

Модуль подсистемы “Пользовательские интерфейсы” <Vision>

Модуль:	Vision
Имя:	Рабочий пользовательский интерфейс (QT)
Тип:	Пользовательские интерфейсы
Источник:	ui_Vision.so
Версия:	1.3.0
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	2
1. Назначение	3
2. Инструмент графического формирования интерфейса СВУ	5
2.1. Стили	14
2.2. Связывание с динамикой	16
3. Исполнение интерфейсов СВУ	18
4. Представление базовых элементов (примитивов)	20
4.1. Примитив элементарная фигура (ElFigure)	21
4.2. Примитив текста (Text)	22
4.3. Примитив элементов формы (FormEl)	23
4.4. Примитив отображения медиа-материалов (Media)	24
4.5. Примитив построения диаграм/графиков (Diagram)	25
4.6. Примитив формирования протокола (Protocol)	25
4.7. Примитив формирования отчётной документации (Document)	26
4.8. Примитив контейнера (Box)	27
5. Векторный графический редактор	28
5.1. Назначение	28
5.2. Принципы и функции разработанного графического редактора	28
5.3. Основные принципы работы в описываемом графическом редакторе	29
5.4. Контекстное меню графического редактора	32
5.5. Диалог свойств элементарной фигуры	33
6. Общая конфигурация модуля	35

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 избежания этих проблем, а также создания в кратчайшие сроки полного спектра 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) (рис. 2.а). Данный подход позволяет одновременно редактировать несколько кадров различных размеров. Используются следующие механизмы управления разработкой: панели инструментов, пункты меню и контекстное меню. Большинство действий дублируются разными механизмами, 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.

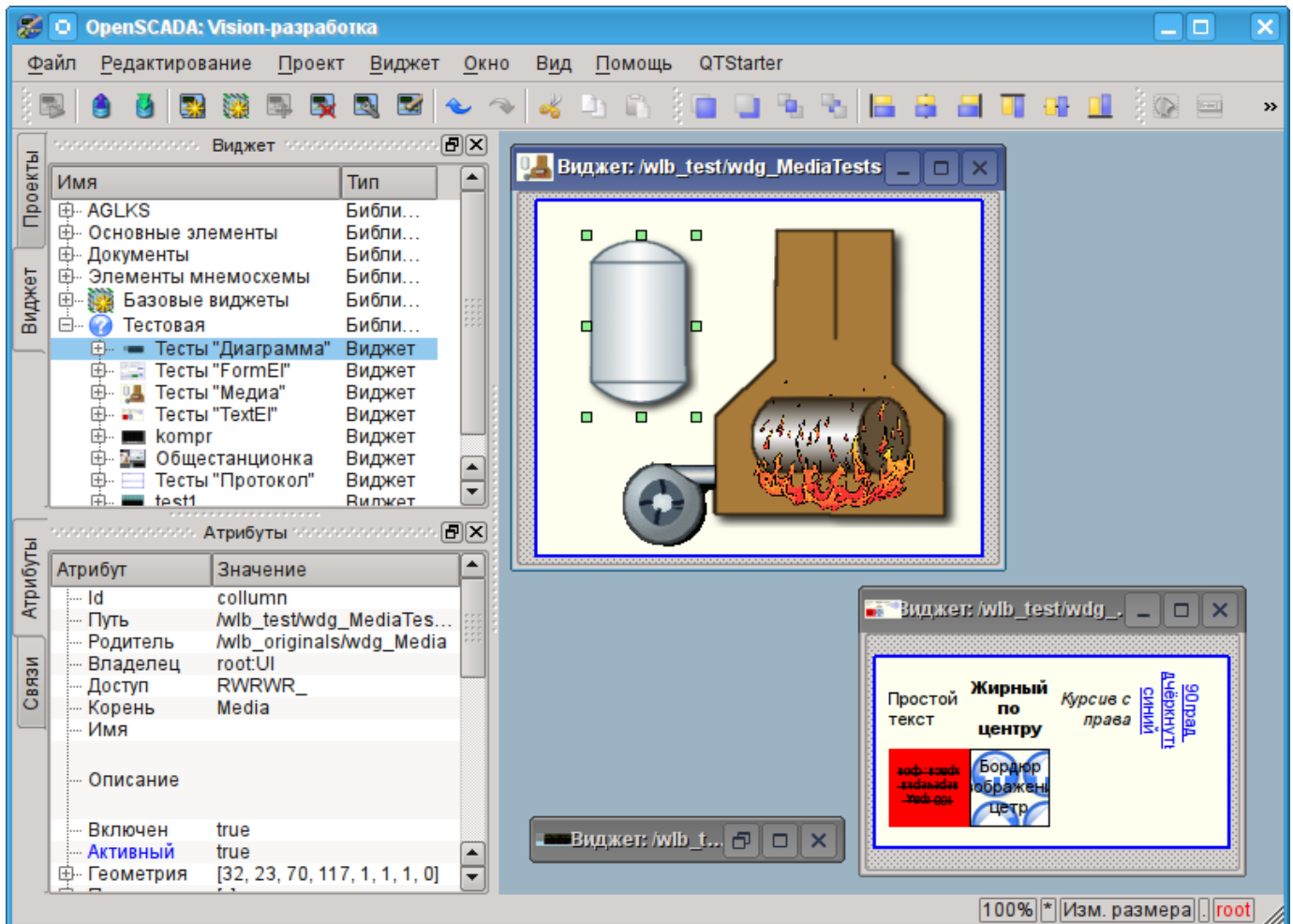


Рис.2.а. Окно разработки интерфейсов СВУ.

Доступ к основным компонентам СВУ производится посредством пришвартованных окон, на рис.2.а 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 очистки и 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.
 - "Очистить визуальный элемент" — очистка визуального элемента с наследованием изменённых свойств или установкой их по умолчанию.
 - "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.
 - "Очистить визуальный элемент" — очистка визуального элемента с наследованием изменённых свойств или установкой их по умолчанию.
 - "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.
 - "Заккрыть" — закрыть окно редактора.
 - "Выход" — Выход из системы OpenSCADA.
- "Редактирование" — Операции редактирования визуальных элементов.
 - "Откат изменения визуального элемента" — осуществление отката последнего изменения визуального элемента.
 - "Повтор изменения визуального элемента" — осуществление повтора изменения визуального 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.
- "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.
- "Очистить визуальный элемент" — очистка визуального элемента с наследованием изменённых свойств или установкой их по умолчанию.
- "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.
 - "Очистить визуальный элемент" — очистка визуального элемента с наследованием изменённых свойств или установкой их по умолчанию.
 - "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.
 - "Заккрыть" — закрыть активное окно.
 - "Заккрыть 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.
 - "Откат изменения визуального элемента" — осуществление отката последнего изменения визуального элемента.
 - "Повтор изменения визуального элемента" — осуществление повтора изменения визуального 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.b. Второй диалог для редактирования свойств самих визуальных элементов, рис. 2.c. 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.

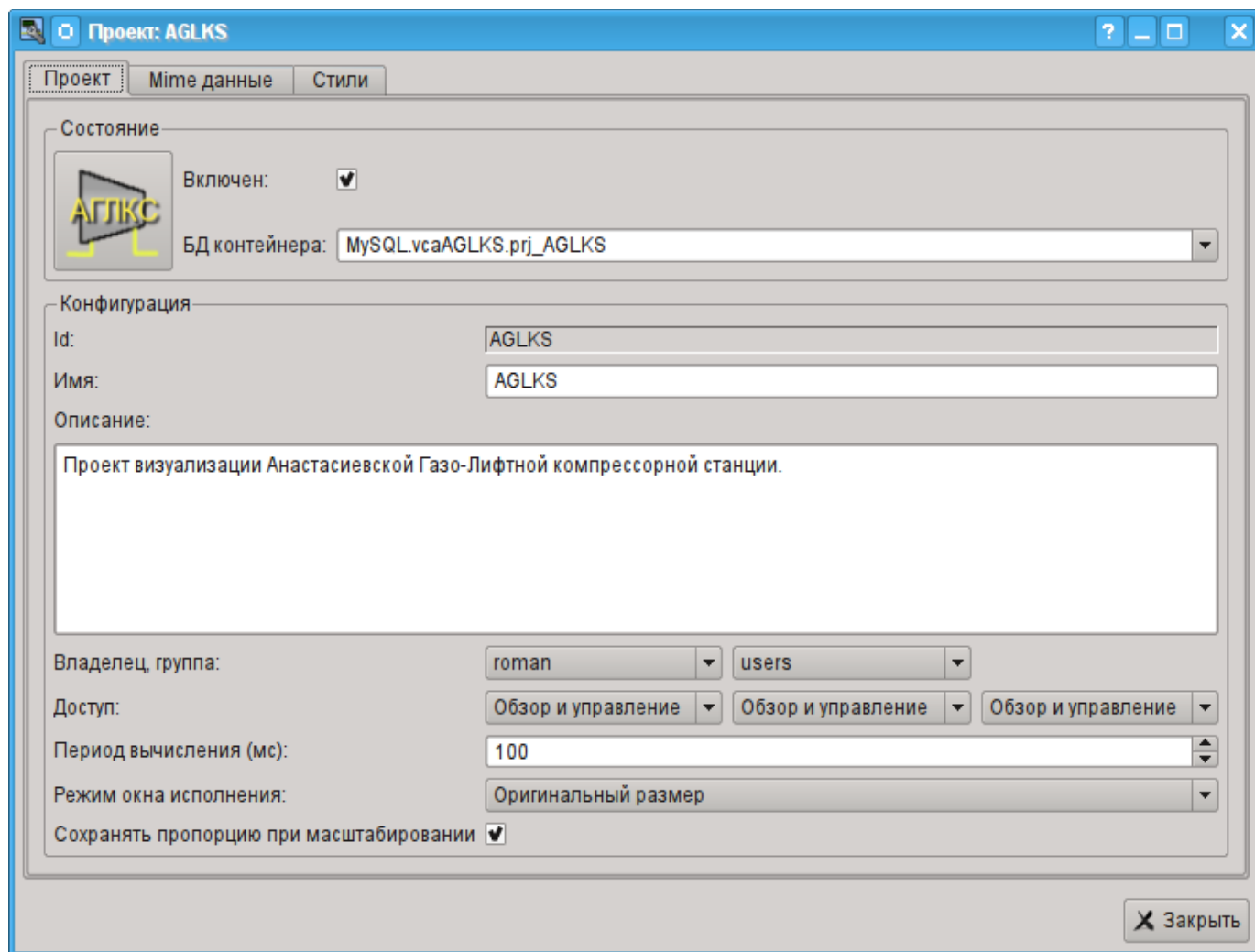


Рис.2.b. Диалог редактирования свойств контейнеров визуальных элементов.

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.

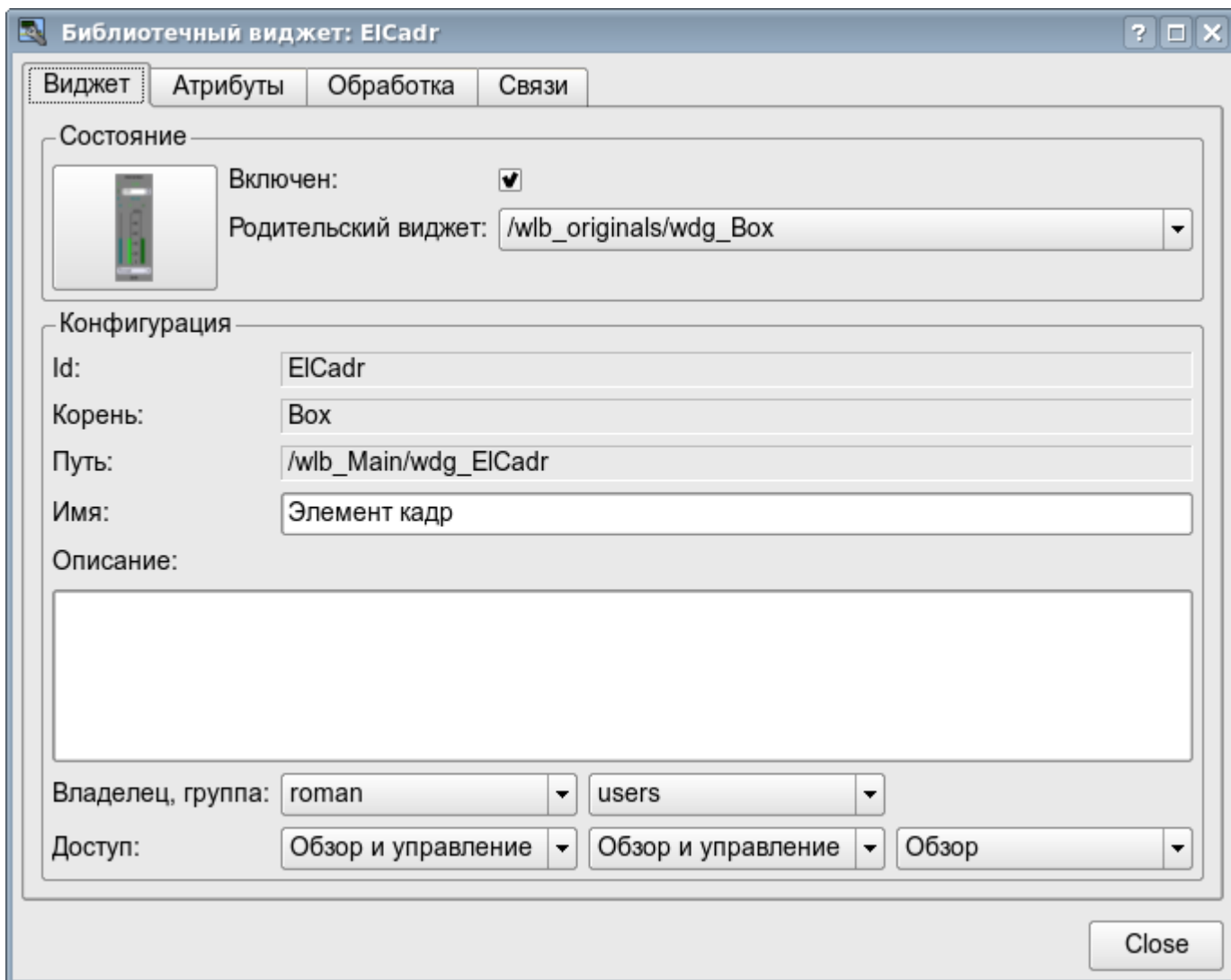


Рис.2.с. Диалог редактирования свойств визуальных элементов.

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.b) и вкладку конфигурации mime-данных контейнеров (рис.2.d).

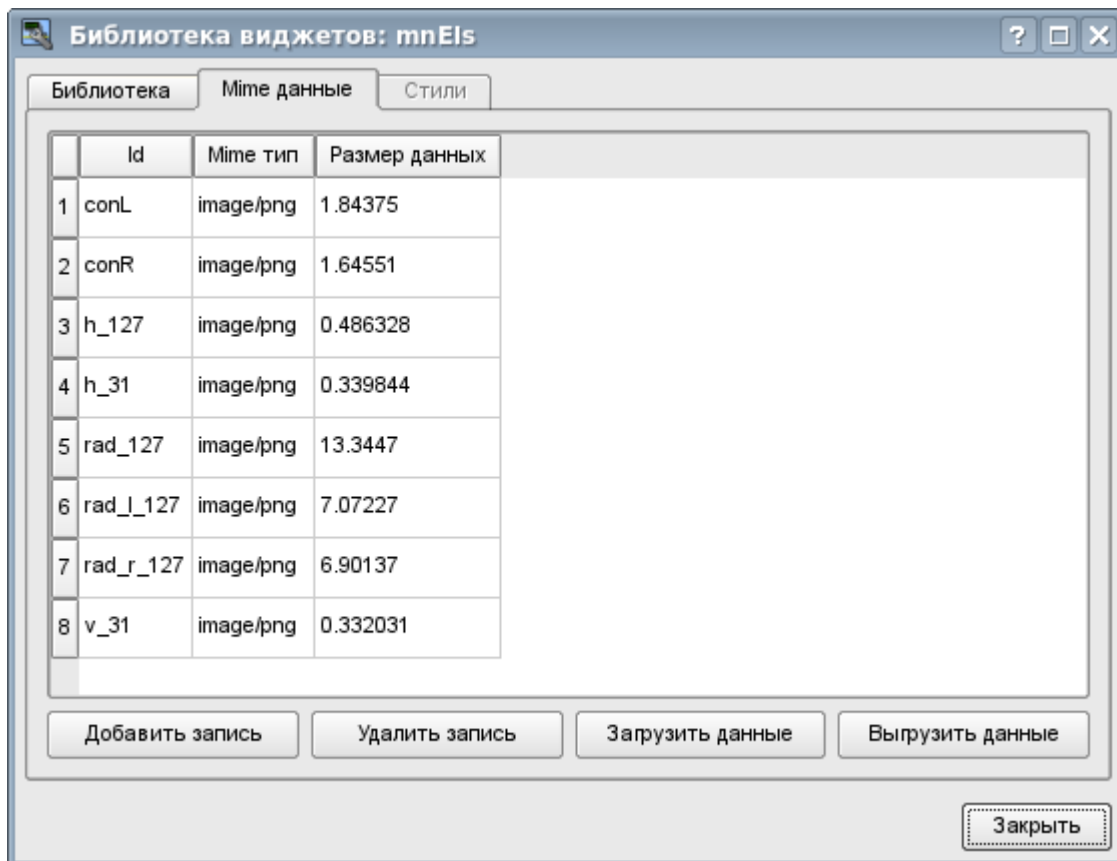


Рис.2.d. Вкладка редактирования mime-данных контейнера визуальных элементов.

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

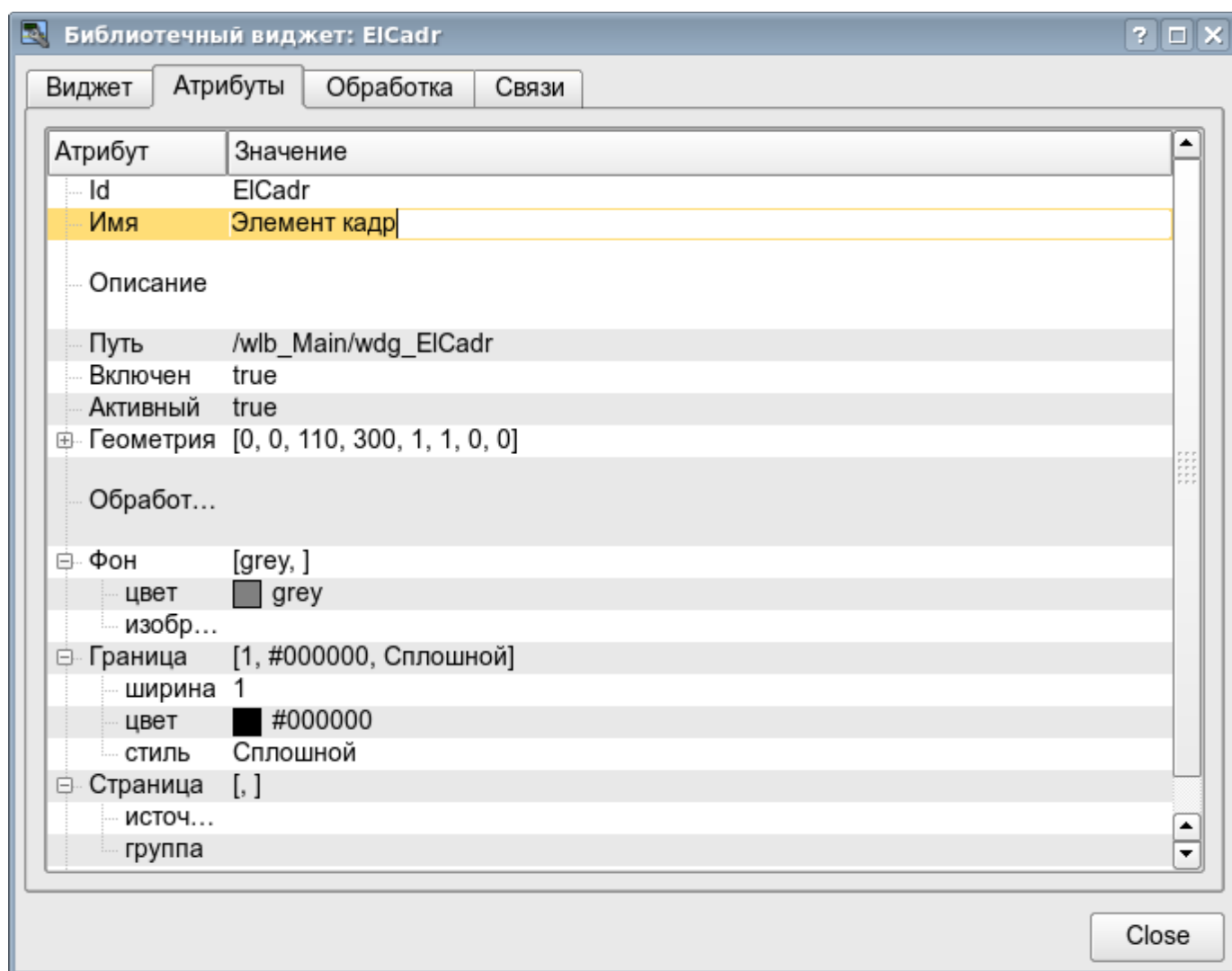


Рис.2.e. Вкладка атрибутов диалога редактирования свойств визуального элемента.

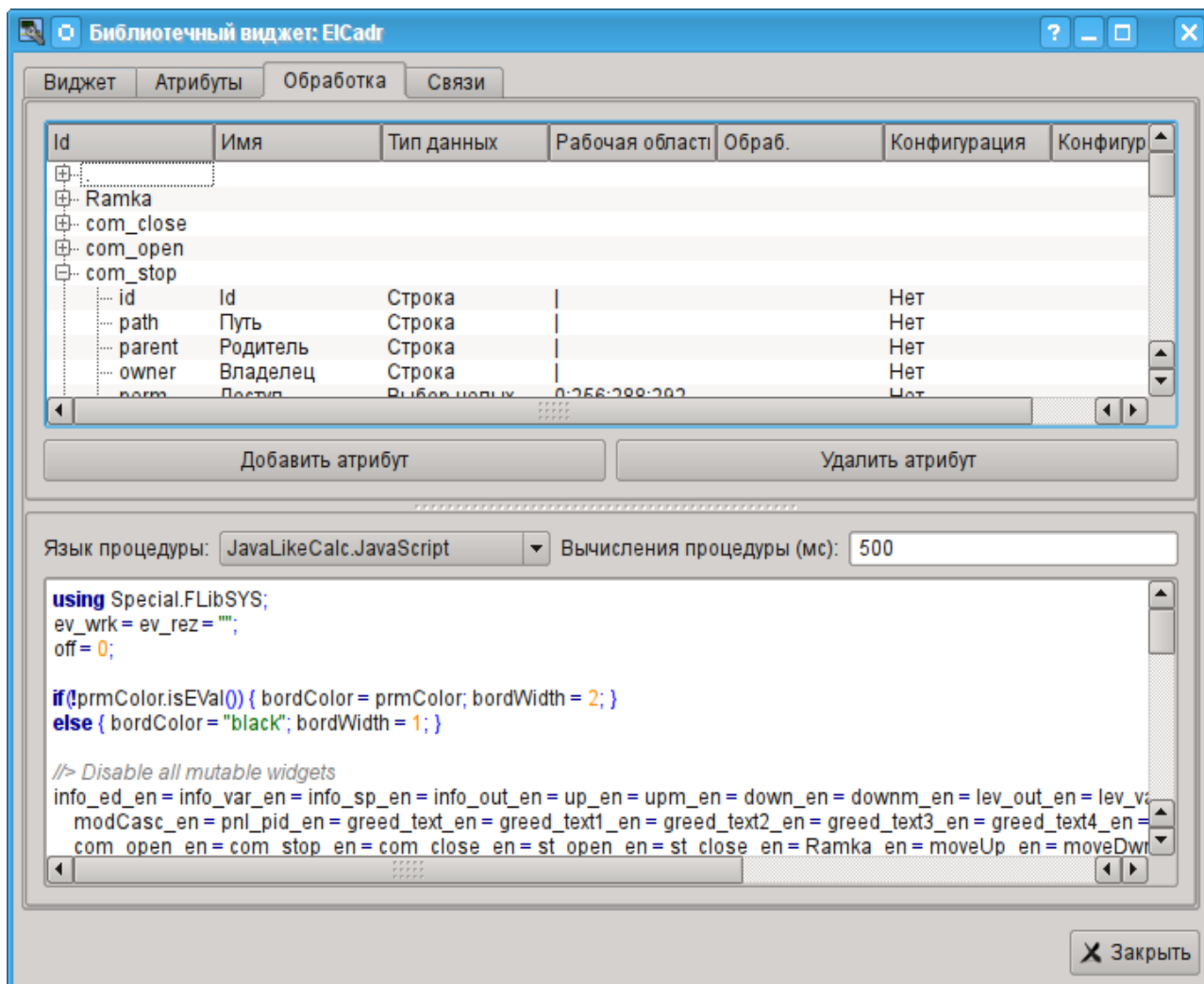


Рис.2.f. Вкладка обработки диалога редактирования свойств визуального элемента.

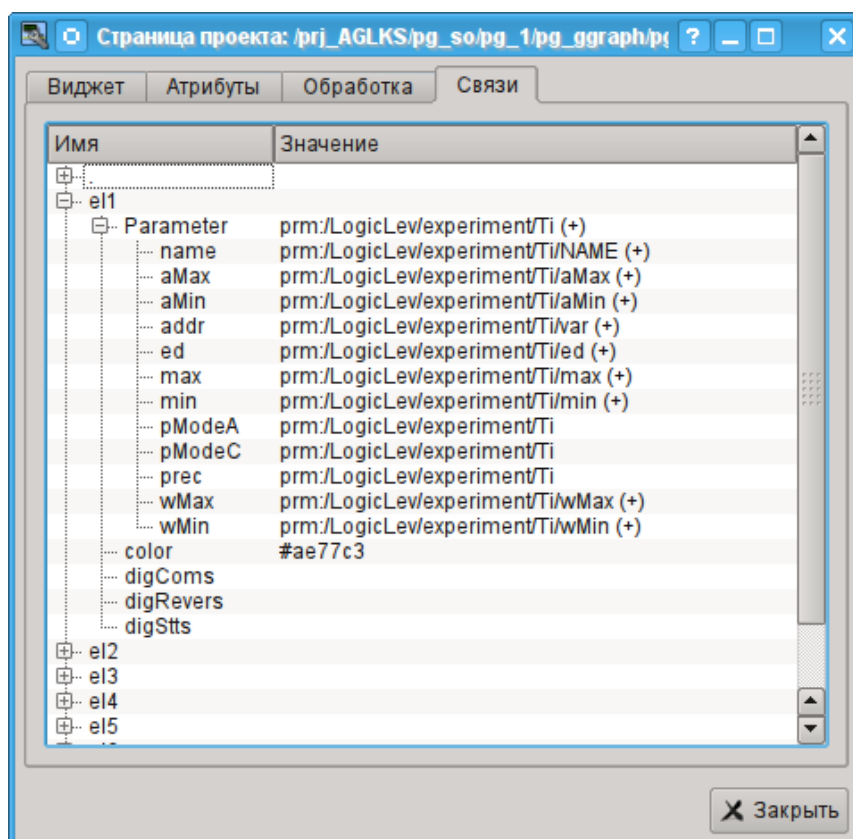


Рис.2.g. Вкладка связей диалога редактирования свойств визуального элемента.

2.1. Стили

Известно, что человек может иметь индивидуальные особенности в восприятии графической информации. Если эти особенности не учитывать то можно получить неприятие и отторжение пользователя к интерфейсу ВУ. Такое неприятие и отторжение может привести к фатальным ошибкам при управлении ТП, а также травмировать человека постоянной работой с таким интерфейсом. В SCADA системах приняты соглашения, которые регламентируют требования по созданию унифицированного интерфейса ВУ, нормально воспринимаемого большинством людей. При этом практически отсутствует учёт особенностей людей с некоторыми отклонениями.

С целью учесть это обстоятельство и предоставить возможность централизованно и просто изменять визуальные свойства интерфейса проектом предусматривается реализация менеджера стилей интерфейса визуализации.

Пользователем может быть создано множество стилей, каждый из которых будет хранить цветовые, шрифтовые и другие свойства элементов кадра. Простая смена стиля позволит быстро преобразить интерфейс ВУ, а возможность назначения индивидуальной стиля для пользователя позволит учесть его индивидуальные особенности.

Для реализации этой возможности, при создании кадров необходимо для свойств цвета, шрифта и других установить параметр «Конфигурация» (таблицы во вкладке «Обработка») в значение «Из стиля» (рис. 2.f). А в параметре «Конфигурационный шаблон» указать идентификатор поля стиля. Далее это поле автоматически появится в менеджере стилей и его можно будет там менять. Менеджер стилей доступен на странице конфигурации проекта во вкладке «Стили» (рис. 2.1). На этой вкладке можно создавать новые стили, удалять старые, изменять поля стиля и удалять ненужные.

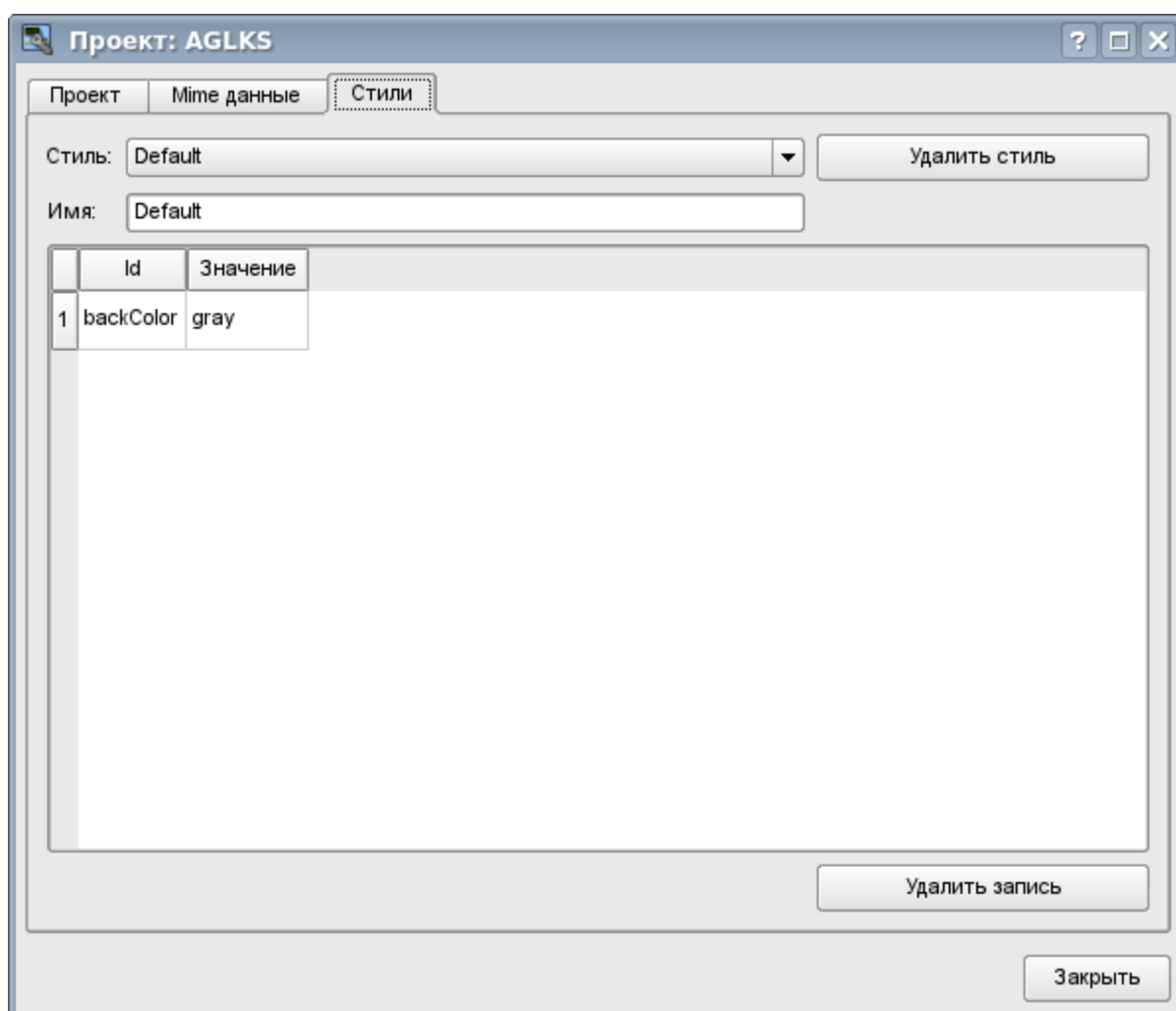


Рис. 2.1. Вкладка "Стили" страницы конфигурации проекта.

В целом стили доступны, начиная с уровня проектов. На уровне библиотек виджетов можно только определять поля стилей у виджетов. На уровне проекта при выборе стиля включается работа со стилями, что предполагает доступ к полям стилей вместо непосредственных значений атрибутов. Фактически это означает, что при чтении или записи атрибута виджета указанные операции будут осуществляться с соответствующим полем выбранного стиля.

При запуске проекта на исполнения будет использован установленный в проекте стиль. В последствии пользователь может выбрать стиль из перечня доступных. Выбранный пользователем стиль будет сохранён и использован при следующем запуске проекта.

2.2. Связывание с динамикой

Для предоставления актуальных данных в интерфейс визуализации должны использоваться данные подсистемы "Сбор данных (DAQ)". Природа этих данных следующая:

1. параметры, содержащие некоторое количество атрибутов;
2. атрибуты параметра могут предоставлять данные четырёх типов: Логический, Целый, Вещественный и Строковый;
3. атрибуты параметра могут иметь историю (архив);
4. атрибуты параметра могут быть для чтения, записи и с полным доступом.

Учитывая первый пункт, нужно обеспечить возможность группового назначения ссылки. Для этого используем концепцию логического уровня.

В соответствии с пунктом 2 связи обеспечивают прозрачное преобразование типов и не требуют специальной конфигурации.

Для удовлетворения возможности доступа к архивам в соответствии с пунктом 3 связи выполняют проверку типа атрибута и, в случае подключения к "Адресу", в значение помещается адрес связи.

В терминах СВУ динамические связи и конфигурация динамики являются одним процессом, для описания конфигурации которого предусматривается вкладка "Обработка" виджетов (рис.2.f). Вкладка содержит таблицу конфигурации свойств атрибутов виджета и текст процедуры вычисления виджета.

Кроме полей конфигурации атрибутов в таблице предусматривается колонка "Обработка" для избирательного использования атрибутов виджетов в вычислительной процедуре виджета и колонки "Конфигурация" и "Конфигурационный шаблон" для описания конфигурации связей.

Колонка "Конфигурация" позволяет указать тип связи для атрибута виджета:

- *Постоянная* - во вкладке связей виджета появляется поле указания постоянной, например особого цвета или заголовка для шаблонных кадров;
- *Входная связь* - связь с динамикой только для чтения;
- *Выходная связь* - связь с динамикой только для записи;
- *Полная связь* - полная связь с динамикой (чтение и запись).

Колонка "Конфигурационный шаблон" позволяет описать группы динамических атрибутов. Например, это могут быть разные типы параметров подсистемы "DAQ". Кроме того, при корректном формировании этого поля работает механизм автоматического назначения атрибутов при указании только параметра подсистемы "DAQ", что упрощает и ускоряет процесс конфигурации. Значение этой колонки имеет следующий формат: **<Параметр>|<Идентификатор>**, где:

- **<Параметр>** - группа атрибута;
- **<Идентификатор>** - идентификатор атрибута, именно это значение сопоставляется с атрибутами параметров DAQ при автоматическом связывании после указания групповой связи.

Установка связей может быть нескольких типов, который определяется префиксом:

- *val:* — Прямая загрузка значения через механизм связей. Например, связь: "val:100" загружает в атрибут виджета значение 100. Часто используется в случае отсутствия конечной точки связи с целью прямой установки значения.
- *prm:* — Связь на атрибут параметра или параметр в целом, для группы атрибутов, подсистемы "Сбор данных". Например, связь "prm:/LogicLev/experiment/Pi/var" осуществляет доступ атрибута виджета к атрибуту параметра подсистемы "Сбор данных". Знак "(+)", в конце адреса, сигнализирует об успешной линковке и присутствии целевого объекта.
- *wdg:* — Связь на атрибут другого виджета или виджет в целом, для группы атрибутов. Например, связь "wdg:/ses_AGLKS/pg_so/pg_1/pg_ggraph/pg_1/a_bordColor" осуществляет доступ атрибута одного виджета к атрибуту другого. На данный момент этот тип связи не

предназначен для установки пользователем вручную, а устанавливается автоматически в режиме динамического связывания!

Обработка связей происходит с периодичностью вычисления виджета в порядке:

- Получение данных входных связей.
- Выполнение вычисления скрипта.
- Передача значений по выходным связям.

На рис. 2.g представлена вкладка связей с возможностью группового и индивидуального назначением атрибутов.

При размещении виджета, содержащего конфигурацию связей, в контейнер виджетов все связи исходного виджета добавляются в список результирующих связей контейнера виджетов.

Из вышесказанного видно, что связи устанавливаются пользователем в процессе конфигурации интерфейса. Однако, для предоставления возможности создания кадров общего назначения с функцией предоставления детализированных данных разных источников одного типа необходим механизм динамической установки связей. Такой механизм предусматривается посредством зарезервированного ключевого идентификатора "<page>" группы атрибутов связей у кадров общего назначения и динамическое назначение связей с идентификатором "<page>" в процессе открытия кадра общего назначения сигналом от другого виджета.

Рассмотрим пример, когда имеется кадр общего назначения "Панель контроля графиком" и множество "Графиков" на разных кадрах. "Панель контроля графиком" имеет связи с шаблонами:

- tSek -> "<page>|tSek"
- tSize -> "<page>|tSize"
- trcPer -> "<page>|trcPer"
- valArch -> "<page>|valArch"

При этом каждый виджет "График" имеет атрибуты tSek, tSize, trcPer и valArch. В случае вызова сигнала открытия "Панели контроля графиком" из любого виджета "График" происходит связывания атрибутов "Панели контроля графиком" в соответствии атрибуту, указанному в шаблоне, с атрибутом виджета "График". Как результат, все изменения на "Панели контроля графиком" будут отражаться на графике посредством связи.

В случае наличия у виджета "График" внешних связей на параметры подсистемы "Сбор данных", связи "Панели контроля графиком" будут устанавливаться на внешний источник. Кроме этого, если у "Панели контроля графиком" будут заявлены связи на отсутствующие непосредственно у виджета "График" атрибуты, то будет производиться поиск на наличие таких атрибутов у внешнего источника, первого на который установлена прямая связь, выполняя тем самым дополнение недостающих связей.

Для наглядного изображения этого механизма приведена таблица 2.2.

Таблица 2.2. Механизм динамической линковки.

Атрибуты "Панели контроля графиком" (шаблон динамической связи)	Атрибуты "Графика"	Атрибуты внешнего "Параметра"	Результирующая связь или значение связываемого атрибута
tSek (<page> tSek)	tSek	-	"График".tSek
tSize (<page> tSize)	tSize	-	"График".tSize
trcPer (<page> trcPer)	trcPer	-	"График".trcPer
valArch (<page> valArch)	valArch	-	"График".valArch
var (<page> var)	var	var	"Параметр".var
ed (<page> ed)	-	ed	"Параметр".ed
max (<page> max)	-	-	EVAL
min (<page> min)	-	-	EVAL

3. Исполнение интерфейсов СВУ

Исполнение интерфейса СВУ заключается в запуске нового сеанса проекта или подключении 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.3.

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 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 сетевой обмен в случае доступа к движку СВУ через сеть.

Иерархически модулем предусматривается возможность размещения страниц проекта как на главном окне исполнения (рис.3), 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 одном разрешении экрана, на разных разрешениях.

Главное окно состоит из меню (сверху) строки статуса (снизу) и исполняемого содержимого сеанса между ними. Меню в режиме исполнения позиционируется как инструмент администратора OpenSCADA, содержащий общесистемные функции, и доступно только привилегированным пользователям, состоящим в группе "root". 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.

С правой стороны строки статуса размещаются индикаторы часов, текущей станции движка СВУ и пользователя, от имени которого выполняется интерфейс СВУ, а также панель с кнопками квитации нарушений, печати и экспорта. По двойному клику на индикаторе пользователя можно пользователя сменить, введя новое имя и пароль пользователя, а по клику на кнопки квитации — сквитировать нарушения полностью или только нужное уведомление. В главное поле строки статуса выводятся различные информационные сообщения и сообщения помощи.

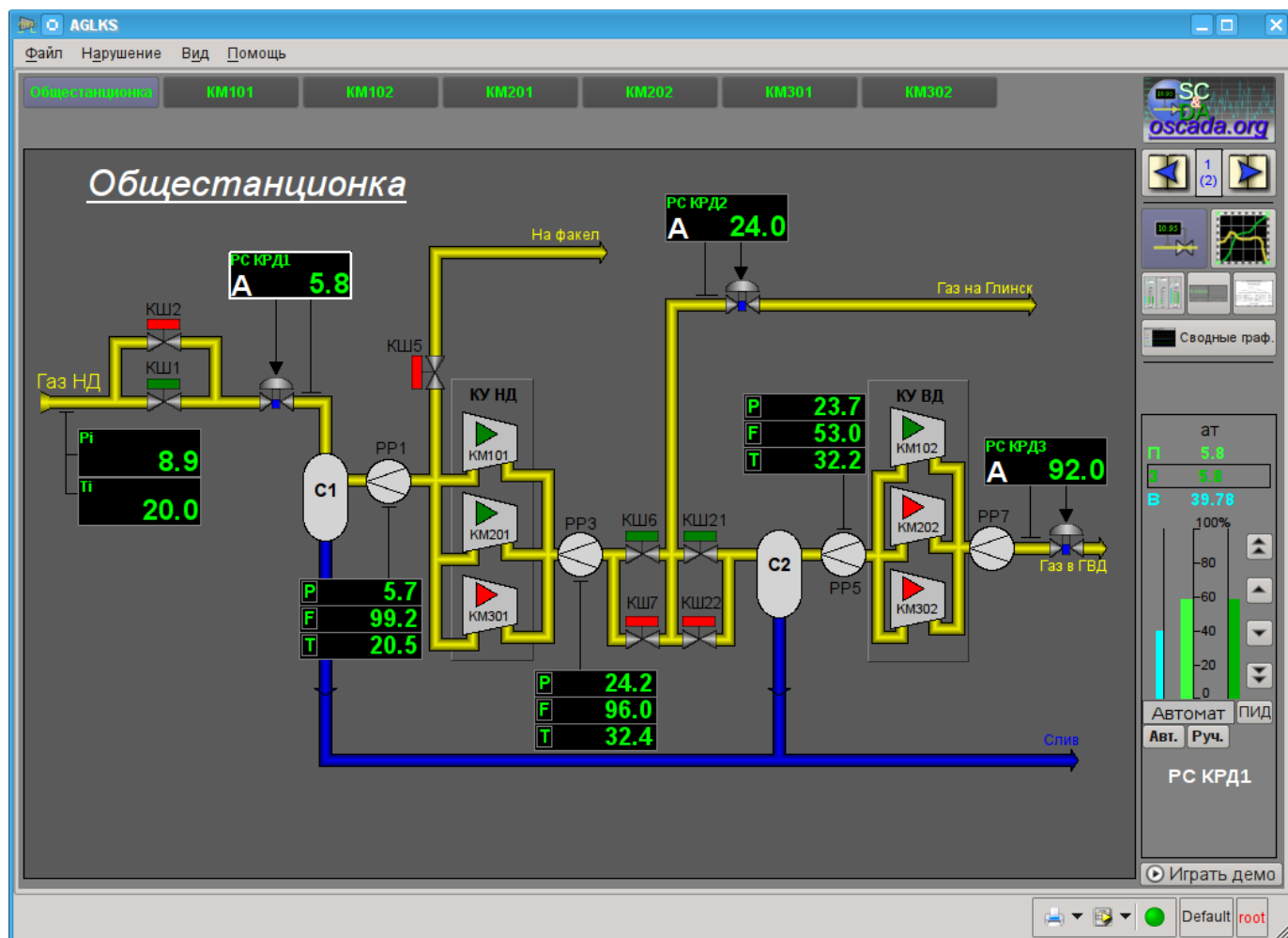


Рис.3. Главное окно режима исполнения.

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 вышеперечисленных 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 полноэкранный, 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 CBY.
Box	Контейнер	Содержит механизм размещения других виджетов с целью формирования новых, более сложных виджетов и страниц конечной визуализации.

Более детально рассмотрим реализацию каждого примитива.

4.1. Прimitives 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. 4.1 представлена часть экрана с кадром, содержащим элементарные фигуры.

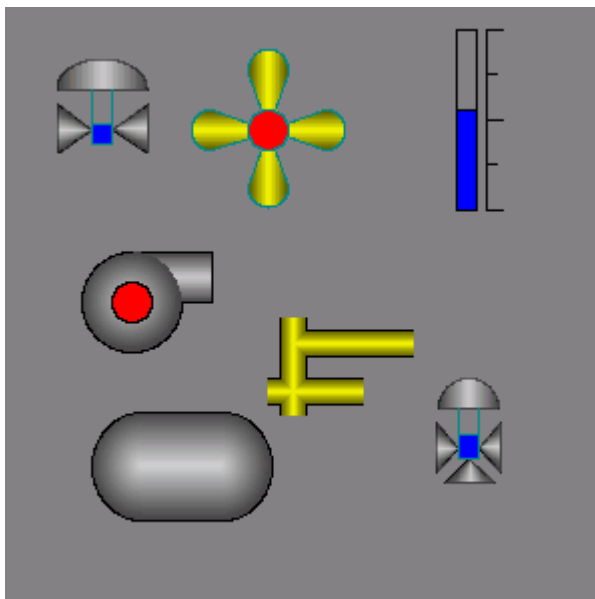


Рис.4.1. Реализация элементарных фигур в Vision.

Фигуры, лежащие в основе данного виджета, содержат точки(начальная и конечная), которые могут стыковаться с соответствующими 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, кликнув 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 Connecions(Привязки)) либо при помощи выделения мышкой.

Связать фигуры друг с другом можно следующим образом:

1. Нажать кнопку Connections.
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, две дуги не могут быть соединены непосредственно друг с другом).

Залить замкнутый контур из фигур можно следующим образом:

1. Нажать кнопку Connections.
2. Создать замкнутый контур.
3. Два раза клацнуть мышкой внутри его.

Удалить заливку замкнутого контура можно из контекстного меню, клацнув правой кнопкой манипулятора "мышь" по заливке; разорвав контур заливки; двойным кликом левой кнопки манипулятора "мышь" по уже имеющемуся залитому контуру.

Поворот фигуры осуществляется вокруг центра виджета.

4.2. Примитив 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. 4.2 представлена часть экрана с кадром, содержащим примеры текста с использованием различных параметров.



Рис.4.2. 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.

Прогресс-бар - Полоска прогресс-бара.

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

На рис. 4.3 представлена часть экрана с кадром, содержащим вышеперечисленные элементы формы.

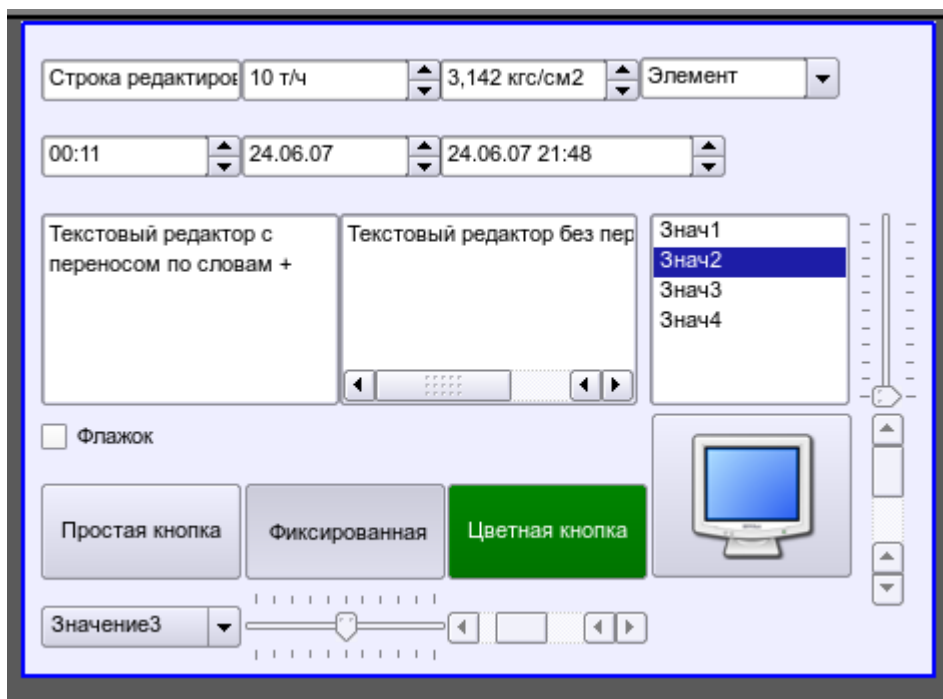


Рис.4.3. Реализация элементов формы в Vision.

4.4. Прimitив отображения медиа-материалов (Media)

Реализована поддержка элемента отображения медиа-материалов со свойствами:

- Указание источника медиа данных (изображения или видео-материала).
- Просмотр изображений большинства известных форматов с возможностью их вписывания в размер виджета.
- Проигрывание простых анимированных форматов изображений и видео с возможностью управления скоростью проигрывания.
- Проигрывание полноформатного видео и аудио, посредством [Phonon](#).
- 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. 4.4 представлена часть экрана с кадром, содержащим примеры просмотра/проигрывания медиа-данных.

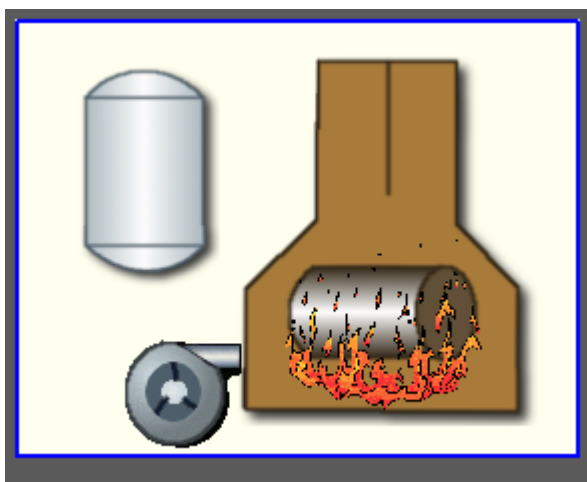


Рис.4.4. Реализация базового элемента отображения медиа-материалов в Vision.

4.5. ПрIMITИВ 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 данных, 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. 4.5 представлена часть экрана с кадром, содержащим примеры диаграммы-тренда.

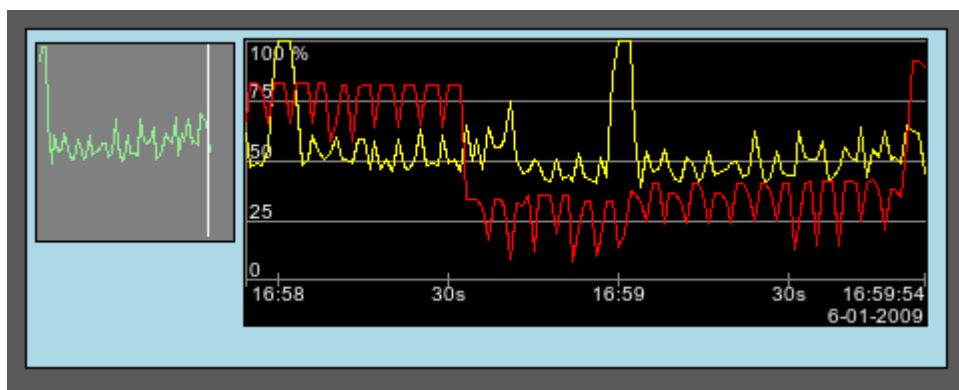


Рис.4.5. Реализация базового элемента отображения диаграммы-тренда в Vision.

4.6. ПрIMITИВ 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 слежения за появлением сообщений в архиве сообщений.

На рис. 4.6 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.4.6. Реализация базового элемента отображения протоколов в Vision.

4.7. Примитив 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 формировании 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 OpenSCADA 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 слинкованными атрибутами могут извлекаться значения из истории, архива.

На рис. 4.7 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.м3	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.4.7. 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. 2.а, а на рис. 3 изображена страница, содержащая контейнер страниц, построенный на основе данного примитива.

5. Векторный графический редактор.

5.1. Назначение

Наличие собственного векторного графического редактора является неотъемлемой частью уважающей себя SCADA системы. Опыт показывает, что наиболее информативной формой представления технологических процессов являются мнемосхемы - совокупность сигнальных устройств, изображений оборудования и внутренних связей контролируемого объекта, выполняемые на персональном компьютере. Для их создания можно использовать любой из существующих графических редакторов. Однако, полученные таким образом мнемосхемы являются статическими и не отражают динамику изменения характеристик процесса, а следовательно, они неадекватны и неудобны для восприятия. Таким образом, одной из задач, стоящих перед разработчиками SCADA систем, является создание графического редактора для изображения объектов, характеристики которых могут быть динамически изменены.

5.2. Принципы и функции разработанного графического редактора

Основой описываемого редактора являются три графических примитива: линия, дуга, кривая Безье. К динамически изменяющимся характеристикам этих примитивов относятся:

- Координаты контрольных точек; используются для задания формы линии, дуги или кривой Безье. При этом линия имеет — 2 контрольные точки, дуга — 5 контрольных точек, кривая Безье — 4 (рис. 5.2.а).
- Ширина линии.
- Ширина бордюра (границы).
- Цвет бордюра (границы).
- Стилль линии (сплошная, пунктирная, точечная).

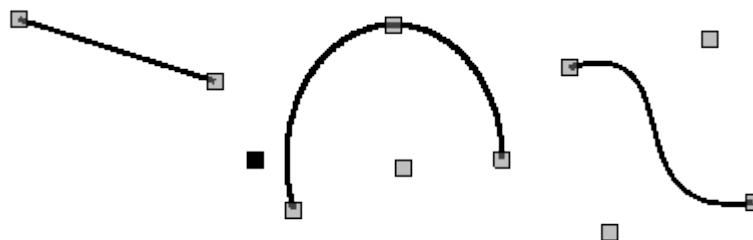


Рис. 5.2.а. Контрольные точки линии, дуги и кривой Безье.

Примеры примитивов различного цвета, толщины, стилей с бордюрами и без бордюров приведены на рис. 5.2.б.

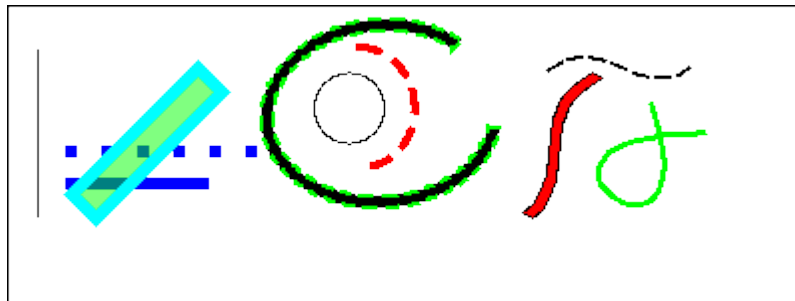


Рис. 5.2.б. Примеры примитивов: линия, дуга и кривая Безье слева направо.

Предусмотрена возможность связи различных графических примитивов для создания сложных графических объектов. Если связанные примитивы образуют замкнутый контур, то он может быть залит цветом и/или изображением (рис. 5.2.с).

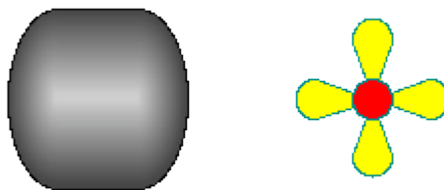


Рис. 5.2.с. Заливки замкнутого контура цветом и/или изображениями.

Графический редактор позволяет масштабировать и поворачивать фигуры (примитивы и сложные графические объекты) (рис. 5.2.d). К возможностям редактора также относятся выделение, перемещение, копирование и удаление фигур.

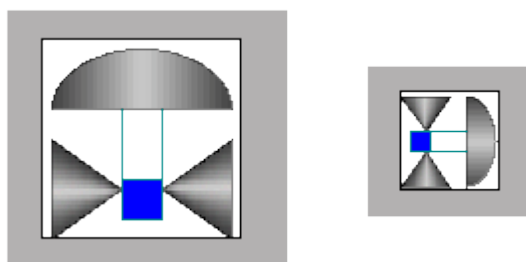




Рис. 5.2.d. Масштабирование и поворот фигур.

5.3. Основные принципы работы в описываемом графическом редакторе

Для того, чтобы приступить к работе с графическим редактором после запуска проекта OpenSCADA необходимо вызвать «Рабочий пользовательский интерфейс». Во вкладке «Виджет» содержится перечень существующих графических библиотек и принадлежащих им элементов. Предположим, что нам необходимо добавить графический элемент в одну из существующих

библиотек. Для этого выделим имя библиотеки и нажмем кнопку  на панели инструментов. В появившееся окно введем идентификатор и имя нового графического элемента. После чего перейдем к этому элементу и нажмем кнопку . Справа появится поле для рисования. С помощью двойного щелчка мышью по этому полю либо при помощи контекстного меню войдем в режим редактирования — режим работы графического редактора, позволяющий осуществлять все предусмотренные манипуляции с фигурами. В тот момент, когда вновь созданный нами графический элемент находится в фокусе, вкладка «Атрибуты» приобретет вид, представленный на рис. 5.3.а.

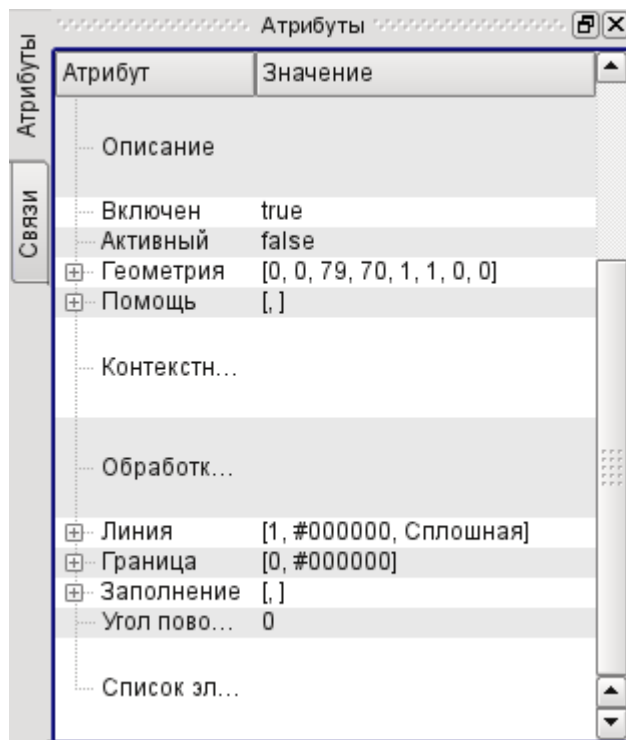


Рис. 5.3.а. Атрибуты элементарной фигуры.

С помощью мыши или поля «Геометрия» вкладки «Атрибуты» зададим размеры поля для рисования и масштабные коэффициенты.

Воспользовавшись полем «Линия» вкладки «Атрибуты», зададим ширину, цвет, стиль линий, которыми будем рисовать. С помощью поля «Граница» зададим ширину и цвет бордюра. Поле «Заполнение» позволяет задать цвет и изображение заливки. «Список элементов» содержит перечень примитивов, используемых при создании графического объекта. Элементы всех полей вкладки «Атрибуты» могут динамически изменяться в скриптах (программах) пользователя. Графические примитивы могут быть нарисованы с помощью мышки или заданы с помощью списка графических примитивов («Список элементов»). В первом случае координаты контрольных точек создаваемого примитива вычисляются автоматически, ширина, цвет, стиль линии, ширина и цвет бордюра устанавливаются по умолчанию из вкладки «Атрибуты». Во втором случае примитив необходимо описать в «Списке элементов» следующим образом:

```
line: (x1|y1): (x2|y2): width: color: border_width: border_color: style (1)
arc: (x1|y1): (x2|y2): (x3|y3): (x4|y4): (x5|y5): width: color: border_width: border_color: style (2)
bezier: (x1|y1): (x2|y2): (x3|y3): (x4|y4): width: color: border_width: border_color: style, (3)
```

Где:

(x1|y1) — координаты первой контрольной точки примитива;
 (x2|y2) — координаты второй контрольной точки примитива;
 (x3|y3) — координаты третьей контрольной точки примитива;
 (x4|y4) — координаты четвертой контрольной точки примитива;
 (x5|y5) — координаты пятой контрольной точки примитива;
 width — ширина линии, которой будет нарисован примитив;
 color — цвет линии, которой будет нарисован примитив;
 border_width — ширина бордюра;
 border_color — цвет бордюра;
 style — стиль линии ("0" — сплошная, "1" — пунктир, "2" — точечная).

При этом приоритетными являются значения ширины, цвета, стиля линии, ширины и цвета бордюра, заданные в «Списке элементов». Если необходимо использовать какие-либо характеристики, устанавливаемые полями «Линия» или «Граница» вкладки «Атрибуты», то при описании примитива их следует пропустить. Например, мы хотим создать линию с шириной 3 красного цвета сплошную и без бордюра. При этом в поле «Линия» заданы ширина линии, равная 3, черный цвет линии, сплошной стиль линии; в поле «Граница» заданы ширина бордюра, равная 5,

зеленый цвет. Тогда описание примитива в «Списке элементов» должно выглядеть следующим образом:

```
line:(x1|y1):(x2|y2)::red:0:: (4)
```

Формы записи (1) - (4) задают статические характеристики примитивов, которые не могут быть изменены программно. Для задания динамических характеристик необходимо использовать форму записи:

```
line:1:2:w1:c1:w2:c2:s1 (5)
```

Тогда в окно «Атрибуты» добавятся поля: Точка 1 (1), Точка 2 (2), Ширина 1 (w1), Ширина 2 (w2), Цвет 1 (c1), Цвет 2 (c2), Стил ь 1 (s1) (рис.5.3.b) Значения этих полей можно изменять программно, с помощью одного из языков программирования проекта OpenSCADA. Очевидно, что при желании динамическими могут быть объявлены не все свойства примитива, а одно или несколько, кроме того одно динамическое свойство может быть использовано многократно.

Редактирование координат контрольных точек осуществляется либо с помощью мыши, либо путем их изменения в «Списке элементов» для статических точек или прямо во вкладке «Атрибуты», имея в наличии динамические точки («Точка 1»...). Редактирование других характеристик осуществляется с помощью изменения содержимого полей «Линия» , «Граница», «Заполнение» вкладки «Атрибуты» или «Списка элементов».

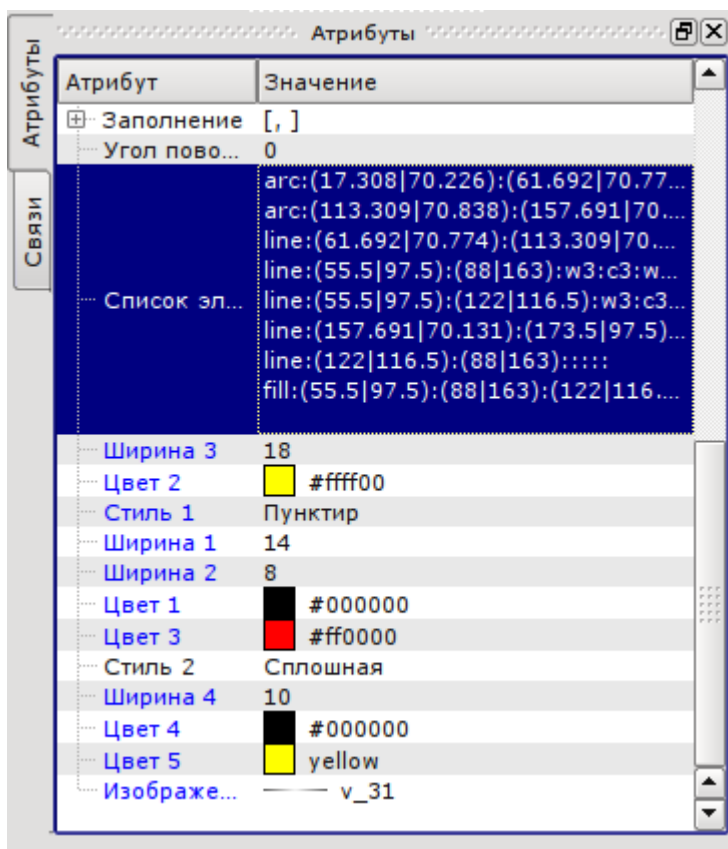


Рис. 5.3.b. Динамические атрибуты элементарной фигуры.

Предусмотрены возможности выделения (щелчок левой кнопкой мыши по фигуре), совместного выделения примитивов (левая кнопка мыши + нажатая клавиша «Ctrl» при выключенной кнопке «Подключиться»), совместного выделения рамкой, рисуемой с помощью нажатой левой кнопки мышки, их перемещения (клавиатура/мышь), копирования и вставки («Ctrl»+«C», «Ctrl»+«V» или из контекстного меню), удаления («Del»).

Для связи примитивов друг с другом необходимо:

- 1) нажать кнопку «Подключиться»('Connections');
- 2) выделить одну из фигур и переместить её начальную или конечную точку к желаемой начальной или конечной точке другой фигуры;

- 3) отпустить левую кнопку мышки, когда появится полупрозрачный кружок вокруг точек.

Для заливки замкнутого контура необходимо два раза кликнуть мышкой внутри него либо задать заливку в «Списке элементов» следующим образом:

fill:(x1:y1):(x2:y2):...:(xn:yn):color:image — (статика);

fill:1:2:3:...:c1:i1 — (динамика).

Где:

(x1:y1):(x2:y2):...:(xn:yn) - координаты начальных/конечных точек примитивов, образующих замкнутый контур;

color - цвет заливки;

image - изображение заливки.

При этом приоритетными являются значения цвета и изображения заливки, заданные в «Списке элементов». Если необходимо использовать характеристики, устанавливаемые полем «Заполнение» вкладки «Атрибуты», то при описании примитива их следует пропустить.

Для масштабирования фигур необходимо в поле «Геометрия» задать значения масштаба по осям «X» и «Y». Кроме того, существует возможность визуально увеличить/уменьшить изображение, не меняя масштаб. Для этого необходимо выйти из режима редактирования, щелкнуть правой кнопкой мыши по полю виджета, в появившемся контекстном меню выбрать пункт «Увеличить (+10%)»/ «Уменьшить (-10%)» либо покрутить колесико мышки, удерживая при этом нажатой клавишу «Ctrl». Выход из режима редактирования осуществляется либо с помощью нажатия клавиши «Esc», либо с помощью контекстного меню виджета.

Существует возможность поворачивать содержимое виджета. Для поворота созданных объектов следует в окне «Атрибуты» задать «Угол поворота» от -360 до 360.

Графический редактор поддерживает прозрачность цвета, задаваемую следующим образом: *color-t*, где *color* - цвет, а *t* - прозрачность от 0 (полностью прозрачный) до 255 (непрозрачный).

5.4. Контекстное меню графического редактора

Предусмотрено контекстное меню, вызываемое щелчком правой кнопки мышки в области редактируемого виджета. В зависимости от того, какой объект находится под указателем мышки, контекстное меню может принимать несколько различных вид. Ниже, на рисунке 5.4 приведены примеры контекстного меню.

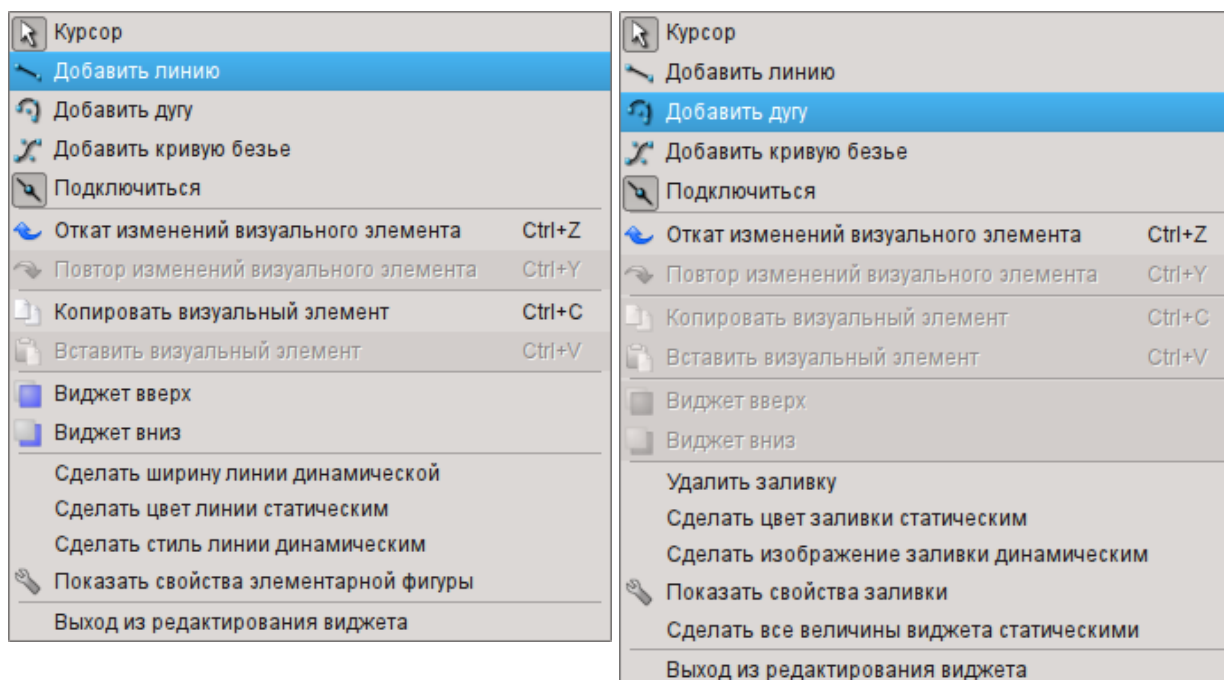


Рис. 5.4. Контекстное меню для линий (линия, дуга, кривая Безье) и для заливки слева направо.

Контекстное меню включает следующие секции (сверху вниз):

- рисования, позволяющая выбирать фигуру для рисования, выбирать режим «привязок» при помощи пункта «Подключиться», а также возвращаться в режим выбора — «Курсор»;
- отката/повтора изменений;
- копирования и вставки выбранных фигур в рамках одного виджета;
- поднятия, опускания выбранных фигур по оси «Z»;
- управления (сделать статическими или динамическими) различными атрибутами примитивов, а также вызова диалога свойств элементарной фигуры;
- выхода из режима редактирования.

5.5. Диалог свойств элементарной фигуры

Диалог, варианты которого представлены на рисунках 5.5.a, 5.5.b, 5.5.c, реализован для интерактивного и удобного управления свойствами фигур(ы).

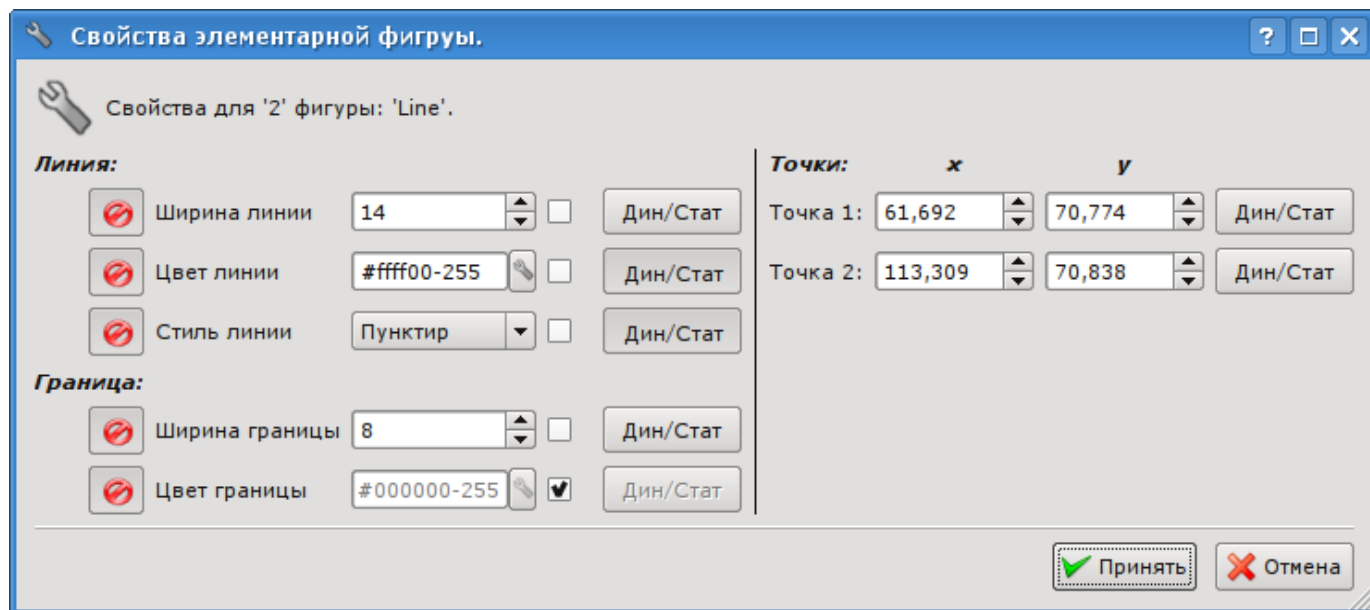
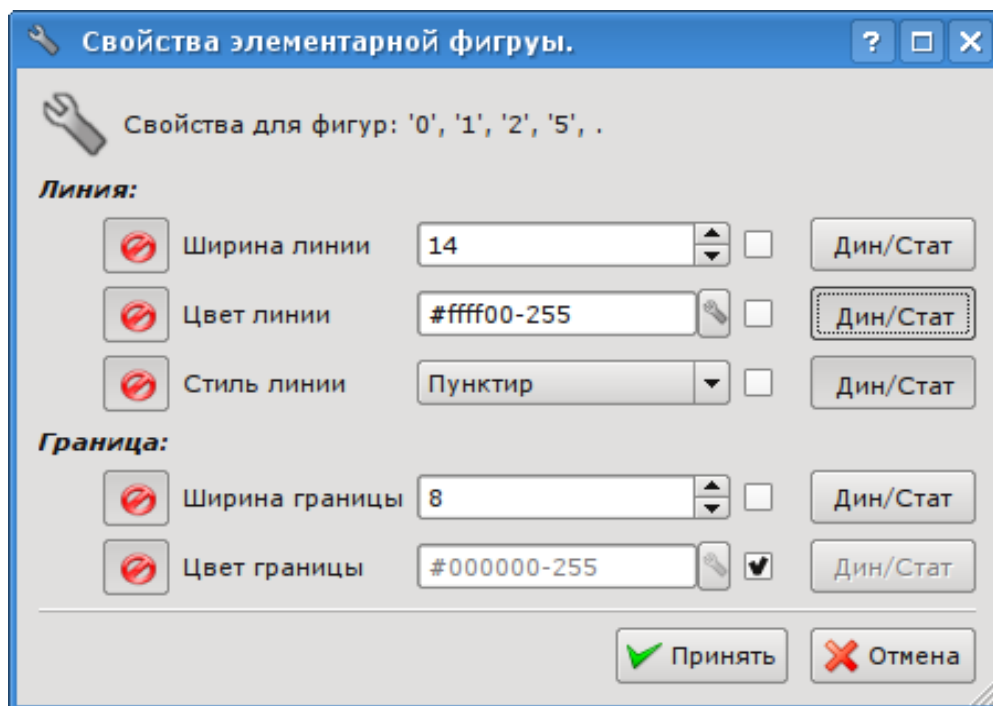


Рис. 5.5.a. Диалог свойств элементарной фигуры для одной линии (линия, дуга или кривая Безье).



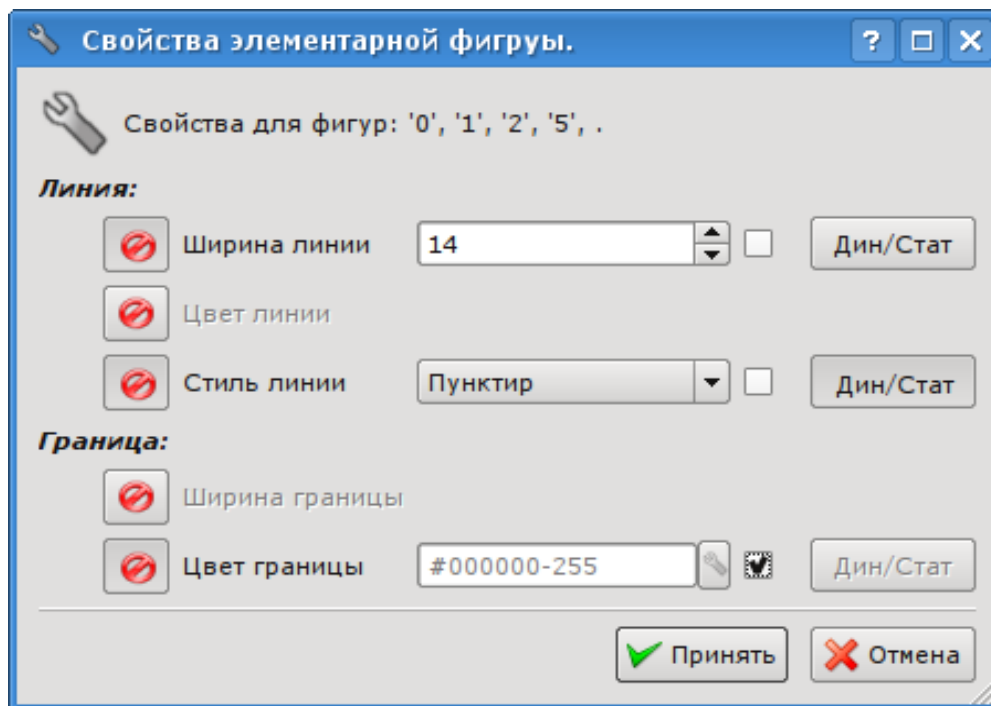


Рис. 5.5.b. Диалог свойств элементарной фигуры для группы выбранных линий.

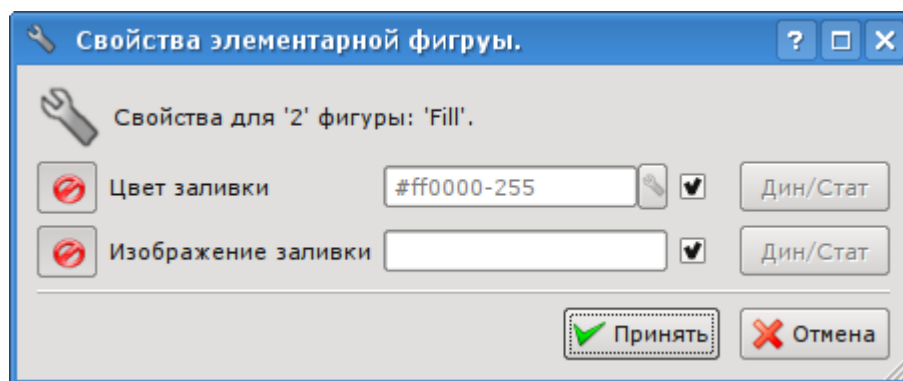



Рис. 5.5.c. Диалог свойств элементарной фигуры для заливки.

Как видно из рисунков выше, диалог может иметь три различных формы в зависимости от объекта, для которого вызывается. В заголовке диалога перечислены номера фигур, для которых он вызван. Эти номера соответствуют позиции фигуры в «Списке элементов», начиная сверху.

Если диалог вызван для одиночной фигуры (линия, дуга или кривая Безье), то в нем существует возможность редактировать координаты точек фигуры (рис. 5.5.a). Если точка соединена с точкой другой фигуры или фигур, и включен режим «привязок» («Подключиться»), то введенные координаты будут применены ко всем фигурам, для которых эта точка общая, а заливки, в которых участвует фигура, будут соответственно перерисованы.

Если же диалог вызван для группы выбранных фигур (рис. 5.5.b), то свойства (атрибуты), указанные в диалоге, будут применены ко всем фигурам, перечисленным в заголовке диалога. При вызове диалога в полях свойств будут отображаться данные той фигуры из группы выбранных, относительно которой было вызвано контекстное меню. Существует возможность включать/исключать отдельные свойства диалога. Для этого предусмотрена кнопка (). В случае исключения отдельных свойств они не будут обрабатываться при подтверждении диалога (кнопка «Принять»). При подтверждении диалога все указанные данные для включенных свойств будут применены для всей группы фигур.

Диалог для свойств заливки (рис. 5.5.c) позволяет управлять свойствами отдельной заливки.

При выборе флажка справа от свойства оно (свойство) после подтверждения диалога устанавливается в значение по умолчанию, которое указано во вкладке «Атрибуты». Кнопки «Дин/Стат» делают соответствующие свойства динамическими либо статическими.

6. Общая конфигурация модуля

Для настройки собственного поведения в неочевидных ситуациях модулем предоставляется возможность настройки отдельных параметров посредством интерфейса управления OpenSCADA (рис. 6.а). Таковыми параметрами являются:

- Состояние модуля: исполнение и доступное количество экранов в библиотеке QT4.
- Имя удалённой OpenSCADA станции с движком визуализации CBY.
- Начальный пользователь конфигуриатора — указывает, от имени какого пользователя открывать конфигуриатор без запроса пароля.
- Время жизни страниц в кеше. Визуализатор осуществляет ускорения работы пользовательского интерфейса путём помещения ранее открытых страниц в кеш, для контроля времени хранения страниц в котором данное свойство и предназначено. Указание нулевого значения вообще исключит очистку кеша.
- Перечень проектов, через символ разделителя ';', для запуска их автоматического исполнения с запуском модуля. Для предоставления возможности указания открытия окна исполнения проекта на нужном дисплее, много дисплейных систем, предусмотрен формат записи имени проекта "PrjName-1", где 1 — номер целевого дисплея.
- Ссылка на страницу конфигурации перечня внешних OpenSCADA станций.

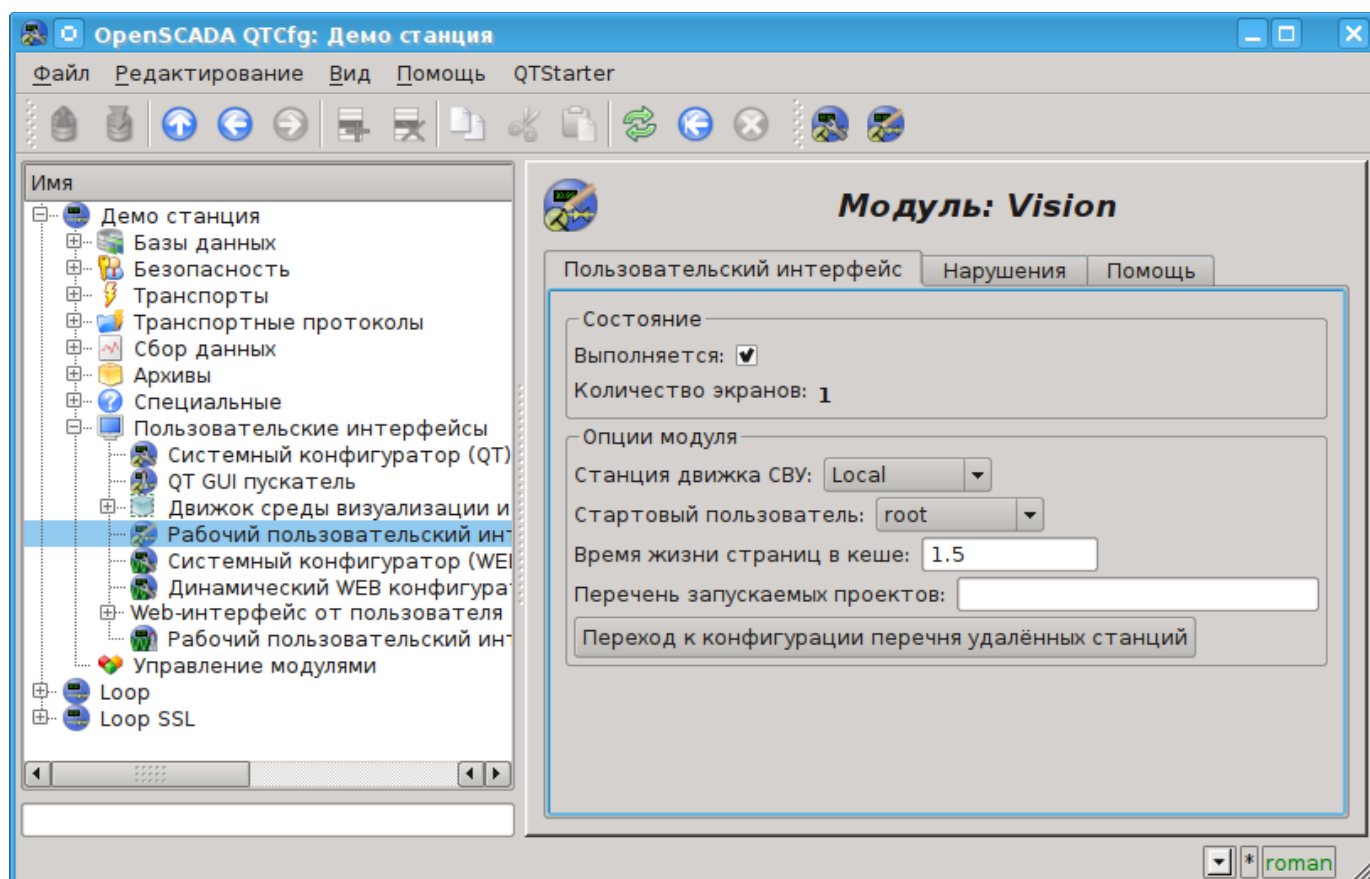


Рис.6.а. Страница конфигурации модуля.

Настройку обработки нарушений можно осуществить во вкладке "Нарушения" (рис.6.b). В этой вкладке можно указать строку команды воспроизведения звуковых файлов, обычно это "play -q %f".

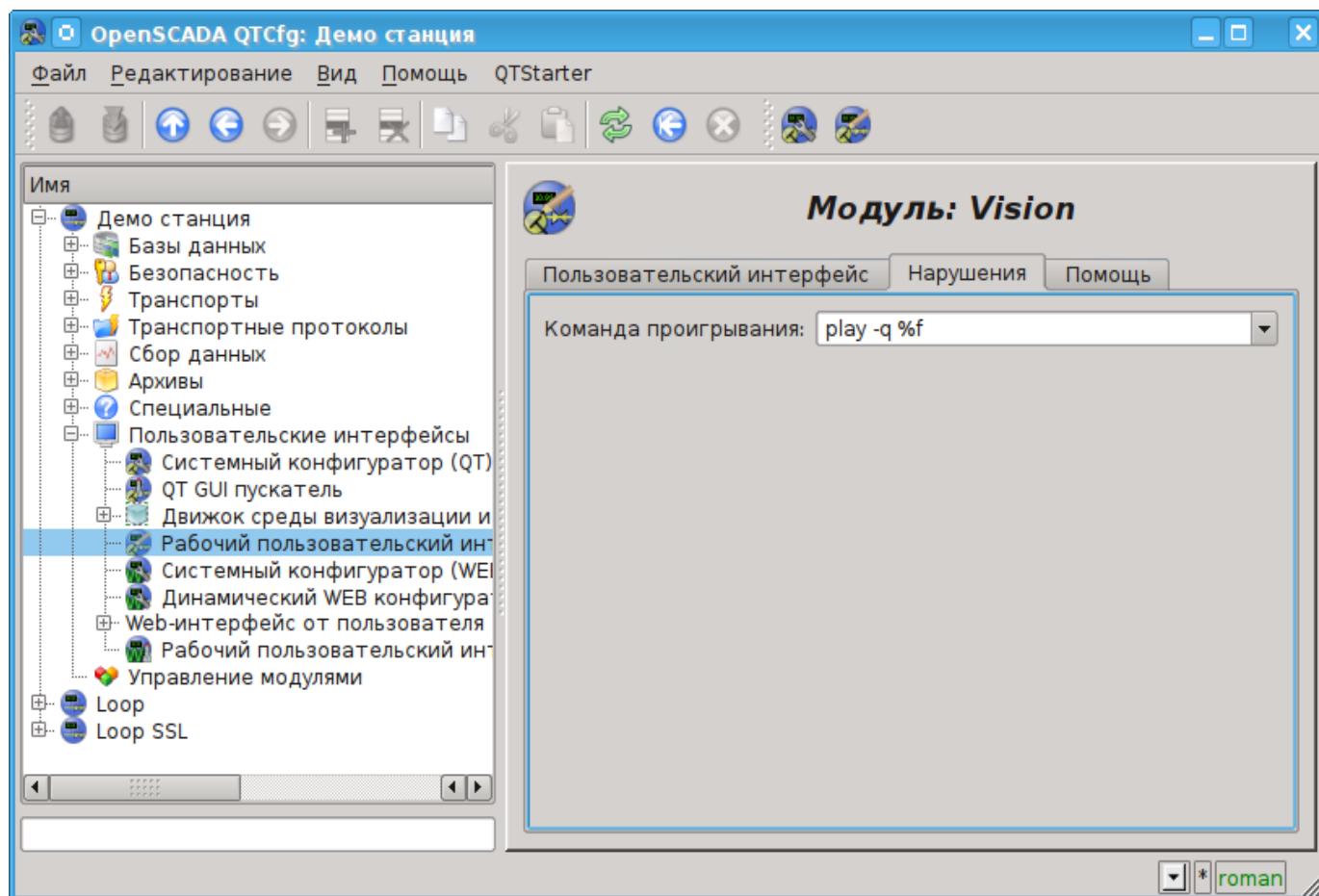


Рис.6.b. Вкладка "Нарушения", страницы конфигурации модуля.