


Модуль подсистемы “Пользовательские интерфейсы” <Vision>

Модуль:	Vision
Имя:	Рабочий пользовательский интерфейс (QT)
Тип:	Пользовательские интерфейсы
Источник:	ui_Vision.so
Версия:	0.8.0
Автор:	Роман Савоченко
Разработчик:	Роман Савоченко, Максим Лысенко, Ксения Яшина
Описание:	Визуальный рабочий пользовательский интерфейс.
Лицензия:	GPL

Оглавление

Модуль подсистемы “Пользовательские интерфейсы” <Vision>.....	1
Введение.....	1
1 Назначение.....	2
2 Инструмент графического формирования интерфейса СВУ.....	4
3 Исполнение интерфейсов СВУ.....	14
4 Представление базовых элементов (примитивов).....	16
4.1 Примитив элементарная фигура (ElFigure).....	17
4.2 Примитив текста (Text).....	18
4.3 Примитив элементов формы (FormEl).....	19
4.4 Примитив отображения медиа-материалов (Media).....	20
4.5 Примитив построения диаграм/графиков (Diagram).....	21
4.6 Примитив формирования протокола (Protocol).....	22
4.7 Примитив формирования отчётной документации (Document).....	23
4.8 Примитив контейнера (Box).....	24
5 Общая конфигурация модуля.....	25
Заключение.....	25

Введение

Модуль Vision предоставляет механизм конечной визуализации среды визуализации и управления (СВУ) в систему OpenSCADA. Модуль основан на многоплатформенной библиотеке графического пользовательского интерфейса (GUI) фирмы TrollTech – QT( <http://www.trolltech.com/qt/>). В своей работе модуль использует данные движка СВУ (модуль [VCAEngine](#)).

Среда визуализации и управления (СВУ) является неотъемлемой составляющей SCADA системы. Она применяется на клиентских станциях с целью доступного предоставления информации об объекте управления и выдачи управляющих воздействий на объект. В различных практических случаях и условиях могут применяться СВУ, построенные на различных принципах визуализации. Например, это могут быть библиотеки виджетов QT, GTK+, wxWidgets или гипертекстовые механизмы на основе технологий HTML, XHTML, XML, CSS и JavaScript или же сторонние приложения визуализации, реализованные на различных языках программирования Java, Python и т.д. Любой из этих принципов имеет свои преимущества и недостатки, комбинация которых может стать непреодолимым

препятствием в возможности использования СВУ в том или ином практическом случае. Например, технологии вроде библиотеки QT позволяют создавать высокореактивные СВУ, что несомненно важно для станций оператора управления технологическим процессом (ТП). Однако необходимость инсталляции данного клиентского ПО в отдельных ситуациях может сделать использование его невозможным. С другой стороны Web-технологии не требуют инсталляции на клиентские системы и являются предельно многоплатформенными (достаточно создать ссылку на Web-сервер в любом Web-браузере), что наиболее важно для различных инженерных и административных станций. Но реактивность и надёжность таких интерфейсов ниже, что практически исключает их использования на станциях оператора ТП.

Система OpenSCADA имеет гибкую архитектуру, которая позволяет создавать внешние интерфейсы, в том числе и пользовательские, на любой основе и на любой вкус. Например, среда конфигурации системы OpenSCADA доступна как на QT библиотеке, так и на Web-основе.

В тоже время независимое создание реализаций СВУ на различной основе может повлечь за собой невозможность использования данных конфигурации одной СВУ в другой, что неудобно и ограничено с пользовательской стороны, а также накладно в плане реализации и последующей поддержки. С целью избежать этих проблем, а также создать в кратчайшие сроки полный спектр различных типов СВУ основан [проект создания концепции СВУ](#). Результатом этого проекта и стал данный модуль непосредственной визуализации (на основе библиотеки QT), модуль непосредственной визуализации [WebVision](#) и движок СВУ [VCAEngine](#).

1 Назначение

Данный модуль непосредственной визуализации СВУ предназначен для формирования и исполнения интерфейсов СВУ в среде графической библиотеки QT.

Финальная версия этого модуля СВУ, построенная на основе данного модуля, обеспечит:

- три уровня сложности в формировании интерфейса визуализации, позволяющие органично осваивать и применять инструментарий по методике от простого к сложному:
 - формирование из шаблонных кадров путём назначения динамики (без графической конфигурации);
 - графическое формирование новых кадров путём использования готовых элементов визуализации из библиотеки (мнемосхемы);
 - формирование новых кадров, шаблонных кадров и элементов отображение в библиотеки.
- построение интерфейсов визуализации различной сложности, начиная от простых плоских интерфейсов мониторинга и заканчивая полноценными иерархическими интерфейсами, используемыми в SCADA системах;
- предоставление различных способов формирования и конфигурации пользовательского интерфейса, основанных на различных интерфейсах графического представления (QT, Web, Java ...) или же посредством стандартного интерфейса управления системой OpenSCADA;
- смену динамики в процессе исполнения;
- построение новых шаблонных кадров на уровне пользователя и формирование специализированных под область применения библиотек кадров (например, включение кадров параметров, графиков и других элементов с увязкой их друг с другом), в соответствии с теорией повторного использования и накопления;
- построение новых пользовательских элементов визуализации и формирование специализированных под область применения библиотек кадров в соответствии с теорией повторного использования и накопления;
- описание логики новых шаблонных кадров и пользовательских элементов визуализации как простыми связями, так и лаконичным, полноценным языком пользовательского программирования;
- возможность включения в пользовательские элементы визуализации функций (или кадров вычисления функций) объектной модели OpenSCADA, практически связывая

представление с алгоритмом вычисления (например, визуализируя библиотеку моделей аппаратов ТП для последующего визуального построения моделей ТП);

- разделение данных пользовательских интерфейсов и интерфейсов представления этих данных, позволяющее строить интерфейс пользователя в одной среде, а исполнять во многих других (QT, Web, Java ...);
- возможность подключения к исполняющемуся интерфейсу для наблюдения и коррекции действий (например, при обучении операторов и контроля в реальном времени за его действиями);
- визуальное построение различных схем с наложением логических связей и последующем централизованным исполнением в фоне (визуальное построение и исполнение математических моделей, логических схем, релейных схем и иных процедур);
- предоставление функций объектного API в систему OpenSCADA, может использоваться для управления свойствами интерфейса визуализации из пользовательских процедур;
- построение серверов кадров, элементов визуализации и проектов интерфейсов визуализации с возможностью обслуживания множественных клиентских соединений;
- простая организация клиентских станций на различной основе (QT, Web, Java ...) с подключением к центральному серверу;
- полноценный механизм разделения полномочий между пользователями, позволяющий создавать и исполнять проекты с различными правами доступа к его компонентам;
- гибкое формирование правил сигнализаций и уведомления, с учётом и поддержкой различных способов уведомления;
- поддержка пользовательского формирования палитры и шрифтовых предпочтений для интерфейса визуализации;
- поддержка пользовательского формирования карт событий под различное оборудование управления и пользовательские предпочтения;
- поддержка профилей пользователей, позволяющая определять различные свойства интерфейса визуализации (цветовая гамма, шрифтовые особенности, предпочтительные карты событий);
- гибкое хранение и распространение библиотек виджетов, кадров и проектов интерфейсов визуализации в БД, поддерживаемых системой OpenSCADA; практически пользователю нужно только зарегистрировать полученную БД с данными.

2 Инструмент графического формирования интерфейса СВУ

Разработка интерфейса СВУ выполняется в одном окне, реализующем многодокументный интерфейс (MDI) интерфейс (рис. 1). Данный подход позволяет одновременно редактировать несколько кадров различных размеров. Используются следующие механизмы управления разработкой: панели инструментов, пункты меню и контекстное меню. Большинство действий дублируются в разных механизмах, что позволяет быстро найти инструмент предпочитаемым способом. Навигационные интерфейсы реализованы присоединяемыми окнами. Конфигурация панелей инструментов и присоединяемых окон сохраняется при выходе и восстанавливается при старте, что позволяет настраивать интерфейс под себя.

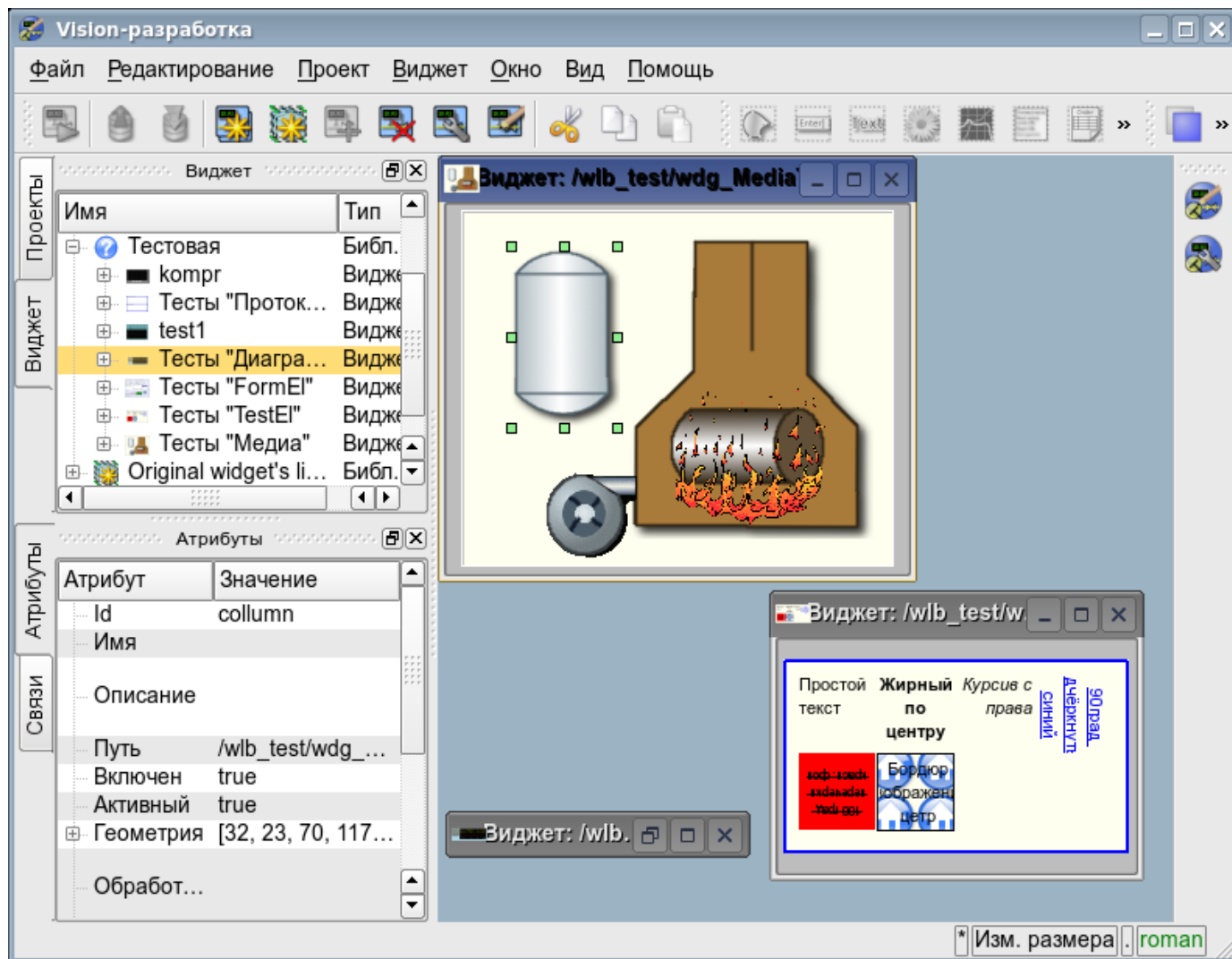


Рис.1. Окно разработки интерфейсов СВУ.

Доступ к основным компонентам СВУ производится посредством пришвартованных окон, на рис.1 эти окна изображены слева. В этих окнах содержатся:

- Дерево библиотек виджетов. С помощью навигатора можно быстро найти нужный виджет или библиотеку и проделать над ними необходимые операции. Реализованы операции: добавления, удаления, копирования, настройки виджетов и библиотек, а также визуального редактирования виджетов. Для гибкого управления поддерживается контекстное меню в составе пунктов:
 - «Новая библиотека» — создание новой библиотеки.
 - «Добавить визуальный элемент» — добавление визуального элемента в библиотеку.

- «Удалить визуальный элемент» — удаление визуального элемента из библиотеки.
- «Свойства визуального элемента» — конфигурация визуального элемента.
- «Редактировать визуальный элемент» — визуальное редактирование элемента.
- «Вырезать визуальный элемент» – вырезание/перемещение визуального элемента, в момент вставки.
- «Копировать визуальный элемент» – копирование визуального элемента, в момент вставки.
- «Вставить визуальный элемент» – вставка визуального элемента.
- «Загрузить из БД» — загрузка данных визуального элемента из БД.
- «Сохранить в БД» — сохранение данных визуального элемента в БД.
- «Обновить библиотеки» — выполняет перечитывание конфигурации и состава библиотек из модели данных.
- Дерево страниц проектов. Предусматривает механизм «Перетащи и брось» для формирования пользовательских кадров на основе элементов библиотек. Для гибкого управления поддерживает контекстное меню в составе пунктов:
 - «Запустить исполнение проекта» — запуск исполнения выбранного проекта.
 - «Новый проект» — создание нового проекта.
 - «Добавить визуальный элемент» — добавление визуального элемента в проект/страницу.
 - «Удалить визуальный элемент» — удаление визуального элемента из проекта/страницы.
 - «Свойства визуального элемента» — Конфигурация визуального элемента.
 - «Редактировать визуальный элемент» — визуальное редактирование элемента.
 - «Вырезать визуальный элемент» – вырезание/перемещение визуального элемента, в момент вставки.
 - «Копировать визуальный элемент» – копирование визуального элемента, в момент вставки.
 - «Вставить визуальный элемент» – вставка визуального элемента.
 - «Загрузить из БД» — загрузка данных визуального элемента из БД.
 - «Сохранить в БД» — сохранение данных визуального элемента в БД.
 - «Обновить проекты» — выполняет перечитывание конфигурации и состава проектов из модели данных.
- атрибуты виджетов;
- внешние связи виджетов.

В основном пространстве рабочего окна размещаются окна страниц проектов, кадров библиотек виджетов, пользовательских элементов и элементов примитивов на момент их визуального редактирования.

Сверху рабочего окна содержится меню. В меню размещены все инструменты, необходимые для разработки интерфейсов СВУ. Меню имеет следующую структуру:

- «Файл» — Общие операции.
 - «Загрузить из БД» — загрузка данных визуального элемента из БД.
 - «Сохранить в БД» — сохранение данных визуального элемента в БД.
 - «Закрыть» — закрыть окно редактора.
 - «Выход» — Выход из системы OpenSCADA.
- «Редактирование» — Операции редактирования визуальных элементов.
 - «Вырезать визуальный элемент» – вырезание/перемещение визуального элемента, в момент вставки.
 - «Копировать визуальный элемент» – копирование визуального элемента, в момент вставки.
 - «Вставить визуальный элемент» – вставка визуального элемента.
- «Проект» — Операции над проектами.
 - «Запустить исполнение проекта» — запуск исполнения выбранного проекта.

- «Новый проект» — создание нового проекта.
- «Добавить визуальный элемент» — добавление визуального элемента в проект.
- «Удалить визуальный элемент» — удаление визуального элемента из проекта.
- «Свойства визуального элемента» — конфигурация визуального элемента.
- «Редактировать визуальный элемент» — визуальное редактирование элемента.
- «Виджет» — Операции над виджетами и библиотеками виджетов.
 - «Новая библиотека» — создание новой библиотеки.
 - «Добавить визуальный элемент» — добавление визуального элемента в библиотеку.
 - «Удалить визуальный элемент» — удаление визуального элемента из библиотеки.
 - «Свойства визуального элемента» — конфигурация визуального элемента.
 - «Редактировать визуальный элемент» — визуальное редактирование элемента.
 - «Вид» — управление расположением визуальных элементов на кадрах.
 - «Виджет на верх» — поднятие виджета на самый верх.
 - «Виджет на низ» — опускание виджета на самый низ.
 - «Поднять виджет» — поднять виджет выше.
 - «Опустить виджет» — опустить виджет ниже.
 - «Выровнять слева» — выравнивание виджета слева.
 - «Выровнять в центре вертикально» — выравнивание виджета вертикально в центре.
 - «Выровнять справа» — выравнивание виджета справа.
 - «Выровнять сверху» — выравнивание виджета сверху.
 - «Выровнять в центре горизонтально» — выравнивание виджета горизонтально в центре.
 - «Выровнять снизу» — выравнивание виджета снизу.
 - «Библиотека: {Имя библиотеки}» — пункты меню для доступа к кадрам/виджетам содержащимся в библиотеке.
- «Окно» — Управление окнами MDI-интерфейса.
 - «Закрыть» — закрыть активное окно.
 - «Закрыть все» — закрыть все окна.
 - «Уложить» — уложить все окна для видимости одновременно.
 - «Каскадировать» — расположить все окна каскадом.
 - «Следующий» — активировать следующее окно.
 - «Предыдущий» — активировать предыдущее окно.
 - «Виджет: {Имя виджета}» — пункты активации конкретного окна.
- «Вид» — Управление видимостью рабочего окна и панелей на нём.
 - «Панель визуальных элементов» — панель управления визуальными элементами.
 - «Функции видимости виджетов» — панель управления видимостью и расположением виджетов на панелях.
 - «Панель элементарных фигур» — дополнительная панель редактирования примитива элементарных фигур («ElFigure»).
 - «Проекты» — пришвартованное окно управления деревом проектов.
 - «Виджеты» — пришвартованное окно управления деревом библиотек виджетов.
 - «Атрибуты» — пришвартованное окно диспетчера атрибутов.
 - «Связи» — пришвартованное окно диспетчера связей.
 - «Библиотека: {Имя библиотеки}» — управление видимостью панелей библиотек виджетов.
- «Помощь» — Помощь по OpenSCADA и модулю Vision.
 - «Про» — информация про данный модуль.
 - «Про QT» — информация о библиотеке QT, используемой модулем.
 - «Что это» — запрос описания элементов интерфейса окна.

Сверху, под меню, или по сторонам, располагаются панели инструментов. Панели могут быть скрыты или расположены, что управляется в пункте меню «Вид». Существуют следующие панели

инструментов:

- «Панель визуальных элементов» — Панель управления визуальными элементами:
 - «Запуск исполнения проекта для выбранного элемента» — производит запуск проекта на исполнение и активацию выбранной страницы проекта.
 - «Загрузить данные элемента из БД» — выполняет загрузку данных выбранного(ых) элементов из БД.
 - «Сохранить данные элемента в БД» — выполняет сохранение данных выбранного(ых) элементов в БД.
 - «Новый проект» — создание нового проекта.
 - «Новая библиотека» — создание новой библиотеки.
 - «Добавить визуальный элемент» — добавление визуального элемента в проект.
 - «Удалить визуальный элемент» — удаление визуального элемента из проекта.
 - «Свойства визуального элемента» — конфигурация визуального элемента.
 - «Редактировать визуальный элемент» — визуальное редактирование элемента.
 - «Вырезать визуальный элемент» – вырезание/перемещение визуального элемента, в момент вставки.
 - «Копировать визуальный элемент» – копирование визуального элемента, в момент вставки.
 - «Вставить визуальный элемент» – вставка визуального элемента.
- «Функции видимости виджетов» — Панель управления видимостью и расположением виджетов на панелях:
 - «Виджет на верх» — поднятие виджета на самый верх.
 - «Виджет на низ» — опускание виджета на самый низ.
 - «Поднять виджет» — поднять виджет выше.
 - «Опустить виджет» — опустить виджет ниже.
 - «Выравнять слева» — выравнивание виджета слева.
 - «Выровнять в центре вертикально» — выравнивание виджета вертикально в центре.
 - «Выровнять справа» — выравнивание виджета справа.
 - «Выровнять сверху» — выравнивание виджета сверху.
 - «Выровнять в центре горизонтально» — выравнивание виджета горизонтально в центре.
 - «Выровнять снизу» — выравнивание виджета снизу.
- «Панель элементарных фигур» — Дополнительная панель редактирования примитива элементарных фигур («ElFig»).
 - «Возврат курсора» — возврат к курсору для действий над фигурами на виджете.
 - «Добавить линию» — добавление линии к элементарной фигуре.
 - «Добавить дугу» — добавление дуги к элементарной фигуре.
 - «Добавить кривую безье» — добавление кривой безье к элементарной фигуре.
 - «Включить привязки» — включение привязок у элементарной фигуры.
- «Библиотека: {Имя библиотеки}» — Управление видимостью панелей библиотек виджетов. Содержимое панели зависит от содержимого библиотеки и включает кнопки вызова элементов библиотеки.
- «QTStarter панель инструментов» — Панель, созданная модулем запуска модулей библиотеки QT. Содержит кнопки для запуска UI модулей системы OpenSCADA, основанных на библиотеке QT. С помощью этой панели можно открыть несколько экземпляров окон данного модуля или других модулей.

Внизу окна разработки СВУ располагается строка статуса. С правой стороны строки статуса размещаются индикаторы визуального масштаба редактируемого кадра, режима изменения размера элементов, режима текущей станции движка СВУ и пользователя, от имени которого ведётся разработка интерфейса СВУ. По двойному клику на индикаторе пользователя можно пользователя сменить, введя новое имя и пароль пользователя. В главное поле строки статуса выводятся различные информационные сообщения и сообщения помощи.

Для редактирования свойств визуальных элементов предусмотрено два диалога. Первый диалог позволяет редактировать свойства контейнеров визуальных элементов (библиотек виджетов и проектов), рис. 2. Второй диалог для редактирования свойств самих визуальных элементов, рис. 3. Изменения, внесённые в диалогах, сразу-же попадают в движок СВУ. Для сохранения этих изменения в БД или восстановления из БД необходимо воспользоваться соответствующими инструментами главного окна разработки.

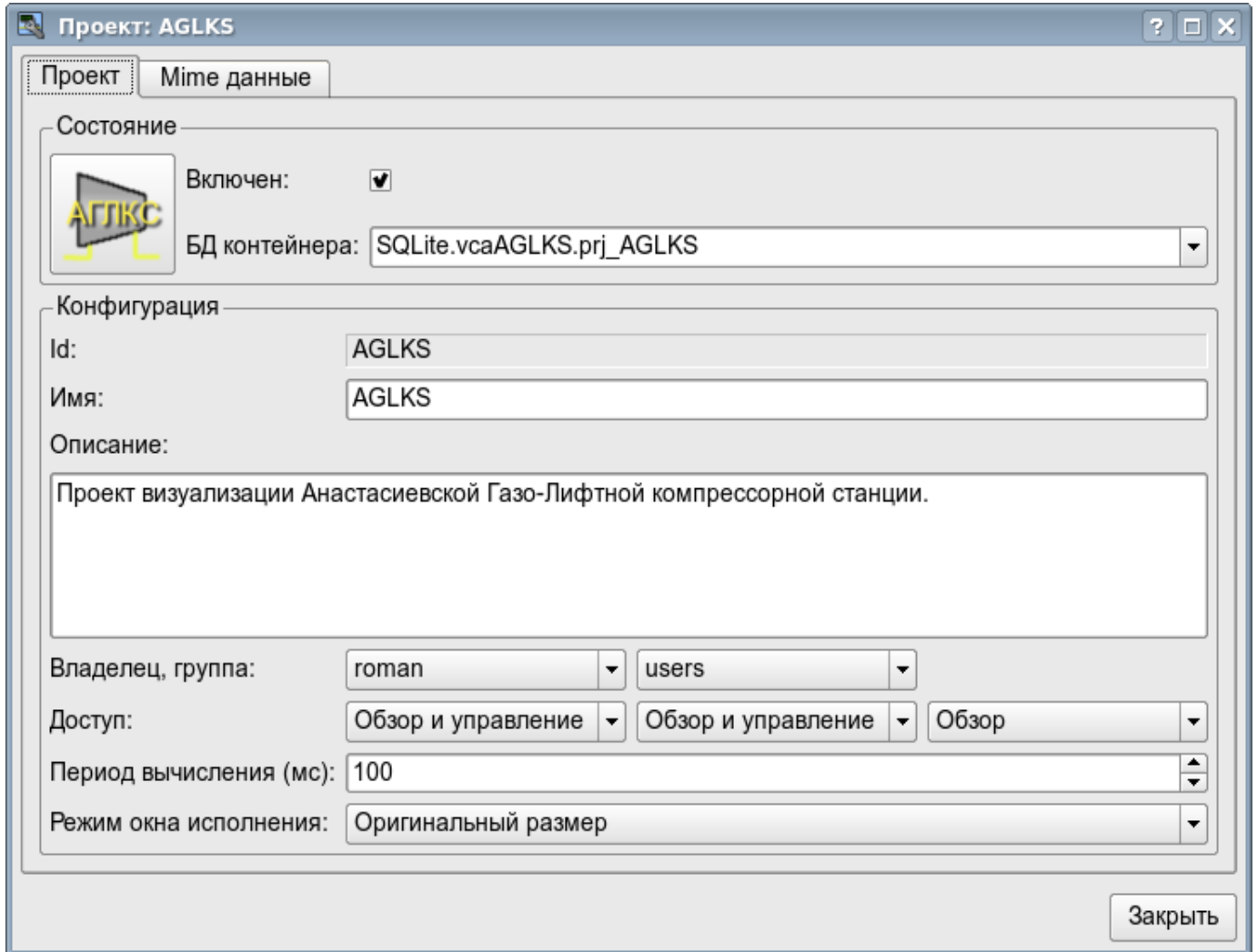


Рис.2. Диалог редактирования свойств контейнеров визуальных элементов.

С помощью главной вкладки этого диалога можно установить:

- Состояние контейнера элементов, а именно: «Включен», БД контейнера.
- Идентификатор, имя и описание контейнера.
- Для проекта: пользователя, группу пользователей и доступ пользователя, группы пользователей и всех остальных.
- Для проекта: период вычисления проекта и режим открытия окна при исполнении.

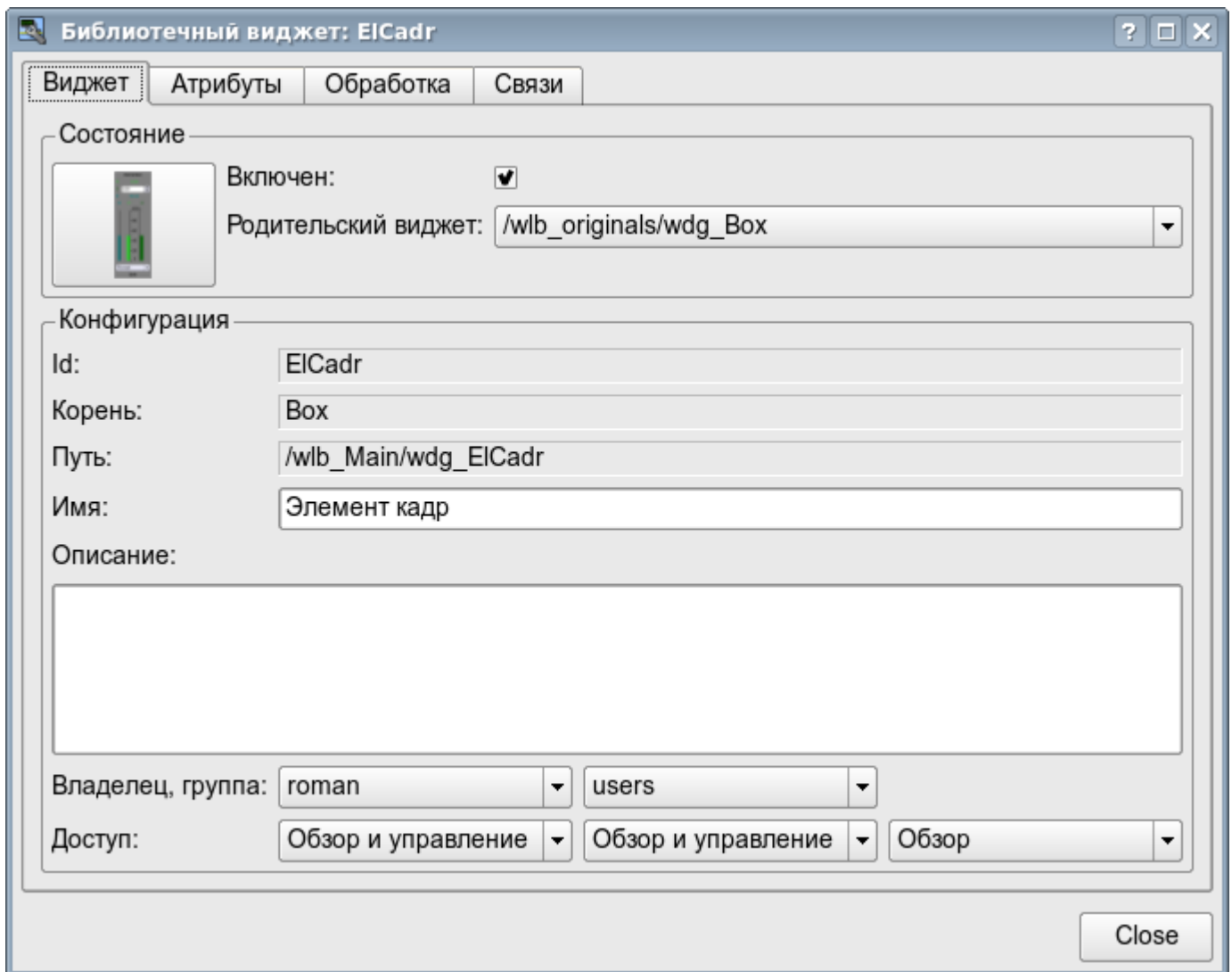


Рис.3. Диалог редактирования свойств визуальных элементов.

С помощью главной вкладки этого диалога можно установить:

- Состояние элемента, а именно: «Включен», родительский виджет.
- Идентификатор, корень, путь, имя и описание элемента.
- Пользователя, группу пользователей элемента и доступ пользователя, группы пользователей и всех остальных.

Диалог редактирования свойств контейнеров визуальных элементов содержит две вкладки: вкладку конфигурации основных параметров (рис.2) и вкладку конфигурации time-данных контейнеров (рис.4).

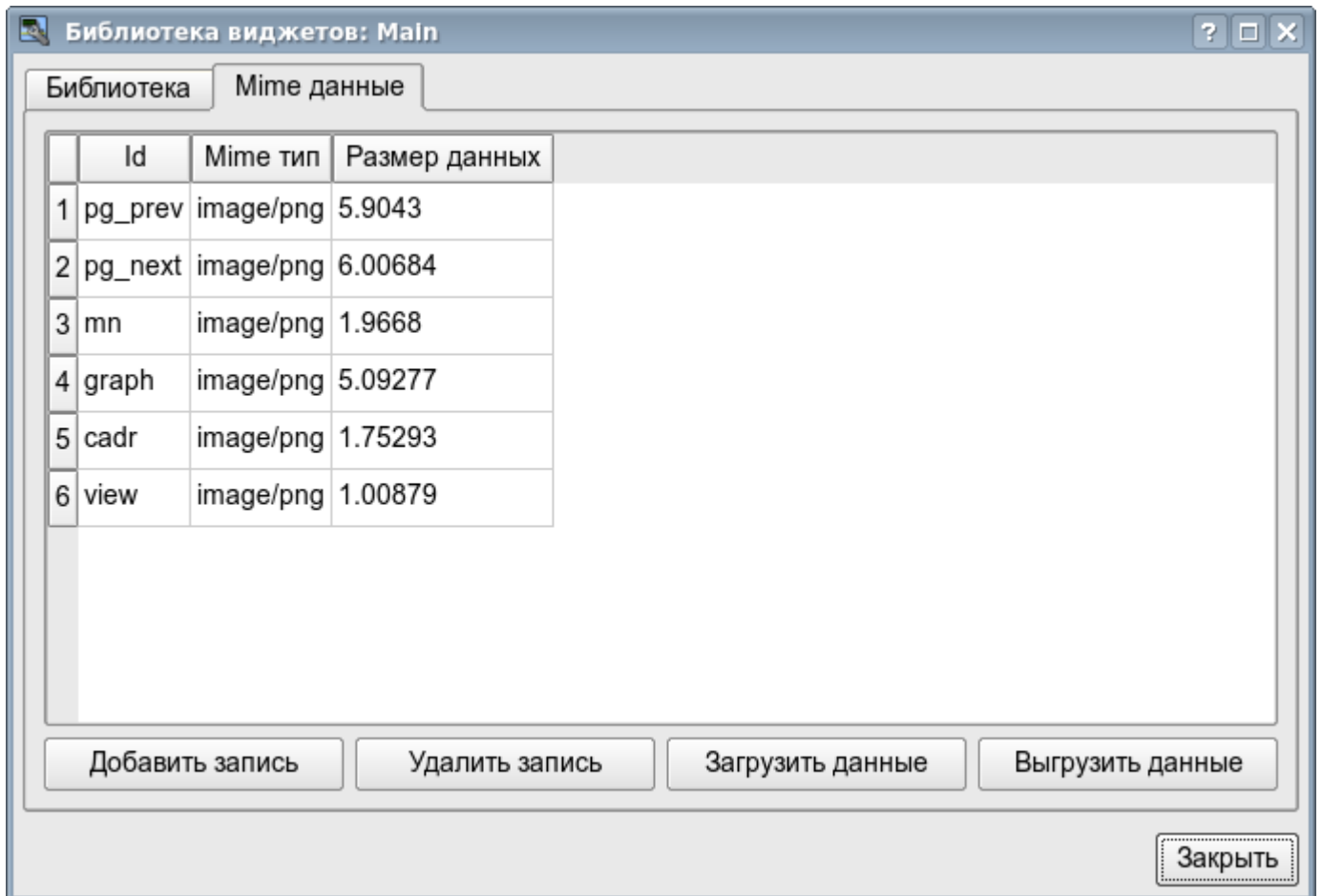


Рис.4. Вкладка редактирования mime-данных контейнера визуальных элементов.

Диалог редактирования свойств визуальных элементов содержит четыре вкладки: вкладку конфигурации основных параметров (рис.2), вкладку атрибутов элемента (рис.5), вкладку обработки элемента (рис.6) и вкладку связей элемента (рис.7). На разных уровнях иерархии визуальных элементов какие-то вкладки могут быть доступны, а какие-то нет.

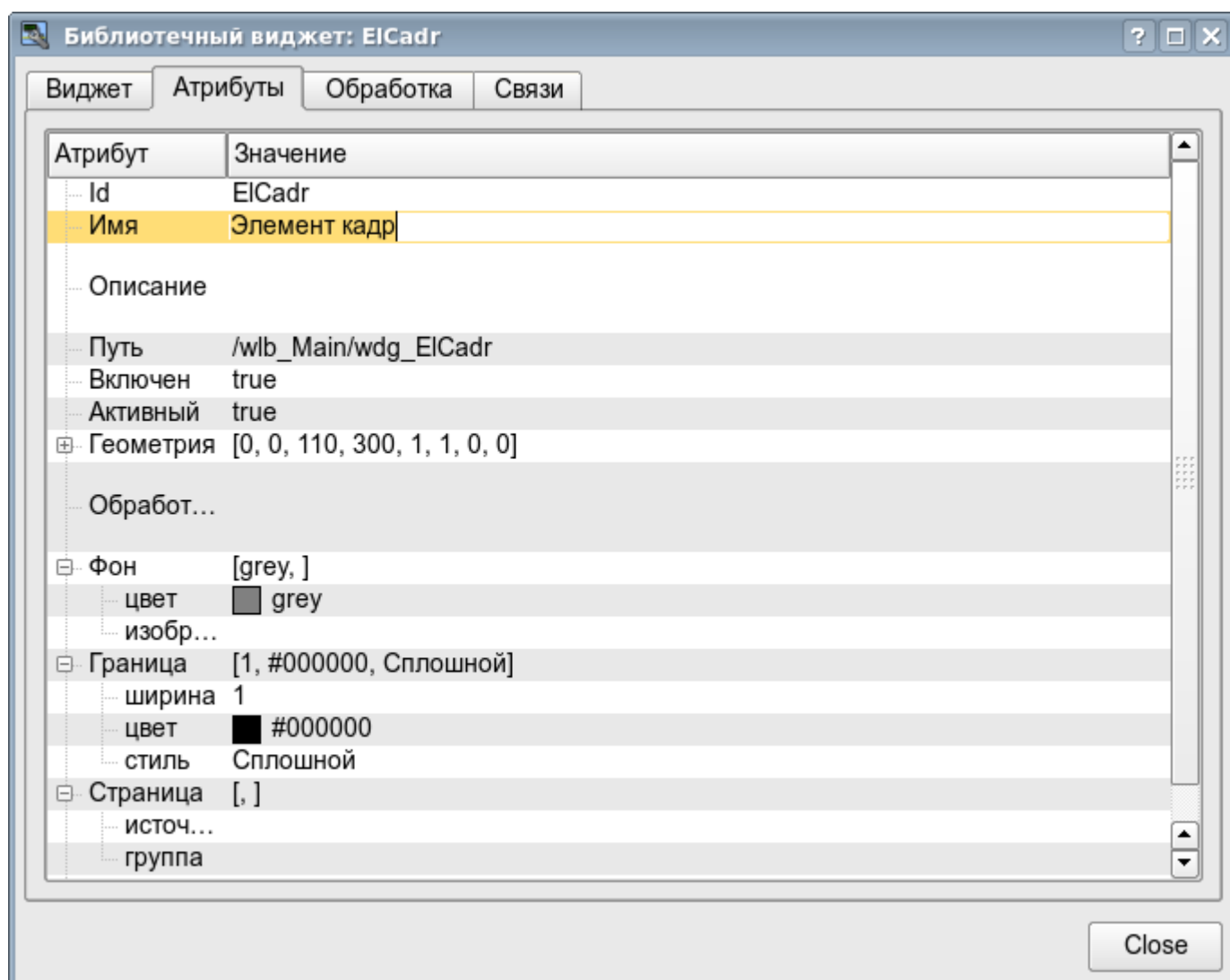


Рис.5. Вкладка атрибутов диалога редактирования свойств визуального элемента.

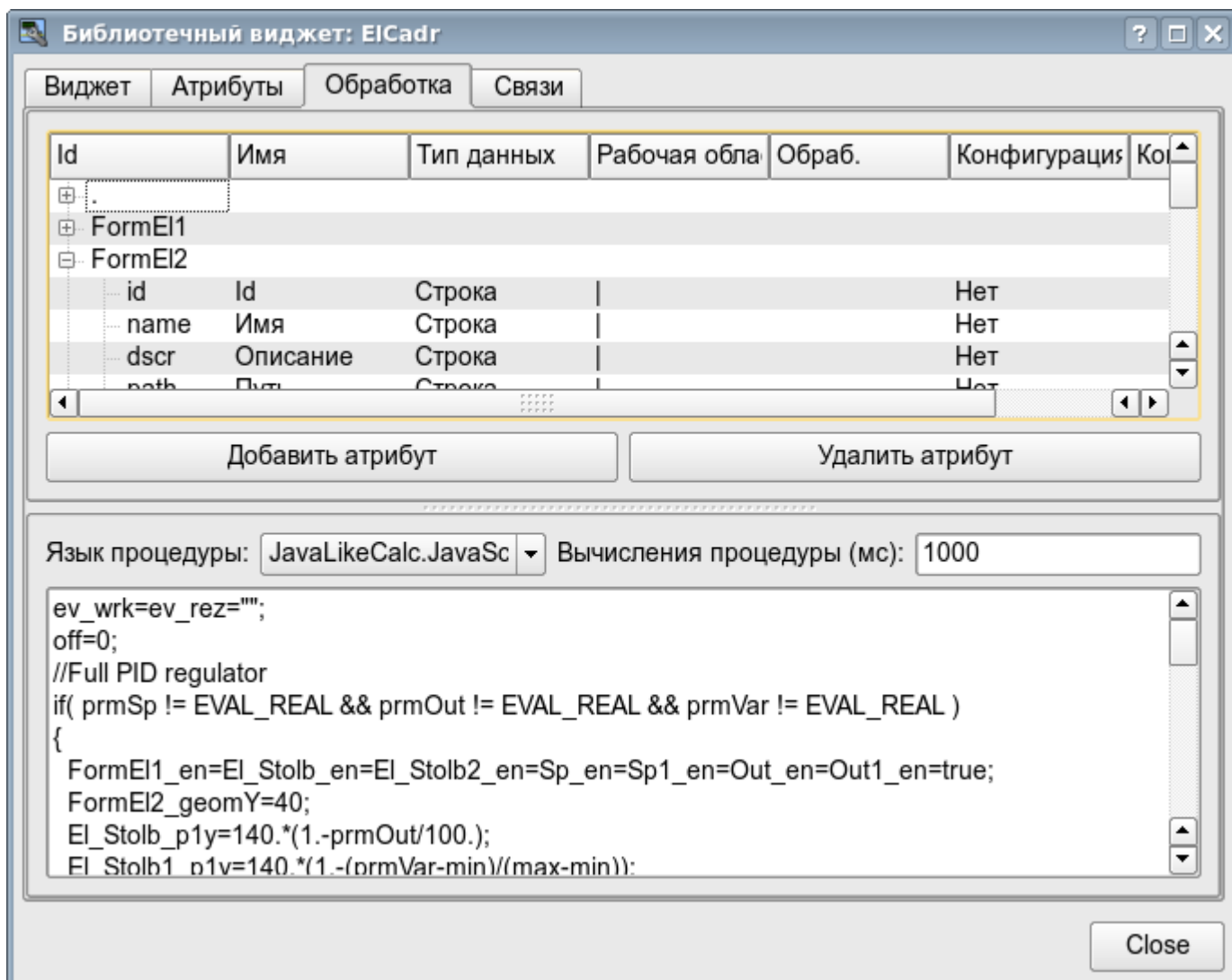


Рис.6. Вкладка обработки диалога редактирования свойств визуального элемента.

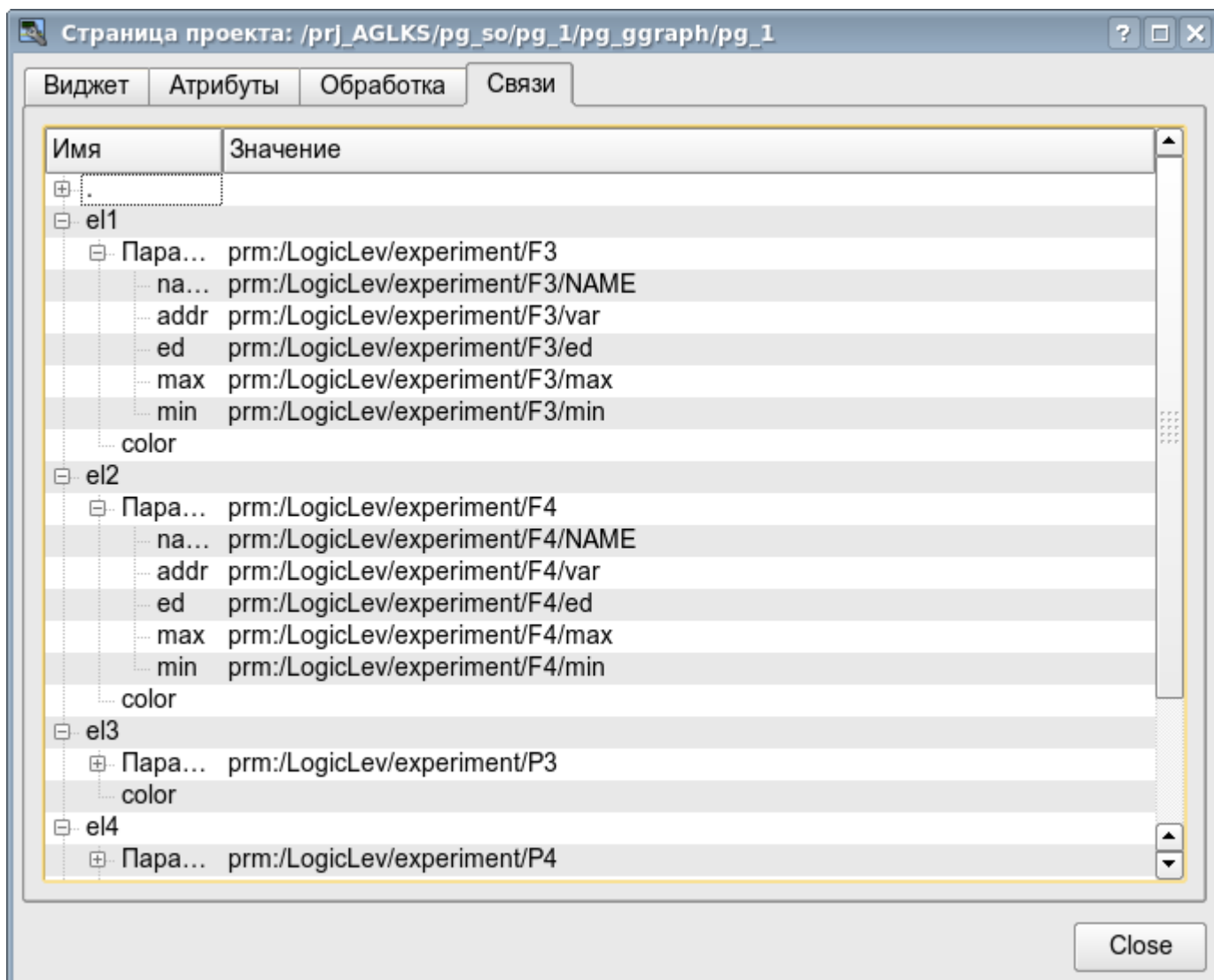


Рис.7. Вкладка связей диалога редактирования свойств визуального элемента.

3 Исполнение интерфейсов СВУ

Исполнение интерфейса СВУ заключается в запуске нового сеанса проекта или подключения к существующему на уровне движка СВУ. Далее модуль непосредственной визуализации отражает и управляет данными сеанса. Главное окно режима исполнения данного модуля имеет вид, представленный на рис.8.

Реализовано обновление содержимого открытых страниц интерфейса визуализации с периодичностью исполнения сессии проекта. В процессе обновления выполняется:

- запрос списка открытых страниц, с признаком модификации страницы, у модели и проверка соответствия реально открытых страниц этому списку;
- запрос ветви данных модифицированных страниц;
- обновление содержимого модифицированных страниц и их виджетов в соответствии с полученными измененными данными.

По закрытию «RunTime» окна производится и закрытие сессии проекта в движке СВУ.

Механизм запроса только изменённых данных основан на абсолютном счётчике исполнения сессии. При внесении реальных изменений в атрибуты виджетов выполняется запоминание значения этого счётчика, что и позволяет идентифицировать изменённые атрибуты. Такой подход позволяет повысить производительность и уменьшить нагрузку на трафик в случае доступа к движку СВУ через сеть.

Иерархически модулем предусматривается возможность размещения страниц проекта как на главном окне исполнения (рис.8), так и вкладывая внутрь виджетов контейнеров, а также путем открытия дополнительных окон поверх основного.

При разворачивании главного окна исполнения или переходе в полноэкранный режим выполняется масштабирование содержимого страницы интерфейса СВУ, заполняя всё пространство окна и позволяя, тем самым, исполнять проекты разработанные на одно разрешение экрана на разных разрешениях.

Главное окно состоит из меню (сверху) строки статуса (снизу) и исполняемого содержимого сеанса между ними. Меню имеет следующую структуру:

- «Файл» — Общие операции.
 - «Печать» — Печать:
 - «Страница» — страницу пользовательского интерфейса;
 - «Диаграмма» — диаграмму на пользовательском интерфейсе;
 - «Документ» — документ на пользовательском интерфейсе.
 - «Экспорт» — Экспорт:
 - «Страница» — страницу пользовательского интерфейса;
 - «Диаграмма» — диаграмму на пользовательском интерфейсе;
 - «Документ» — документ на пользовательском интерфейсе.
 - «Закрыть» — закрыть окно редактора.
 - «Выход» — Выход из системы OpenSCADA.
- «Нарушение» — Квитация нарушений:
 - «Уровень нарушения» — все нарушения;
 - «Световое предупреждение» — световое уведомление;
 - «Предупреждение гудком» — уведомление гудком;
 - «Звуковое/речевое предупреждение» — уведомление звуком/речью.
- «Вид» — Параметры отображения сеанса проекта.
 - «Весь экран» — переключатель полноэкранного режима исполнения.
- «Помощь» — Помощь по OpenSCADA и модулю Vision.
 - «Про» — информация про данный модуль.

- «Про QT» — информация о библиотеке QT, используемой модулем.
- «Что это» — запрос описания элементов интерфейса окна.

С правой стороны строки статуса размещаются индикаторы текущей станции движка СВУ и пользователя, от имени которого ведётся разработка интерфейса СВУ, а также кнопки квитации нарушений. По двойному клику на индикаторе пользователя можно пользователя сменить, введя новое имя и пароль пользователя, а по клику на кнопку квитации – сквитировать нарушения полностью или только нужное уведомление. В главное поле строки статуса выводятся различные информационные сообщения и сообщения помощи.

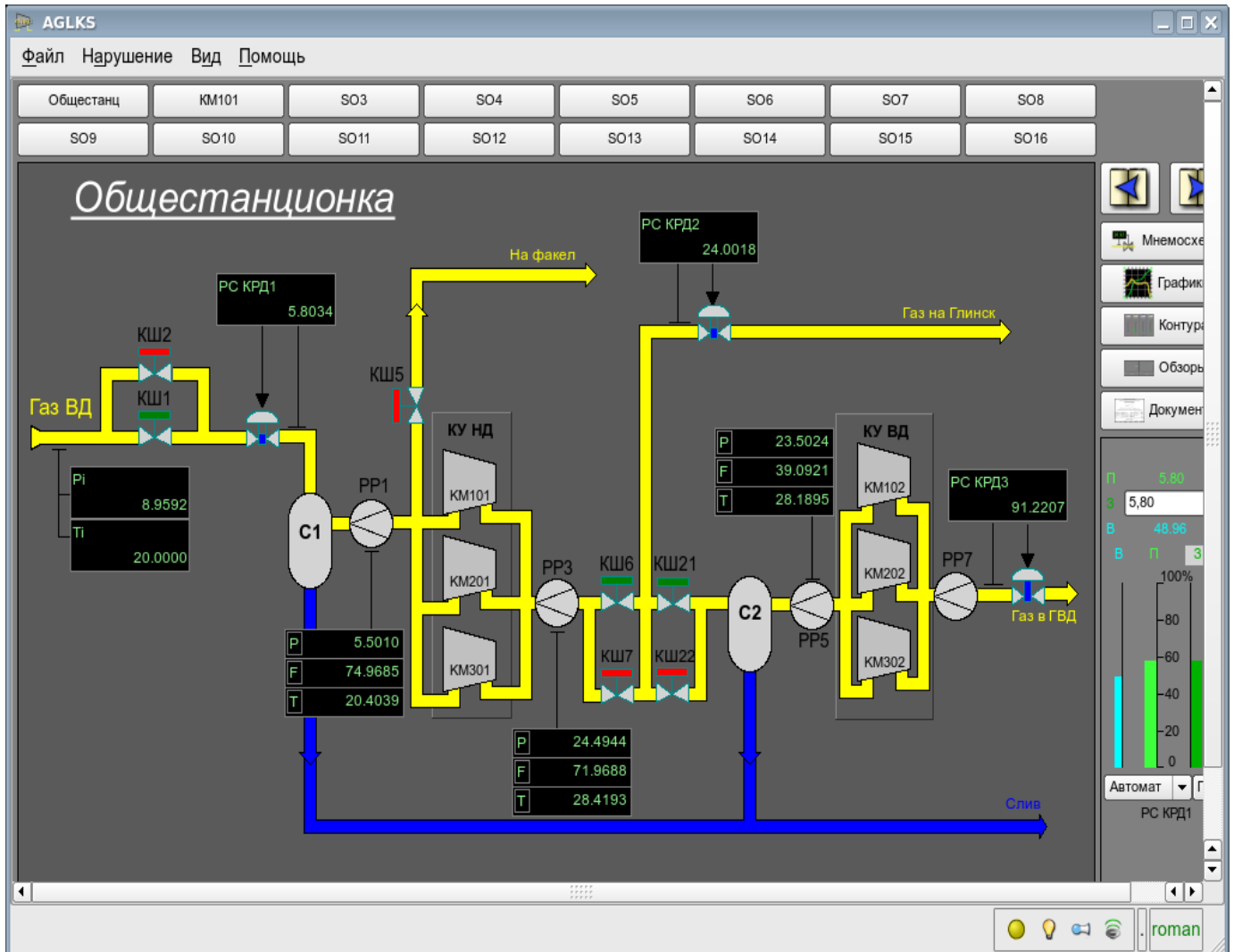


Рис.8. Главное окно режима исполнения.

4 Представление базовых элементов (примитивов)

В данной версии этого модуля реализованы не все образы примитивов заложенные этим проектом. В общем же проектом заложены примитивы:

Id	Наименование	Функция
ElFigure	Элементарные графические фигуры	<p>Примитив является основой для отрисовки элементарных графических фигур со всевозможной комбинацией их в одном объекте. Предусматривается поддержка следующих элементарных фигур:</p> <ul style="list-style-type: none"> • Линия. • Дуга. • Кривая безье. • Заливка замкнутого пространства. <p>Для всех фигур, содержащихся в виджете устанавливаются единые свойства толщины, цвета и т.д., но это не исключает возможность указания вышеперчисленных атрибутов конкретно для каждой фигуры отдельно.</p>
FormEl	Элементы формы.	<p>Включает поддержку стандартных компонентов формы:</p> <ul style="list-style-type: none"> • Редактирование строки. • Редактирование текста. • Флажок. • Кнопка. • Поле выбора из списка. • Список. • Слайдер. • Строка прокрутки.
Text	Текст	Элемент текста(метки). Характеризуется типом шрифта, цветом, ориентацией и выравниванием.
Media	Медиа	Элемент отображения растровых и векторных изображений различных форматов, проигрывания анимированных изображений, проигрывание аудио фрагментов и просмотр видео-фрагментов. Возможно в него стоит включить поддержку OpenGL!
Diagram	Диаграмма	Элемент диаграммы с поддержкой возможности отображения нескольких потоков трендов и различных режимов отображения, от минималистического до полноэкранного, двухмерного, трёхмерного, кругового и т.д.
Protocol	Протокол	Элемент протокола, визуализатора системных сообщений, с поддержкой различных режимов работы при различных размерах и установках.
Document	Документ	Элемент формирования отчётов, журналов и другой документации на основе указанных данных.
Function	Функция API объектной модели OpenSCADA	Невизуальный, на стороне исполнения, виджет, позволяющий включать вычислительные функции объектной модели OpenSCADA в СБУ.
Box	Контейнер	Содержит механизм размещения других виджетов с целью формирования новых, более сложных виджетов и страниц конечной визуализации.

Id	Наименование	Функция
Link	Связующая линия	Данный элемент является не только визуальным, но и логическим, и служит для визуальной и логической связи элементов контейнера (Box) между собой. Связи накладываются между узловыми точками, описанными для виджетов индивидуально. Узловые точки могут описывать только визуальные связи или же содержать логические свойства для наложения связей.

Более детально рассмотрим реализацию каждого примитива.

4.1 Примитив элементарная фигура (ElFigure)

Реализована поддержка элементарных фигур: линии, эллиптической дуги, кривой Безье и заливка замкнутых контуров цветом и/или изображением. Для элементарных фигур реализованы следующие операции:

- создание/удаление фигур;
- копирование фигур;
- перемещение и изменение размеров фигур с помощью мыши и клавиатуры;
- возможность связывать элементарные фигуры друг с другом, получая более сложные, для которых доступны все свойства исходных элементарных фигур;
- возможность одновременного перемещения нескольких фигур;
- заливка замкнутого контура цветом и/или изображением;
- генерация событий клавиш мыши в момент клика мышью на залитые контура;
- масштабирование;
- поворот.

На рис. 9 представлена часть экрана с кадром, содержащим элементарные фигуры.

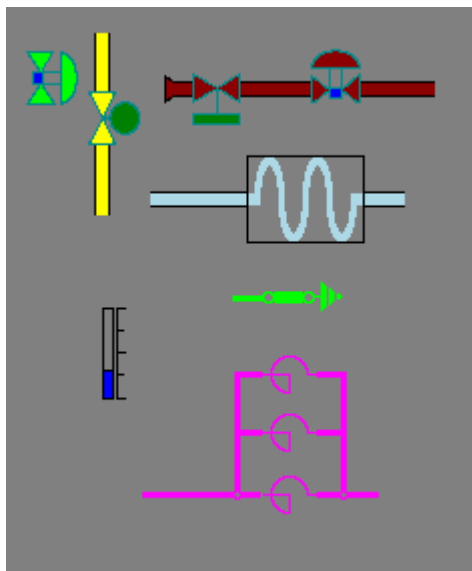


Рис.9 Реализация элементарных фигур в Vision.

Фигуры, лежащие в основе данного виджета, содержат точки(начальная и конечная), которые могут стыковаться с соответственными точками других фигур, и точки, с помощью которых изменяется геометрия фигуры.

Добавить фигуру можно с помощью мыши:

1. Выбрать желаемую фигуру из контекстного меню.
2. Задать с помощью левой кнопки мыши начальную и конечную точки (для линии при удерживании клавиши SHIFT происходит ортогональная её отрисовка).

Удалить фигуру(ы) можно путём нажатия кнопки “Del”, имея выделенную(ые) фигуру(ы).

Скопировать фигуру(ы) можно путём нажатия комбинации клавиш “Ctrl” + “C”, имея выделенную(ые) фигуру(ы).

Передвинуть/изменить габариты фигуры можно с помощью мыши или клавиатуры:

1. Выделить фигуру, нажав по ней левой кнопкой мыши.
2. Перетащить (с помощью мыши или управляющих клавиш) фигуру или одну из её контрольных точек в желаемое место и отпустить кнопку мыши(клавишу).

Существует возможность перемещать несколько выделенных фигур, выбранных при помощи удержания “Ctrl”(эта опция работает при отключенной кнопке Holds(Привязки)) либо при помощи выделения мышкой.

Связать фигуры друг с другом можно следующим образом:

1. Нажать кнопку Holds.
2. Выделить одну из фигур и переместить её начальную или конечную точку к желаемой начальной или конечной точке другой фигуры так, чтобы она попадала в появившуюся окружность. Связанные фигуры перемещаются также как и отдельные, общая точка перемещается для всех фигур, к которым она относится(приоритет отдается дуге).

Залить замкнутый контур из фигур можно следующим образом:

1. Нажать кнопку Holds.
2. Создать замкнутый контур.
3. Два раза нажать мышкой внутри его.

Удалить заливку замкнутого контура можно, разорвав контур, либо двойным кликом мышки по уже имеющемуся залитому контуру.

Поворот фигуры осуществляется вокруг центра виджета.

4.2 ПрIMITив текста (Text)

Реализована поддержка элемента текста со свойствами:

- Шрифт со свойствами: типа/класса шрифта, размера, усиления, наклонности, подчёркивания и перечёркивания.
- Цвет текста.
- Ориентация текста.
- Автоматический перенос по словам.
- Выравнивание текста по горизонтали и вертикали со всеми вариантами.
- Отображение фона в виде цвета и/или изображения.
- Отображение бордюра вокруг текста, с указанным цветом, шириной и стилем.
- Формирование текста из атрибутов различного типа и свойств.

На рис. 10 представлена часть экрана с кадром, содержащим примеры текста с использованием различных параметров.



Рис.10. Реализация базового элемента текста в Vision.

4.3 Примитив элементов формы (FormEl)

Реализована поддержка элементов формы на кадрах СВУ. Реализованы заложенные свойства, включая следующие элементы формы:

Редактор строки — Представлено следующими видами: «Текст», «Combo», «Целое», «Вещественное», «Время», «Дата», «Время и Дата». Все виды редактора строки поддерживают подтверждение ввода.

Редактор текста — Представляет редактор плоского текста с подтверждением или отказом от ввода.

Поле флажка — Предоставляет поле бинарного флажка.

Кнопка — Предоставляет кнопку с поддержкой: цвета кнопки, изображения в кнопке и режима фиксации.

Выбор из списка — Предоставляет поле выбора элемента, со списка указанных элементов.

Список — Предоставляет поле списка с контролем за текущим элементом.

Слайдер — Элемент слайдера.

Прогрес-бар — Полоска прогресс-бара.

Реализованы режимы: «Включен» и «Активен», а также передача изменений и событий в модель данных СВУ (движок).

На рис. 11 представлена часть экрана с кадром, содержащим вышеперечисленные элементы формы.

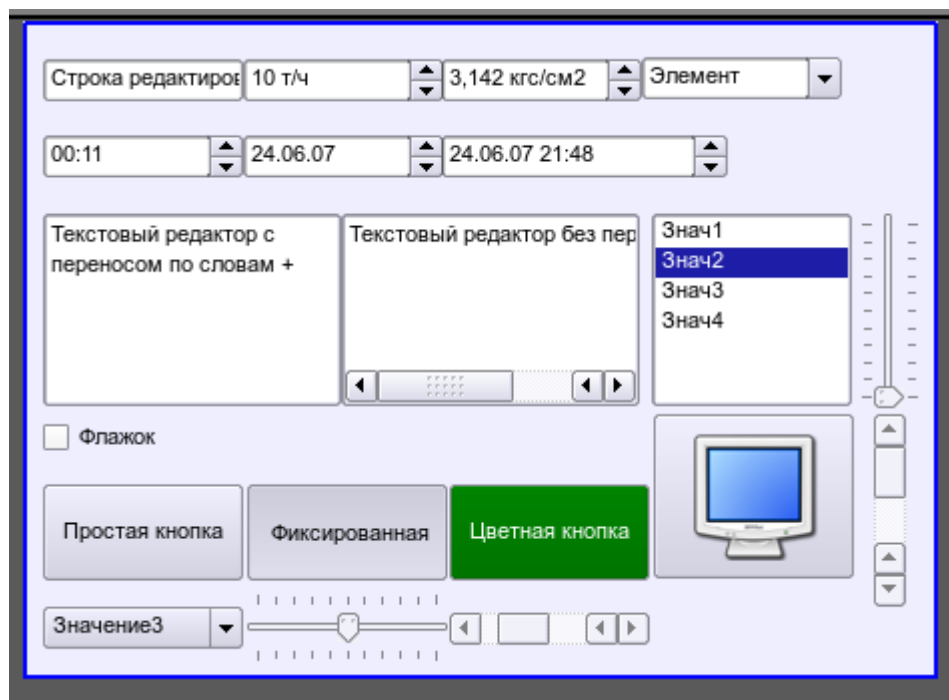


Рис.11. Реализация элементов формы в Vision.

4.4 Прimitives отображения медиа-материалов (Media)

Реализована поддержка элемента отображения медиа-материалов со свойствами:

- Указания источника медиа данных (изображения или видео-материала).
- Просмотра изображений большинства известных форматов с возможностью их вписывания в размер виджета.
- Проигрывания простых анимированных форматов изображений и видео, с возможностью управления скоростью проигрывания.
- Отображение фона в виде цвета и/или изображения.
- Отображение бордюра вокруг текста, с указанным цветом, шириной и стилем.
- Формирования активных областей и генерация событий при их активации.

На рис. 12 представлена часть экрана с кадром, содержащим примеры просмотра/проигрывания медиа-данных.

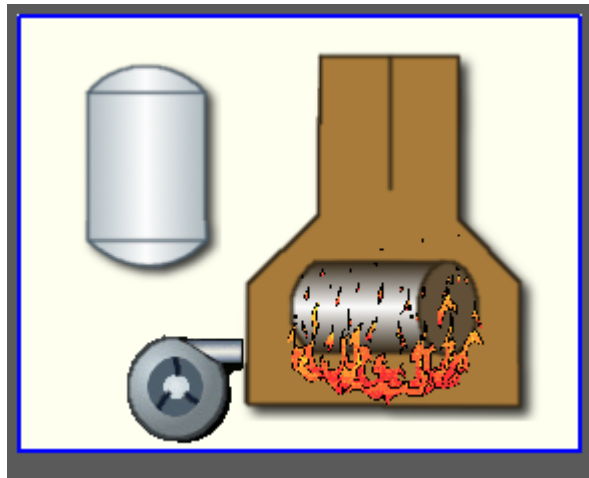


Рис.12. Реализация базового элемента отображения медиа-материалов в Vision.

4.5 Примитив построения диаграм/графиков (Diagram)

Реализована поддержка элемента построения диаграм/трендов со свойствами:

- Построение графиков/трендов:
 - Построения графика для: архивных данных, текущих данных и формирования промежуточного буфера отображения для параметров без архива.
 - Построение как одиночных графиков со значением параметра по оси ординат, так и сводных графиков, включающих до 10 параметров, с процентной шкалой.
 - Возможность адаптации графика параметра к значениям данным, подгон шкалы.
 - Широкий диапазон масштабирования и адаптации горизонтальной шкалы, с автоматическим усреднением на уровне сервера и самого примитива.
 - Возможность отображение размерной сетки и маркеров по горизонтали и вертикали, с адаптацией к диапазону отображения.
 - Поддержка активного режима, с курсором и получением значений под курсором.

На рис. 13 представлена часть экрана с кадром, содержащим примеры диаграммы-тренда.

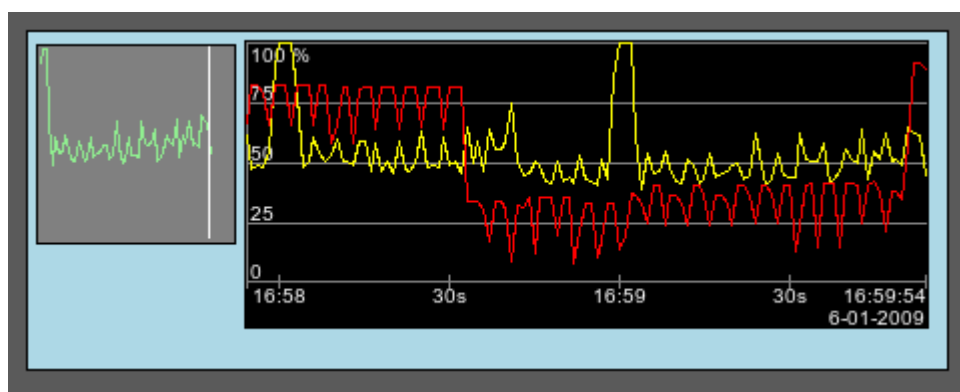


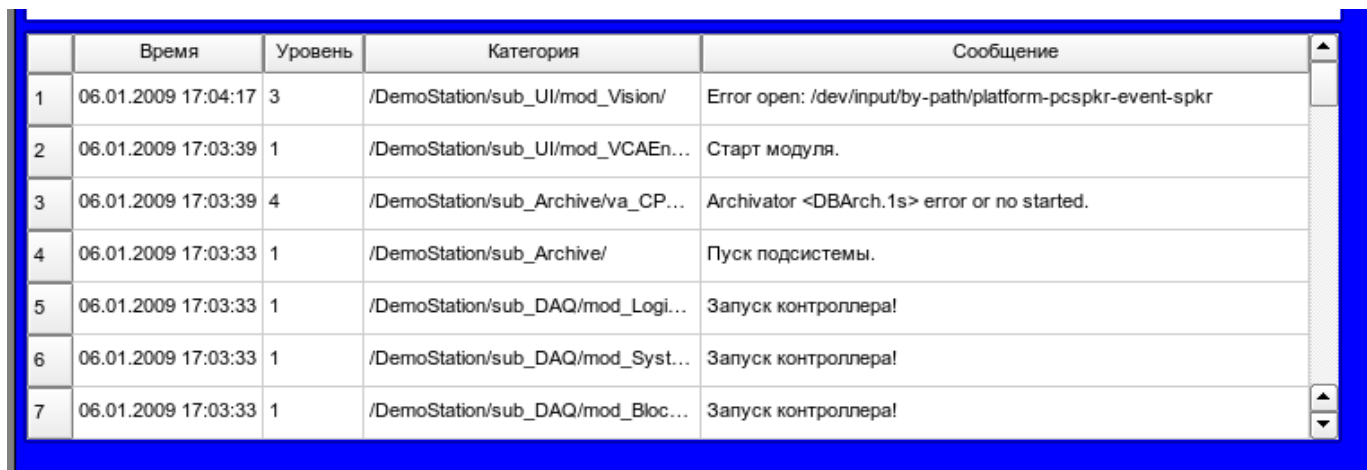
Рис.13. Реализация базового элемента отображения диаграммы-тренда в Vision.

4.6 Примитив формирования протокола (*Protocol*)

Реализована поддержка элемента формирования протокола со свойствами:

- Формирование протокола из архива сообщений за указанное время и глубину.
- Запрос данных из указанных архиваторов сообщений.
- Выборка данных из архивов по уровню важности и шаблону категории сообщений.
- Поддержка режима слежение за появлением сообщений в архиве сообщений.

На рис. 14 представлена часть экрана с кадром, содержащим пример протокола.



	Время	Уровень	Категория	Сообщение
1	06.01.2009 17:04:17	3	/DemoStation/sub_UI/mod_Vision/	Error open: /dev/input/by-path/platform-pcspkr-event-spkr
2	06.01.2009 17:03:39	1	/DemoStation/sub_UI/mod_VCAEn...	Старт модуля.
3	06.01.2009 17:03:39	4	/DemoStation/sub_Archive/va_CP...	Archivator <DBArch.1s> error or no started.
4	06.01.2009 17:03:33	1	/DemoStation/sub_Archive/	Пуск подсистемы.
5	06.01.2009 17:03:33	1	/DemoStation/sub_DAQ/mod_Logi...	Запуск контроллера!
6	06.01.2009 17:03:33	1	/DemoStation/sub_DAQ/mod_Syst...	Запуск контроллера!
7	06.01.2009 17:03:33	1	/DemoStation/sub_DAQ/mod_Bloc...	Запуск контроллера!

Рис.14 Реализация базового элемента отображения протоколов в Vision.

4.7 ПрIMITив формирования отчётной документации (*Document*)

Реализована поддержка элемента формирования отчётной документации со свойствами:

- Гибкое формирование структуры документа на основе языка гипертекстовой разметки. Это предоставляет поддержку широких возможностей форматирования документов.
- Формирование документов по команде или по графику. Необходимо для формирования отчётной документации в архив с последующим просмотром архива.
- Формирование документа в режиме реального времени. Для формирования документов полностью динамически и на основе архивов за указанное время.
- Использование атрибутов виджета для передачи значений и адресов на архивы в документ. Позволяет использовать виджет документа как шаблон при формирования отчётов с другими входными данными.

В основе любого документа лежит XHTML-шаблон. XHTML-шаблон это тег 'body', WEB-страницы, содержащий статику документа в стандарте XHTML 1.0 и элементы исполняемых инструкций на одном из языков пользовательского программирования Open SCADA² в виде `<?dp <procedure> ?>`. Результирующий документ формируется путём исполнения процедур и вставки их результата в документ.

Источником значений исполняемых инструкций являются атрибуты виджета этого примитива, а также все механизмы языка пользовательского программирования. Атрибуты могут добавляться пользователем и линковаться на реальные атрибуты параметров или-же являться автономными, значения которых будут формироваться в скрипте виджета. В случае со ссылками на атрибуты могут извлекаться значения из истории, архива.

На рис. 15 представлен кадр, содержащий пример документа.

ОАО "Днепропетровский меткомбинат"
(наименование предприятия, на котором установлен расходомер)

СУТОЧНЫЙ ОТЧЁТ

за "_____" "_____" 2008 г.
составлен _____
(дата) (время)

"ФЛОУТЭК-ТМ" _____ Трубопровод _____
(имя вычислителя или корректора) (имя нитки)

Характеристики Расходомера со стандартным сужающим устройством:

Контрактный час	17 20	Тип отбора	Угловой	Атм. давление, кПа	95
Мол. доля N ₂ , %	70	К-т шероховатости	0.2	Отсечка, кПа	80
Мол. доля CO ₂ , %	10	К-т притупления	0.1	Верхн. предел ПД, кПа	150
Диаметр трубы, мм	100	К-т L (альфа)	0.3	Порог переключ., кПа	85
Диаметр СУ, мм	60	Отн. площадь СУ	0.6	Дин. вязкость, кгс/м ²	32

Часовые данные

Дата	Время		Объём, тыс. м ³	Ср. разность давления, кПа	Среднее давление, МПа	Средняя температура, °C	Средняя плотность, кг.м ³
	начало	конец					
06 01 2009	09:00	10:00	Пусто	Пусто	Пусто	Пусто	Пусто
06 01 2009	10:00	11:00	1335.44	6.07	9.01	14.86	0.98
06 01 2009	11:00	12:00	1531.57	6.06	9.00	14.98	1.00
06 01 2009	12:00	13:00	1394.48	6.06	8.99	14.99	1.00
06 01 2009	13:00	14:00	1445.92	6.06	9.00	14.99	1.00
06 01 2009	14:00	15:00	1645.92	6.06	8.99	15.01	1.00
06 01 2009	15:00	16:00	1845.94	6.06	9.00	15.01	1.00
06 01 2009	16:00	17:00	1551.68	6.06	9.00	15.01	1.00

Рис.15 Реализация базового элемента отображения отчётной документации в Vision.

4.8 Примитив контейнера (Box)

Реализована поддержка примитива контейнера, по совместительству выполняющего роль страниц проектов. Данный примитив является единственным элементом-контейнером, который может включать в себя ссылки на кадры из библиотеки, формируя тем самым пользовательские элементы нужной конфигурации. Примитив реализует предусмотренные проектом свойства. Перечислим по пунктам свойства данного примитива:

Контейнер — Позволяет формировать нужные объекты путём группировки базовых в рамках данного примитива. *Страница* — Элементы, построенные на данном примитиве, могут выполнять роль страницы пользовательского интерфейса. *Контейнер страниц* — Свойство замещения собственного содержимого другой страницей в процессе исполнения. Используется для формирования фреймов на страницах пользовательского интерфейса. Например, главная страница традиционной SCADA системы с объектами сигнализации строится именно таким образом. *Фон* — Поддерживает возможность указания фона в виде цвета или изображения. *Бордюр* — Поддерживает возможность изображения бордюра с указанным цветом, толщиной и стилем.

Пример редактирования кадра, основанного на данном примитиве, приведен на рис. 1, а на рис. 8 изображена страница, содержащая контейнер страниц, построенный на основе данного примитива.

5 Общая конфигурация модуля

Для настройки собственного поведения в не очевидных ситуациях модулем предоставляется возможность настройки отдельных параметров посредством интерфейса управления OpenSCADA (рис. 16). Таковыми параметрами являются:

- Начальный пользователь конфигуратора — указывает от имени какого пользователя открывать конфигуратор без запроса пароля.
- Перечень проектов для запуска их автоматического исполнения с запуском модуля.
- Имя удалённой OpenSCADA станции с движком визуализации СВУ.
- Ссылка на страницу конфигурации перечня внешних OpenSCADA станций.

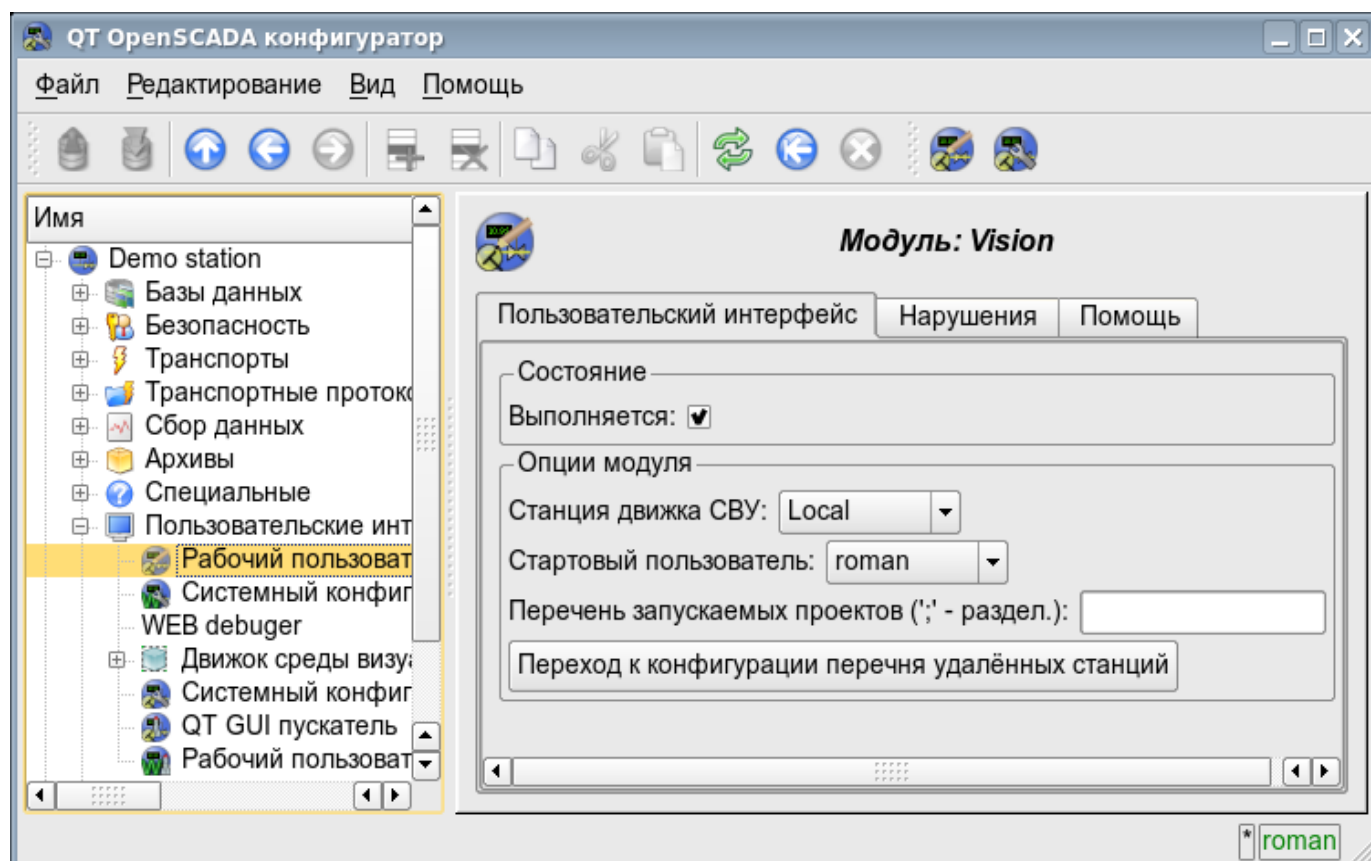


Рис.16. Страница конфигурации модуля.

Заключение

Модуль ещё не реализует всех запланированных функций, однако уже позволяет выполнять базовый их набор, с помощью которого можно строить достаточно развитые интерфейсы ВУ.