

# Модуль подсистемы “Пользовательские интерфейсы” <WebVision>

Модуль:	WebVision
Имя:	Рабочий пользовательский интерфейс (WEB)
Тип:	Пользовательские интерфейсы
Источник:	ui_WebVision.so
Версия:	0.9.1
Автор:	Роман Савоченко
Разработчик:	Роман Савоченко, Максим Лысенко, Ксения Яшина
Описание:	Web визуальный рабочий пользовательский интерфейс для исполнения проектов среды визуализации и управления (СВУ).
Лицензия:	GPL

## Оглавление

Модуль подсистемы “Пользовательские интерфейсы” <WebVision>	1
Введение	2
1. Назначение	2
2. Исполнение интерфейсов СВУ	4
3. Представление базовых элементов (примитивов)	6
3.1. Примитив элементарная фигура (ElFigure)	7
3.2. Примитив текста (Text)	8
3.3. Примитив элементов формы (FormEl)	9
3.4. Примитив отображения медиа-материалов (Media)	10
3.5. Примитив построения диаграмм/графиков (Diagram)	10
3.6. Примитив формирования протокола (Protocol)	11
3.7. Примитив формирования отчётной документации (Document)	11
3.8. Примитив контейнера (Box)	12
4. Общая конфигурация модуля	13
Заключение	14

# Введение

Модуль WebVision предоставляет механизм конечной визуализации среды визуализации и управления (CBU) в систему OpenSCADA. Модуль основан на WEB технологиях (XHTML, JavaScript, CSS, AJAX). В своей работе модуль использует данные движка CBU (модуль [VCAEngine](#)).

Среда визуализации и управления (CBU) 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 ситуациях и условиях могут применяться CBU, 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 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 CBU в том или ином практическом случае. Например, технологии вроде библиотеки QT позволяют создавать высокореактивные CBU, 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. С другой стороны 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 такой ситуации независимое создание реализаций CBU на различной основе может повлечь за собой невозможность использования данных конфигурации одной CBU 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 CBU, основан [проект создания концепции CBU](#). Результатом этого проекта и стал данный модуль непосредственной визуализации на основе WEB технологий, модуль непосредственной визуализации [Vision](#) и движок CBU [VCAEngine](#).

## 1. Назначение

Данный модуль непосредственной визуализации CBU предназначен только для исполнения интерфейсов CBU в среде WEB-технологий.

Интерфейс пользователя формируется в WEB-браузере путём обращения к WEB-серверу и получения от него XHTML-документа по протоколу HTTP. В данном случае в роли WEB-сервера выступает система OpenSCADA, которая поддерживает стандартные коммуникационные механизмы TCP-сетей (модуль Transport.Sockets), протокол передачи гипертекста (модуль Protocol.HTTP), а также шифрование трафика между браузером и сервером (Transport.SSL). Исходя из этого, для получения доступа к интерфейсу пользователя, предоставляемого этим модулем, необходимо в OpenSCADA настроить транспорт (Transport.Sockets или Transport.SSL) в связке с протоколом HTTP (Protocol.HTTP). В поставке с системой OpenSCADA идут конфигурационные файлы, содержащие настройки Transport.Sockets для портов 10002 и 10004. Следовательно, интерфейс модуля в конфигурации OpenSCADA по умолчанию будет доступен по URL: <http://localhost:10002> или <http://localhost:10004>.

В финальной версии 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 вторичного 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 последующим 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. Исполнение интерфейсов СВУ

Исполнение интерфейса СВУ заключается в запуске нового сеанса проекта или подключения к существующему на уровне движка СВУ (рис.2). Перед запросом на подключение к сеансу выполняется запрос на аутентификацию пользователя. Далее модуль непосредственной визуализации отражает и управляет данными сеанса. Главное окно режима исполнения данного модуля имеет вид, представленный на рис.3.

Интерфейс окна исполнения полностью строится динамически скриптом JavaScript, исходя из содержимого сеанса проекта путём прямых XML запросов к серверу.

Реализовано обновление содержимого открытых страниц интерфейса визуализации с периодичностью одна секунда. 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 изменёнными данными.

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 WEB-браузера (рис.3), так и вкладывая внутрь виджетов контейнеров.

Рис.1. Страница аутентификации.

# OpenSCADA. Рабочий пользовательский интерфейс (WEB)

Подключение к открытой сессии

[tmpISO](#)

Создание новой сессии для доступного проекта

[AGLKS](#)

Рис.2. Подключение или создание нового сеанса исполнения проекта СВУ.

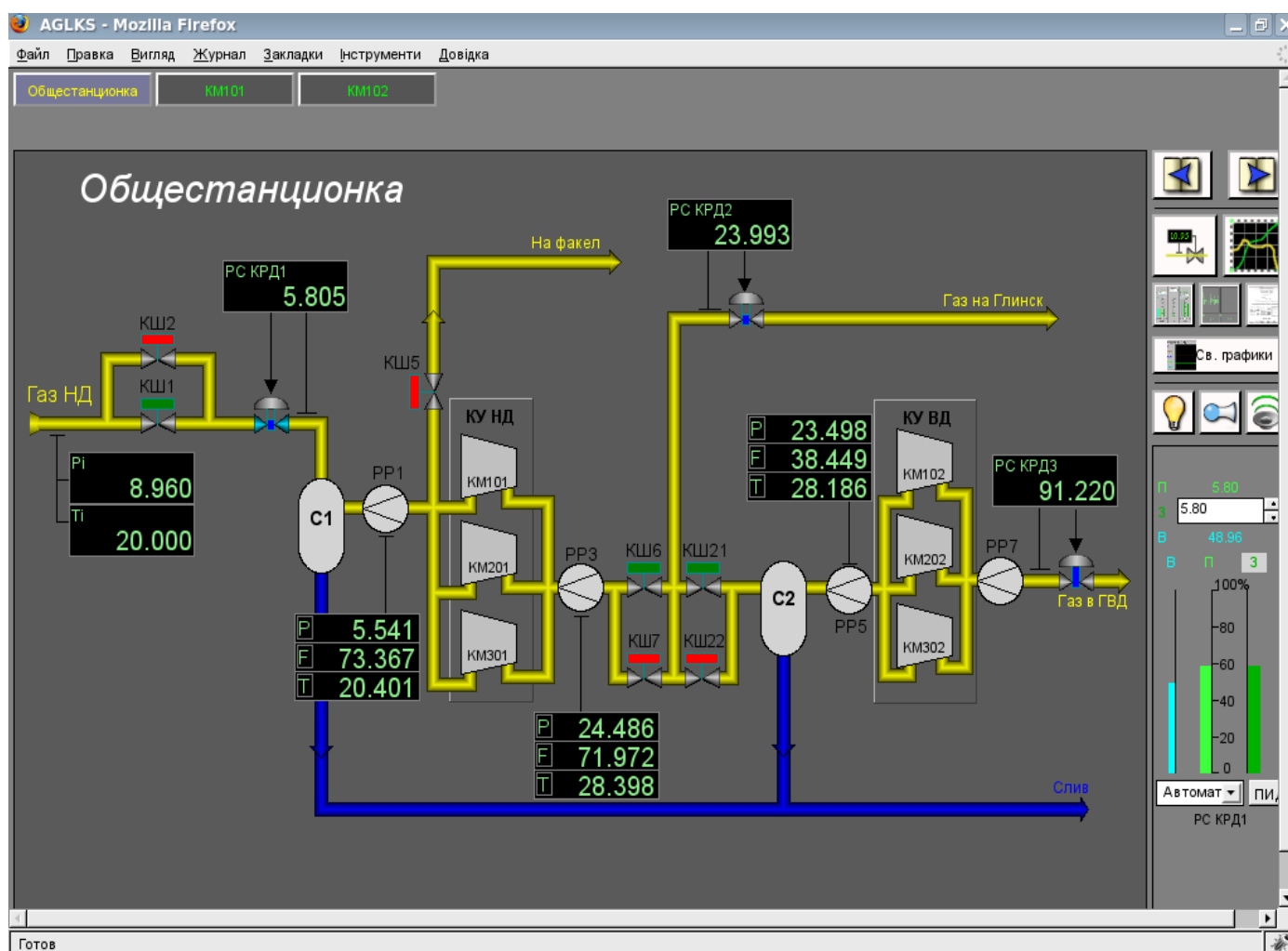


Рис.3. Главное окно режима исполнения.

### 3. 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"> <li>• Линия.</li> <li>• Дуга.</li> <li>• Кривая безье.</li> <li>• Заливка замкнутого пространства.</li> </ul> <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"> <li>• Редактирование строки.</li> <li>• Редактирование текста.</li> <li>• Флажок.</li> <li>• Кнопка.</li> <li>• Поле выбора из списка.</li> <li>• Список.</li> <li>• Слайдер.</li> <li>• Строка прокрутки.</li> </ul>
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 СВУ.
Box	Контейнер	Содержит механизм размещения других виджетов с целью формирования новых, более сложных виджетов и страниц конечной визуализации.

Более детально рассмотрим реализацию каждого примитива.

### 3.1. Примитив элементарная фигура (ElFigure)

Реализована поддержка элементарных фигур: линии, эллиптической дуги, кривой Безье и заливка замкнутых контуров цветом и 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 представлена часть экрана с кадром, содержащим вышеперечисленные элементарные фигуры.

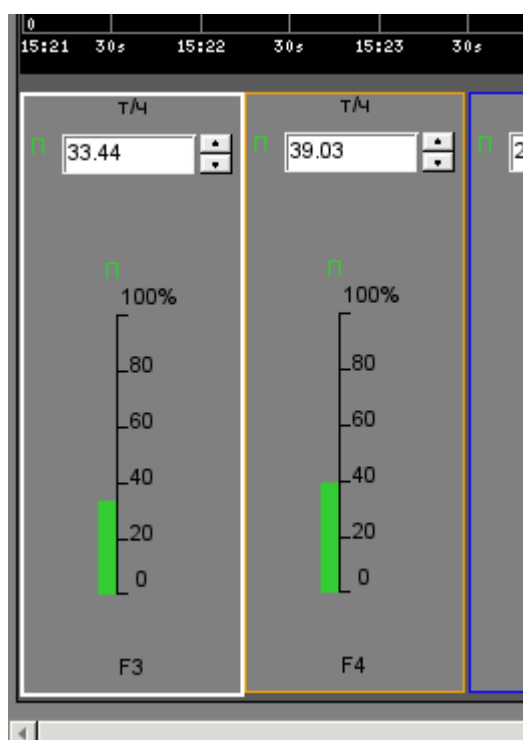


Рис.4 Реализация элементарных фигур в WebVision.

### 3.2. 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. 5 представлена часть экрана с кадром, содержащим примеры текста с использованием различных параметров.

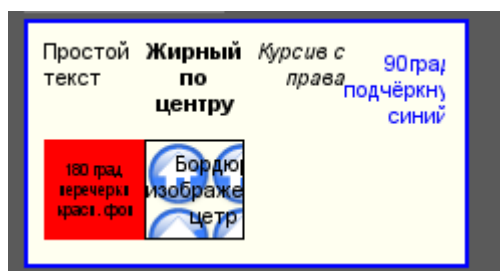


Рис.5. Реализация базового элемента текста в WebVision.



### 3.3. 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».

*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 (не реализован).

*Прогресс-бар* — Полоска прогресс-бара (не реализован).

Реализованы режимы: «Включен» и «Активен», а также передача изменений и событий в модель данных СВУ (движок). Для всех реализованных представлений поддерживается активный режим, т.е. элементы могут быть использованы для создания форм пользовательского ввода.

На рис. 6 представлена часть экрана с кадром, содержащим вышеперечисленные элементы формы.

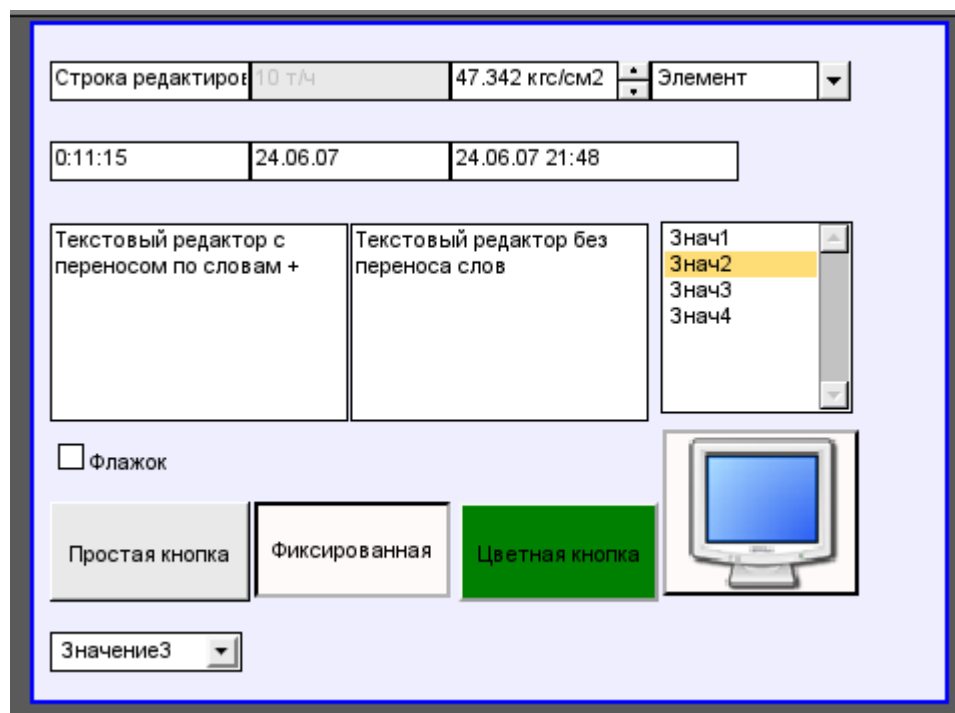


Рис.6. Реализация элементов формы в WebVision.

### 3.4. Примитив 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.
- 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. 7 представлена часть экрана с кадром, содержащим примеры просмотра/проигрывания медиа-данных.

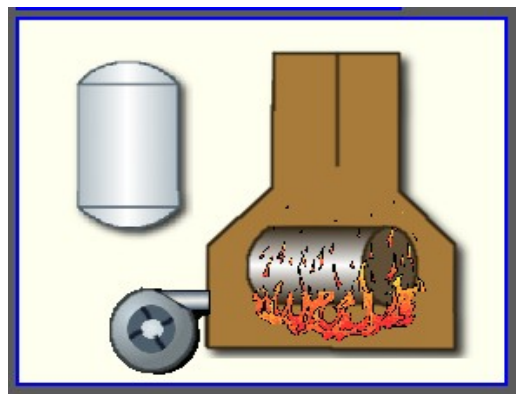


Рис.7. Реализация базового элемента отображения медиа-материалов в WebVision.

### 3.5. Примитив построения диаграмм/графиков (Diagram)

Реализована поддержка элемента построения диаграмм/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.
  - Возможность установки курсора тренда мышью.

На рис. 8 представлена часть экрана с кадром, содержащим примеры диаграммы-тренда.

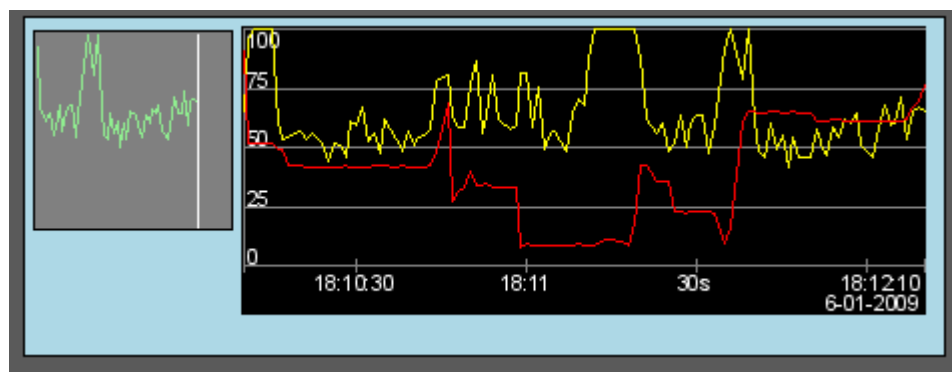


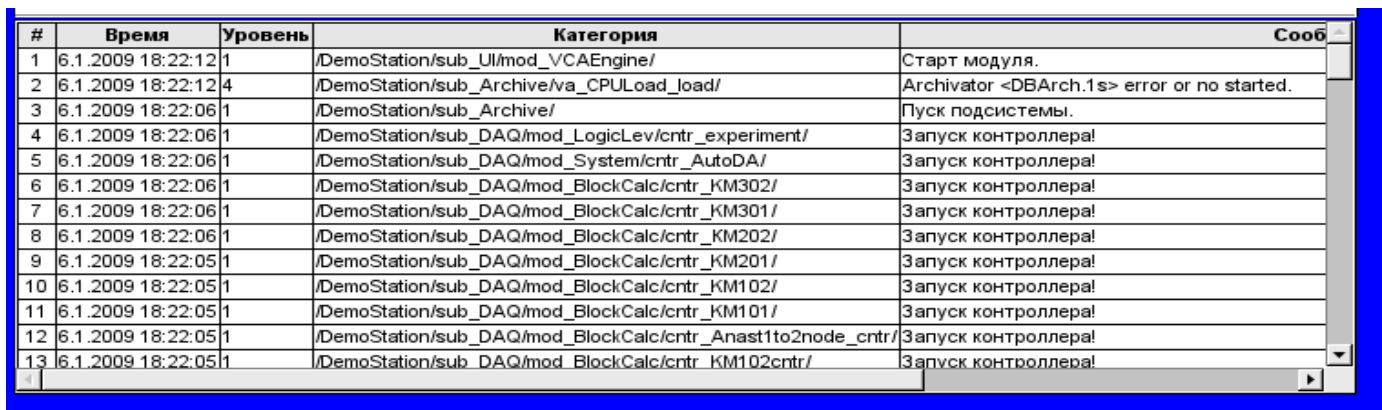
Рис.8. Реализация базового элемента отображения диаграммы-тренда в WebVision.

### 3.6. 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. 9 представлена часть экрана с кадром, содержащим пример протокола.



#	Время	Уровень	Категория	Сообщение
1	6.1.2009 18:22:12	1	/DemoStation/sub_UI/mod_VCAEngine/	Старт модуля.
2	6.1.2009 18:22:12	4	/DemoStation/sub_Archive/va_CPULoad_load/	Archivator <DBArch.1s> error or no started.
3	6.1.2009 18:22:06	1	/DemoStation/sub_Archive/	Пуск подсистемы.
4	6.1.2009 18:22:06	1	/DemoStation/sub_DAO/mod_LogicLev/cntr_experiment/	Запуск контроллера!
5	6.1.2009 18:22:06	1	/DemoStation/sub_DAO/mod_System/cntr_AutoDA/	Запуск контроллера!
6	6.1.2009 18:22:06	1	/DemoStation/sub_DAO/mod_BlockCalc/cntr_KM302/	Запуск контроллера!
7	6.1.2009 18:22:06	1	/DemoStation/sub_DAO/mod_BlockCalc/cntr_KM301/	Запуск контроллера!
8	6.1.2009 18:22:06	1	/DemoStation/sub_DAO/mod_BlockCalc/cntr_KM202/	Запуск контроллера!
9	6.1.2009 18:22:05	1	/DemoStation/sub_DAO/mod_BlockCalc/cntr_KM201/	Запуск контроллера!
10	6.1.2009 18:22:05	1	/DemoStation/sub_DAO/mod_BlockCalc/cntr_KM102/	Запуск контроллера!
11	6.1.2009 18:22:05	1	/DemoStation/sub_DAO/mod_BlockCalc/cntr_KM101/	Запуск контроллера!
12	6.1.2009 18:22:05	1	/DemoStation/sub_DAO/mod_BlockCalc/cntr_Anast1to2node_cntr/	Запуск контроллера!
13	6.1.2009 18:22:05	1	/DemoStation/sub_DAO/mod_BlockCalc/cntr_KM102cntr/	Запуск контроллера!

Рис.9 Реализация базового элемента отображения протоколов в WebVision.

### 3.7. 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 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 ссылками на атрибуты могут извлекаться значения из истории, архива.

На рис. 10 представлен кадр, содержащий пример документа.

SO9	SO10	SO11	SO12	SO13	SO14	SO15	SO16
-----	------	------	------	------	------	------	------

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 <b>18 20</b>	Тип отбора <b>Угловой</b>	Атм. давление, кПа <b>95</b>
Мол. доля N <sub>2</sub> , % <b>70</b>	К-т шероховатости <b>0.2</b>	Отсечка, кПа <b>80</b>
Мол. доля CO <sub>2</sub> , % <b>10</b>	К-т притупления <b>0.1</b>	Верхн. предел ПД, кПа <b>150</b>
Диаметр трубы, мм <b>100</b>	К-т L (альфа) <b>0.3</b>	Порог переключ., кПа <b>85</b>
Диаметр СУ, мм <b>60</b>	Отн. площадь СУ <b>0.6</b>	Дин. вязкость, кгс/м <sup>2</sup> <b>32</b>

**Часовые данные**

Дата	Время		Объём, тыс.м <sup>3</sup>	Ср. разность давления, кПа	Среднее давление, МПа	Средняя температура, °С	Средняя плотность, кг/м <sup>3</sup>
	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

Рис.10 Реализация базового элемента отображения отчётной документации в WebVision.

### 3.8. 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.

## 4. Общая конфигурация модуля

Для настройки собственного поведения, в не очевидных ситуациях, модулем предоставляется возможность настройки отдельных параметров посредством интерфейса управления OpenSCADA (рис. 11). Таковыми параметрами являются:

- Время жизни сессии аутентификации.

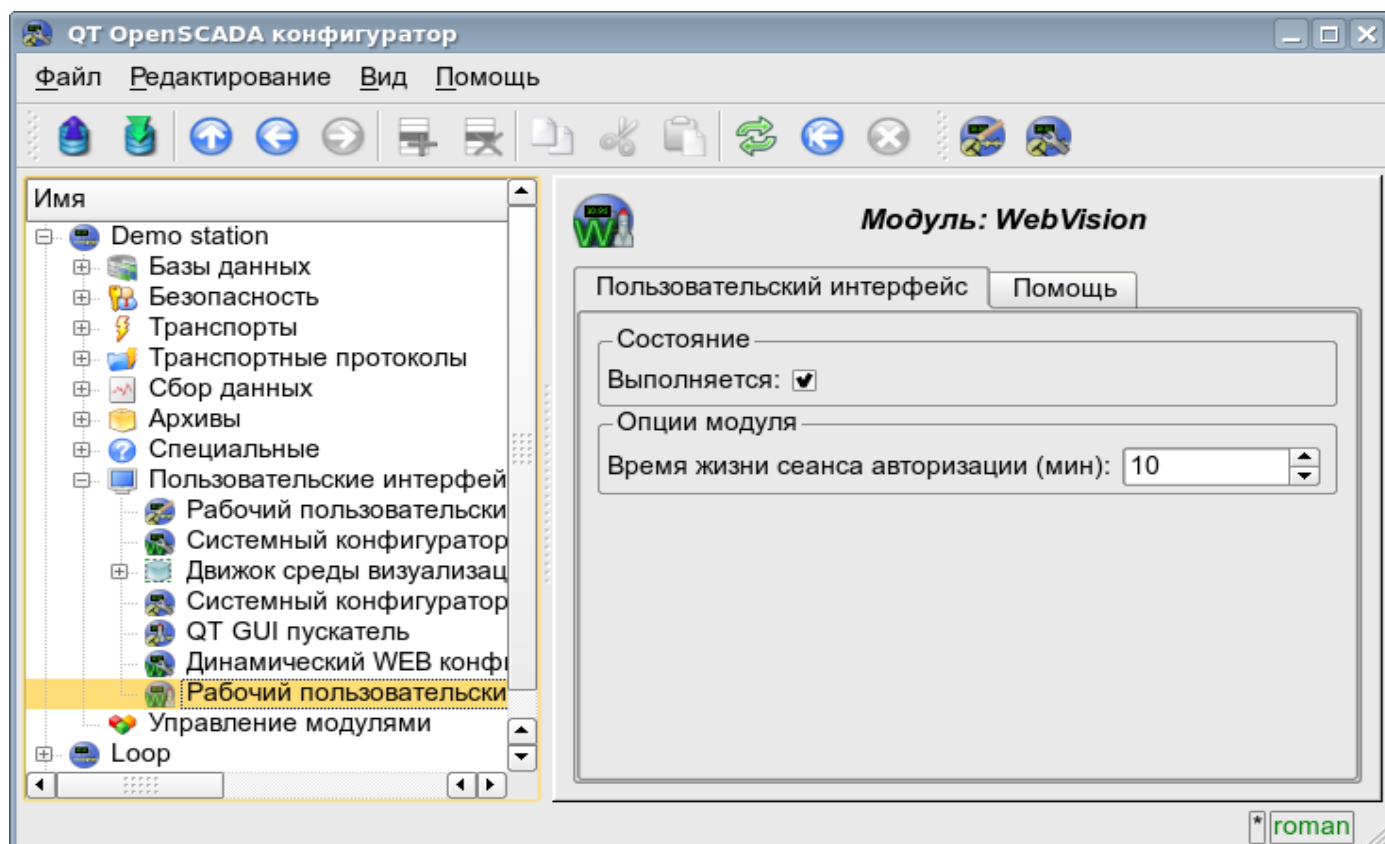


Рис.11. Страница конфигурации модуля.

## **Заключение**

На данном этапе модуль может быть использован для построения реальных интерфейсов пользователя с поддержкой основных функций. Однако отдельные проблемы могут возникать как по причине недоработанности, так и различия браузеров. Сейчас достаточно качественно обеспечена работоспособность на браузерах: FireFox, Konqueror, Opera и Google Chromium.