сириус-Центр»:

- Комплекс моделирования процессов транспортировки газа на базе газодинамической модели;
- Подсистема обучения диспетчерского персонала на базе газодинамической модели;
- Система предиктивной аналитики;
- Система статистической обработки данных;
- Отчетные системы, интегрированные с объектной БД;
- Специализированные аналитические комплексы (ЭХЗ, КМСО).

Перечисленные решения в качестве источника данных используют единую объектную БД, тем самым достигается непротиворечивость исходных данных при решении различных технологических задач. Взаимодействие с ПК «Сириус-Центр» осуществляется посредством использования стандартного API.

10 Особенности построения серверного и пользовательского ПО

Реализованы следующие особенности построения серверного и пользовательского ПО:

- Функционал системы разбит на приложения, приложения разбиты на библиотеки, максимально используется код сторонних открытых библиотек. Применяются менеджеры пакетов;
- Минимизация зависимостей - приложения и модули разрабатываются таким образом, чтобы минимально зависеть от других приложений и модулей;
- Прозрачность - прозрачные программы для облегчения последующего пересмотра и отладки, применение общепринятых паттернов и методик, документирование кода;
- Единый интерфейс, API через который происходит взаимодействие с системой;
- Применение только открытых и кроссплатформенных решений.

Для каждого проекта на основе объектной БД строится единое описание предметной области, которым пользуются все приложения, разрабатываемые для данного проекта, в том числе разрабатываемые сторонними организациями. Физически ОВД строится на базе PostgreSQL, работа с данной ОВД осуществляется на базе Java API.

Серверные модули системы взаимодействуют через данное API. К модулям логики относятся задачи, адаптеры для получения данных реального времени, серверные модули интерфейсов пользователя (Отчеты, АРМ и других приложений), модели и т.д.

Серверные модули реализуют соответствующие Веб API для пользовательских интерфейсов (протоколы HTTP(S), WebSocket, STOMP), а также позволяет взаимодействовать со смежными системами, имеющими соответствующие интерфейсы.

10.1 Стек технологий

Ввиду разделения проектов на серверные и клиентские части имеется два соответствующих стека технологий.

Клиент

- AngularJS – JavaScript фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC-шаблона, а также упрощение тестирования и разработки;
- Bootstrap – оформление пользовательского интерфейса;
- Flot – графики, диаграммы, используется в формах представления данных;
- Grunt – сборка ПО;
- Bower – менеджер пакетов;
- Node.js – обеспечение работы Grunt и Bower.

Сервер

- Spring framework – отвечает за:
 - Реализацию принципа IoC (Inversion of Control) для слабой связанности кода. Каждый компонент системы должен быть как можно более изолированным от других, не полагаясь в своей работе на детали конкретной реализации других компонентов;
 - Управление транзакциями;
 - Авторизацию и аутентификацию;
 - Валидацию;
 - Web API;
- Hibernate – ORM, абстракция от конкретной СУБД;
- Postgres/H2DB/Oracle/MySql и др.СУБД;
- Maven – сборка, менеджер пакетов.

10.2 Структура проектов

Серверная часть реализуется на Java (поддерживается реализация OpenJDK) и разворачивается на серверах приложений Payara5, Glassfish, Tomcat. Клиентская часть - HTML и Java Script.

Клиентская часть представляет собой одностраничные приложения и загружаются полностью при первом открытии приложения в браузере, дальнейшее взаимодействие осуществляется только по данным посредством RESTful API. Для упрощения отладки сборщик позволяет запустить клиентскую часть на выполнение локально, обеспечивая проксирование запросов до сервера.

Для клиентских и серверных частей общий и базовый функционал вынесен в отдельную библиотеку (common). Эти библиотеки содержат логику (серверная часть) и интерфейсы пользователя (клиентская) по работе с сущностями объектной модели, например, с деревьями классов и объектов, правилами, категориями параметров и атрибутов, свойствами объектов, свойствами параметров и т.д. Кроме того, общие библиотеки имеют все базовые зависимости в менеджерах пакетов, таким образом, для сервера - это Spring и Hibernate, а для клиента - AngularJS и Bootstrap.

Взаимодействие клиентской и серверной части происходит посредством HTTP RESTful API. Это единый интерфейс для взаимодействия с системой. API документируется в коде серверной части. Документированию подлежат функции контроллеров, их параметры, а также объекты классов DTO (классы передачи данных между подсистемами приложения). Для разработчиков создан проект Веб - портал с автоматически создаваемой документацией API на базе библиотеки Swagger.