

The background of the advertisement features two men in light blue business suits. The man on the left is smiling and looking towards the man on the right. They are both looking at a tablet held by the man on the right. The tablet screen displays a complex SCADA (Supervisory Control and Data Acquisition) interface with various charts, graphs, and data points. The overall scene is set against a blurred background of a modern office or industrial environment.

SIEMENS

*Ingenuity for Life**

WinCC

Open Architecture

Открытая SCADA - платформа для
создания OEM - решений

По-настоящему модульная архитектура, гибкая концепция инжиниринга на базе объектно-ориентированного подхода и открытого API, развитый инструментарий среды разработки, широкая поддержка открытых стандартов и передовых технологий – таковы ключевые составляющие успеха при создании систем управления и диспетчеризации на базе флагманской SCADA-платформы компании Siemens – WinCC OA. Благодаря схеме OEM-лицензирования WinCC OA теперь у Вас есть возможность выпуска и распространения Ваших продуктов/решений, построенных на базе WinCC OA и ориентированных на нишевые/отраслевые/корпоративные приложения – под собственной торговой маркой.

www.siemens.com/wincc

Инжиниринг проектов на базе SCADA-системы SIMATIC WinCC OA

SIEMENS

Рассматривается подход к разработке проектов на базе SCADA-системы WinCC OA. Приводится обзор основных инструментальных средств для инжиниринга проектов, имеющихся в арсенале WinCC OA.

ООО «Сименс», г. Москва

Эффективность применения программных SCADA-пакетов при построении систем сбора, обработки и визуализации данных в приложениях диспетчерского контроля и автоматизации технологических процессов, от которой в конечном счете зависит эффективность решения пользовательских бизнес-задач, включает такие составляющие, как функциональность, надежность, безопасность, возможность сопровождения средств базовой платформы, а также особенности реализации, технические характеристики и удобство эксплуатации собственно прикладного решения.

Не последнюю роль в перечне факторов, определяющих выбор программной SCADA-платформы для построения конкретной системы, играет эффективность и удобство инструментальных средств инжиниринга, а также сама философия процесса разработки прикладных проектов. Особенно критичными данные факторы становятся при разработке и сопровождении больших иерархических систем, трудоемкость создания которых может составлять десятки человеко-месяцев, а цена ошибки, обнаруженной на этапе коммерческой эксплуатации решения, может оказаться слишком высокой для бизнеса заказчика.

Рассмотрим подход к разработке проектов на базе SCADA-системы WinCC OA [1, 2]. Также в статье будет сделан обзор основных инструментальных средств для инжини-

ринга проектов, имеющихся в арсенале WinCC OA.

Философия инжиниринга проектов WinCC OA: объектно ориентированный подход, информационная модель и эффективные инструментальные средства

Инжиниринг проектов в системе WinCC OA основан на объектно ориентированном подходе, реализация которого обеспечивается информационной моделью проекта, встроенными средствами программирования и инструментарием среды разработки.

Ключевым элементом информационной модели проекта WinCC OA является понятие точки данных (Data Point) – совокупности переменных, характеризующих образ процесса. Для каждого элемента точки данных (Data Point Element) могут быть определены конфигурационные элементы (Configs), описывающие его свойства и действия в его отношении, такие как обработка сигналов (например, сглаживание или задание диапазонов значений), связь с внешними системами, архивирование, формирование алармов и др. Точки данных могут создаваться по определяемому пользователем шаблону типа данных (Data Point Type). Тип точки данных фактически является аналогом класса в объектно ориентированном программировании. Точки данных, которые могут быть созданы на основании этого типа, наследуют его структуру и свойства (в том чис-

ле конфигурационные элементы – «методы»), что аналогично созданию экземпляра объекта определенного класса в объектно ориентированном программировании. При этом WinCC OA поддерживает как линейную модель структуры данных, так и иерархическую.

Объектно ориентированный подход к построению информационной модели системы естественным образом сочетается с концепцией событийно ориентированного программирования с использованием встроенного языка CONTROL (CTRL). При этом сценарии (скрипты), написанные на языке CTRL, могут представлять собой как обработчики событий, связанных с элементами графического интерфейса, так и процедуры (в том числе сложные) обработки данных (сценарии общего назначения).

Практическое воплощение объектно ориентированного подхода при разработке масштабных SCADA-проектов было бы невыполнимо без развитого инструментария среды разработки. Такой инструментарий в составе WinCC OA реализован в виде конфигурационного редактора PARA и графического редактора GEDI, включающих редактор модели данных, средства массового параметрирования и конфигурирования точек данных, интерактивный редактор и отладчик сценариев, написанных на языке CTRL. Также в составе WinCC OA имеются инструментальные средства, обеспечивающие организацию

и эффективность процесса инжиниринга – средства администрирования проекта, интерфейс к системам управления версиями, встроенный симулятор, интерфейс прикладного программирования (API), средства работы с базами данных и др.

Развитый инжиниринговый инструментарий, естественным образом повышающий эффективность разработки любого проекта, абсолютно необходим в случае создания распределенных систем, обрабатывающих десятки, сотни тысяч точек данных и более, – целевых проектов для WinCC OA. При этом на повестку дня, помимо собственно вопросов эффективного массового инжиниринга, выходит также весь комплекс вопросов, характерных для проектов создания комплексных информационных систем в сфере классического ИТ (информационная структура системы, характеристики каналов связи и топология сети, информационная безопасность и т. п.).

SCADA-система WinCC OA располагает развитыми средствами построения пользовательского интерфейса, такими как библиотеки графических элементов, отвечающие современным тенденциям в области HMI (трехмерные объекты, анимация и пр.), средства визуализации трендов и алармов, расширенные редакторы свойств, средства управления иерархией панелей, а также обес-

печивает возможность поддержки визуального программирования, создания пользовательских интерфейсов для устройств на базе iOS и Android, создания многоязычных проектов и др.

Важным аспектом являются возможности платформы WinCC OA в части обмена данными с внешними системами, резервирования, построения распределенных архитектур.

В то же время разнообразные инструментальные средства WinCC OA, предоставляющие широкие возможности при инжиниринге систем, требуют от разработчика высочайшей инженерной дисциплины и ответственного выбора проектных решений, обеспечивающих надежное функционирование создаваемой системы на всех этапах жизненного цикла.

Рассмотрим подробнее процесс создания проекта в WinCC OA и возможности основных инжиниринговых средств и инструментов системы.

Процесс создания проекта

Процесс создания проекта WinCC OA сочетает этапы, характерные для традиционного подхода к разработке прикладных SCADA-проектов, и шаги, напрямую связанные с эффективным инжинирингом масштабных систем с использованием объектно ориентированной методологии. Типовой процесс создания проекта WinCC OA показан на рис. 1.

Отправной точкой при создании проекта в WinCC OA является разработка информационной модели будущей системы, включающей иерархию и структуры данных, алармов и их свойств. Основой для формирования информационной модели может служить (в зависимости от назначения и масштаба системы) техническая структура объектов, включенных в SCADA-систему (аналогия «элемент – устройство – агрегат – ...»), организационная модель объекта, например иерархия диспетчерских пунктов (местный уровень – региональный уровень – центральный уровень), или их сочетание. Информационная модель выражается в виде совокупности типов точек данных и созданных на их основе структур для хранения конкретных переменных процесса.

Следующим этапом является определение графических элементов, которые служат для визуализации созданных ранее информационных объектов системы. Данный шаг включает как формирование собственно графических символов для последующего использования при реализации операторских интерфейсов, так и их «привязку» к переменным процесса (элементам точек данных). Также на данном этапе должны быть подготовлены шаблоны конфигурационных элементов, детальных представлений, диалоговые окна, шаблоны трендов и др.

Результаты первых двух этапов обеспечивают необходимую базу для выполнения массового инжиниринга: на этом этапе с помощью специального инструментария системы WinCC OA осуществляется импорт точек данных из внешних источников или единой базы данных, применяемой в процессе инжиниринга (например, системы Comos). При использовании же более традиционного для SCADA-систем способа визуального инжиниринга – создания графических объектов путем их перетаскивания (drag & drop) на панели – соответствующие им структуры данных формируются автоматически.

Создание операторских интерфейсов завершается разработкой мнемосхем. На этом этапе разработчик получает в свое распоряжение мощные библиотеки стандартных объектов и широкие воз-



Рис. 1. Типовой процесс создания проекта WinCC OA

возможности графического редактора GEDI. В качестве элементов графического интерфейса вместе с созданными на предшествующих этапах объектами на базе стандартных библиотек могут также использоваться графические объекты, разработанные самим пользователем, — виджеты (External Widget Objects).

Заключительным шагом создания проекта является его локализация, то есть перевод всех элементов проекта, содержащих текстовую информацию для человека-оператора, на язык целевой системы. Локализация может быть выполнена с применением встроенной утилиты WinCC OA, обеспечивающей возможность извлечения текстов из готового прикладного проекта. В дальнейшем данные тексты доступны в формате ASCII. Для поддержки процесса локализации в составе WinCC OA имеется функция словаря. WinCC OA обеспечивает поддержку одновременно до 32 различных языков в проекте (используется кодировка Unicode UTF-8).

Отметим, что описанная в настоящем разделе технология создания проекта системы WinCC OA охватывает только непосредственно фазу реализации программного решения и не отменяет необходимости выполнения других этапов типового цикла разработки прикладного ПО (определение требований, проектирование, тестирование, документирование). Для построения крупномасштабных систем на базе WinCC OA требуется неукоснительно соблюдать весь порядок проектирования и разработки. Это является необходимым условием и фактором успеха при реализации проекта, его внедрении и промышленной эксплуатации.

Информационная модель: точки данных, конфигурационные элементы и подстановочные параметры

Как было указано выше, ключевым элементом при построении информационной модели в WinCC OA является понятие точки данных, связывающей переменные (теги), относящиеся к некоторому устройству или объекту, в единую древовидную структуру с возможностью создания произвольных подуровней. На практике в точку данных, отно-

Реальное устройство:
Запорный клапан



Модель устройства:

| Сигналы состояния | | |
|----------------------------|---------------------|-----------------|
| ⇒ SS | Open | Открыт |
| ⇒ SS | Closed | Занрит |
| ⇒ SS | Position | Положение |
| Команды | | |
| ⇨ CMD | Open | Открыть |
| ⇨ CMD | Close | Закреть |
| ⇨ CMD | Stop | Останов |
| Алармы | | |
| ⇨ AL | Torque | Крутящий момент |
| ⇨ AL | Failure | Ошибка |
| Вспомогательная информация | | |
| • | Operating hours | Время наработки |
| • | Identification code | Идентификатор |
| • | Type | Тип |

Рис. 2. Информационная модель реального устройства на примере запорного клапана

сящуюся к одному устройству, объединяются примерно от 4 до 30 переменных. При обработке данных, полученных от интеллектуальных устройств, число переменных, включаемых в состав точки данных, может быть существенно больше.

Рассмотрим подробнее процесс создания информационной модели средствами WinCC OA на примере запорного клапана. В данном случае соответствие между реальным физическим устройством и его информационной моделью может быть проиллюстрировано схемой (рис. 2). В соответствии с идеологией WinCC OA для создания точек данных с такой структурой должен быть опре-

делен соответствующий тип точки данных (GATE_VALVE) (рис. 3), который будет выполнять роль шаблона при формировании множества точек данных с такой структурой. Создание типов точек данных и самих точек данных на их основе в системе WinCC OA осуществляется с помощью редактора PARA. В нашем примере соответствующие структуры данных будут выглядеть в редакторе PARA, как это показано на рис. 4 (точки данных V001...V005 типа GATE_VALVE; подробно показана структура точки данных V001).

Для гибкости и эффективности процесса инжиниринга важно, что при изменении типа точки

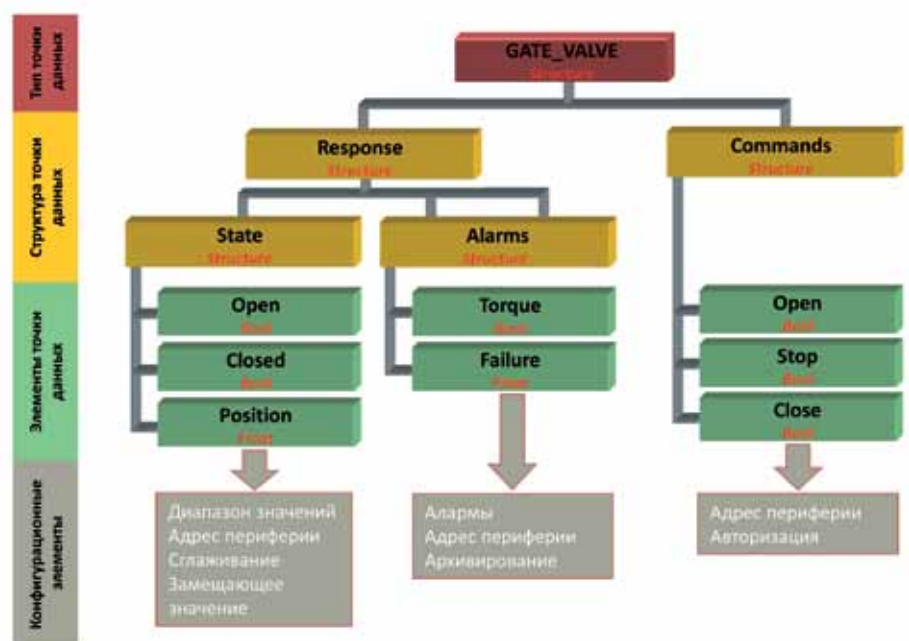


Рис. 3. Структура типа точки данных GATE_VALVE

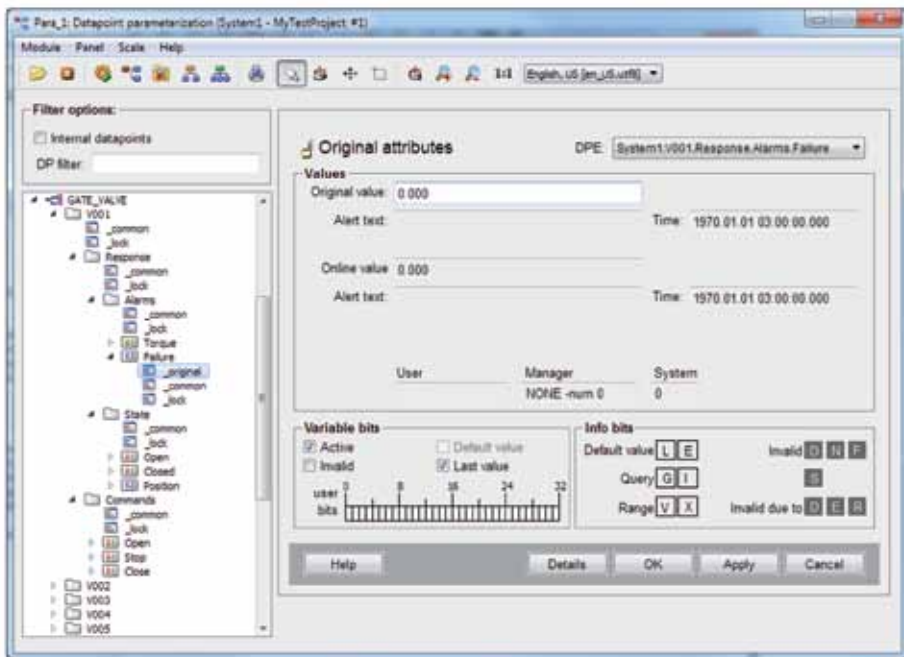


Рис. 4. Редактор PARA

данных также изменяются соответствующие точки данных (экземпляры). Например, при добавлении в тип точки данных GATE_VALVE структуры Maintenance с элементами OperatingHours и Services такие элементы появятся во всех точках данных, созданных на основе типа GATE_VALVE.

Помимо собственно структуры для хранения параметров и данных устройства, WinCC OA поддерживает назначение управляющих функций в отношении элементов точки данных – эта задача решается с помощью конфигурационных элементов. Например, конфигурационный элемент `_address` позволяет связать элемент точки данных с периферийным устройством, `_archive` – настроить параметры архивирования, `_smooth` – задать различные алгоритмы сглаживания, `_auth` – определить уровни полномочий пользователей, `_default` – задать заменяющее значение (рис. 3). Вместо того чтобы настраивать по отдельности конфигурационные элементы каждого элемента каждой точки данных, можно настроить конфигурационные элементы одновременно для целого класса устройств – для этой цели используются мастер-точки данных.

Адресация каждого элемента точки данных определяется по пути к нему в структуре точки данных. В нашем примере адрес перемен-

ной состояния Open («Открыто») устройства V002 может быть описан следующим образом:

V002.Response.State.Open .

В общем случае синтаксис обращения к точкам данных, их элементам и атрибутам выглядит следующим образом:

[System:]dpName.[dpElem1 [.dpElem2...]]:_Config.[Detail]_Attribute ,

где System – имя системы, dpName – имя точки данных,

dpElem1, dpElem2 – элементы точки данных, _Config, _Attribute – конфигурационный элемент и его атрибут. Выражения в квадратных скобках являются опциональными. Опцию Detail могут использовать только конфигурационные элементы блокировки, прав пользователя, преобразования и обработки сообщений.

Однако такая абсолютная адресация не всегда удобна. Типична ситуация, когда в проекте применяется множество точек данных одного типа (например, GATE_VALVE) и для каждой точки данных в общем случае необходимо создание соответствующих графических объектов (панелей). Использование абсолютной адресации привело бы к необходимости создания множества фактически одинаковых графических объектов, отличающихся исключительно идентификатором точки данных (в нашем примере – V001, V002, V002 и т. д.).

Система WinCC OA предлагает элегантное решение данной проблемы – создание символов устройств (включая все информационные панели, диалоги и т. д.) как многократно используемых объектов (reusable objects). При обращении к элементам точек данных в таких графических объектах используется \$-параметр (\$-parameter) – подстановочный параметр для имени точки данных. Когда объект, использующий \$-параметр, добавляется в какое-либо конкретное место на панели, необходимо просто указать, с каким физи-

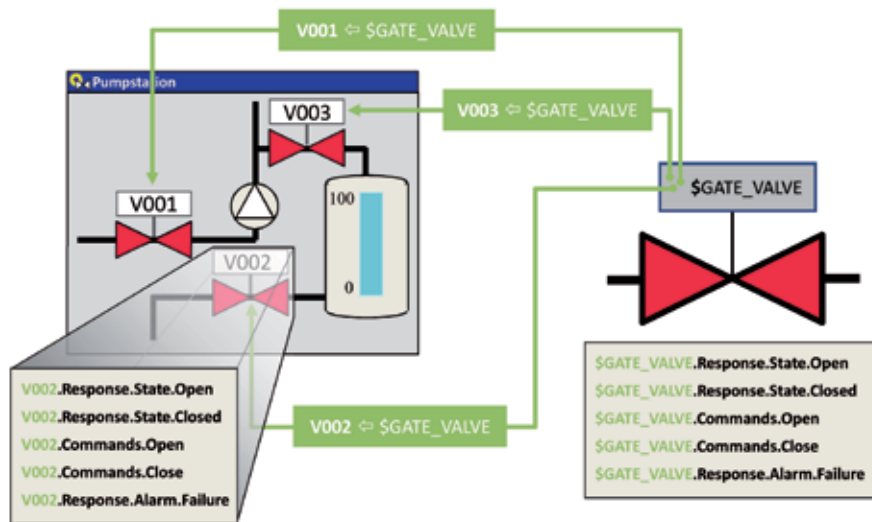


Рис. 5. Использование \$-параметров при обращении к объектам на графических панелях

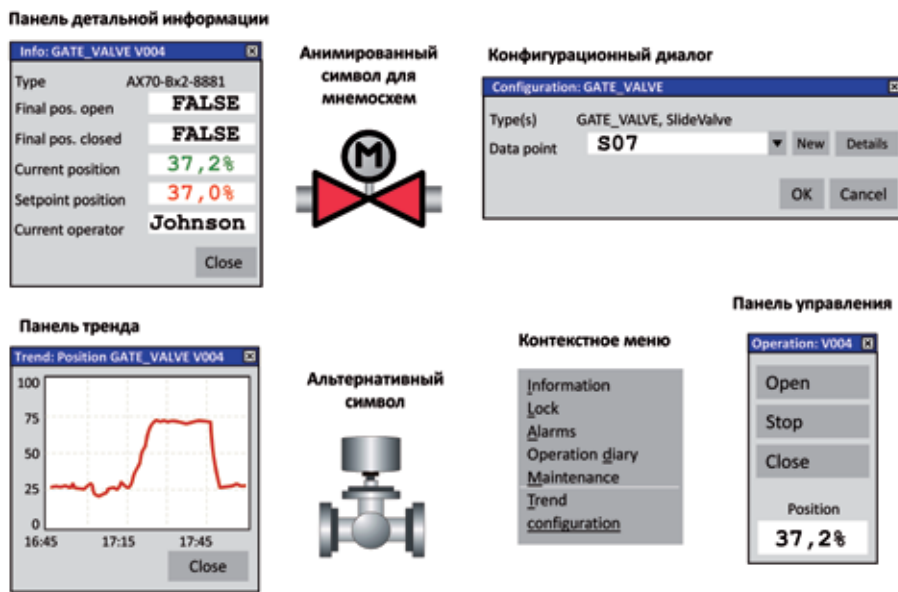


Рис. 6. Графические элементы объекта GATE_VALVE

ческим устройством (точкой данных) он связан (рис. 5).

Механизм \$-параметров является универсальным и может применяться для передачи параметров в графические символы, операторские панели или диалоговые окна, а диалоговые окна, в свою очередь, могут возвращать значения параметров вызывающей панели. Подстановочные \$-параметры могут использоваться для указания не только имен точек данных, но и цвета, размера, режима работы и других свойств объекта. При этом каждый графический объект может иметь несколько \$-параметров, а сами \$-параметры могут обрабатываться как строки.

В качестве примера практического воплощения этих принципов на рис. 6 показаны возможные графические элементы объекта GATE_VALVE, использующие \$-параметры.

Преимуществом механизма подстановочных параметров при работе с графическими объектами является прежде всего обеспечение эффективности процесса инжиниринга, гибкость внесения изменений и стандартизация визуальных интерфейсов:

- ▶ один символ устройства является шаблоном для создания экземпляров всех устройств этого типа в проекте;

- ▶ при добавлении полностью однотипных устройств в проект необходимо передать только один параметр — имя точки данных этого устройства (например, V002);

- ▶ если необходимо изменить внешний вид символов всех экземпляров однотипных устройств, можно изменить только параметры панели-источника, после чего автоматически изменятся все экземпляры устройств в различных образах процессов (наследование);
- ▶ аналогичный подход можно также применить и при создании диалоговых окон для символов устройств — в этом случае наследование также имеет место.

Таким образом, создание и настройка множества тегов, относящихся к определенному устройству или группе устройств, а также формирование соответствующих графических объектов сводятся к одной

операции — созданию точки данных. Ранее созданные типы точек данных, соответствующие какому-либо устройству или объекту, можно включать в состав новых типов точек данных. За счет такого иерархического подхода к созданию точек данных (тип в типе) обеспечиваются дополнительные возможности повышения эффективности в процессе разработки.

Среда разработки

Рассмотрим основной инженерный инструмент среды разработки системы WinCC OA, к которому относятся редактор внутренней базы данных PARA, инструменты массового параметрирования и графический редактор GEDI.

Инженерный инструмент вместе с базовыми средствами системы WinCC OA (концепция менеджеров, поддержка развитых коммуникаций, распределенных систем и пр.) составляют единую среду разработки, полностью отвечающую идеологии WinCC OA, — эффективный инжиниринг открытых систем (рис. 7).

Редактор внутренней базы данных PARA

Редактор PARA представляет собой инструмент для доступа к внутренней базе данных системы WinCC OA, позволяющий создавать, удалять и редактировать точки данных и типы точек данных. Как типы точек данных, так и точки данных визуально отображаются в редакторе

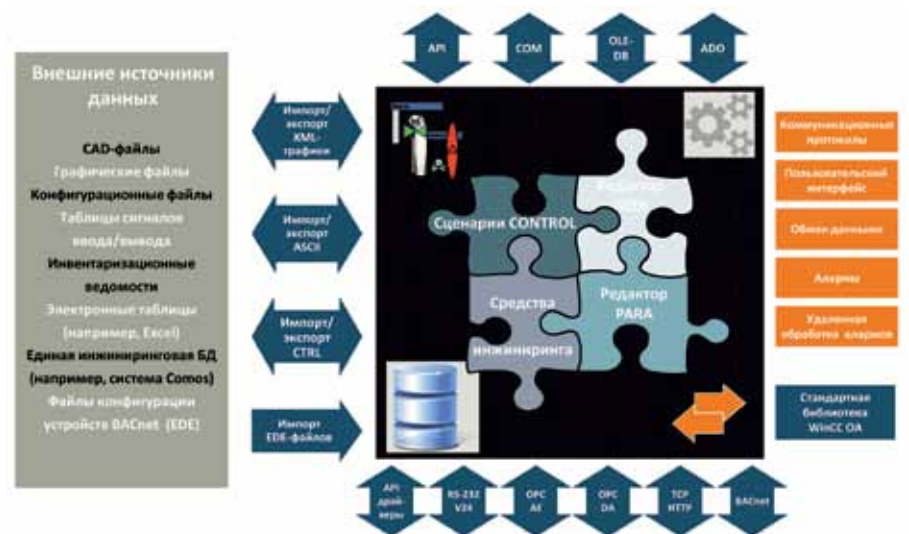


Рис. 7. Среда разработки WinCC OA обеспечивает эффективный инжиниринг открытых систем

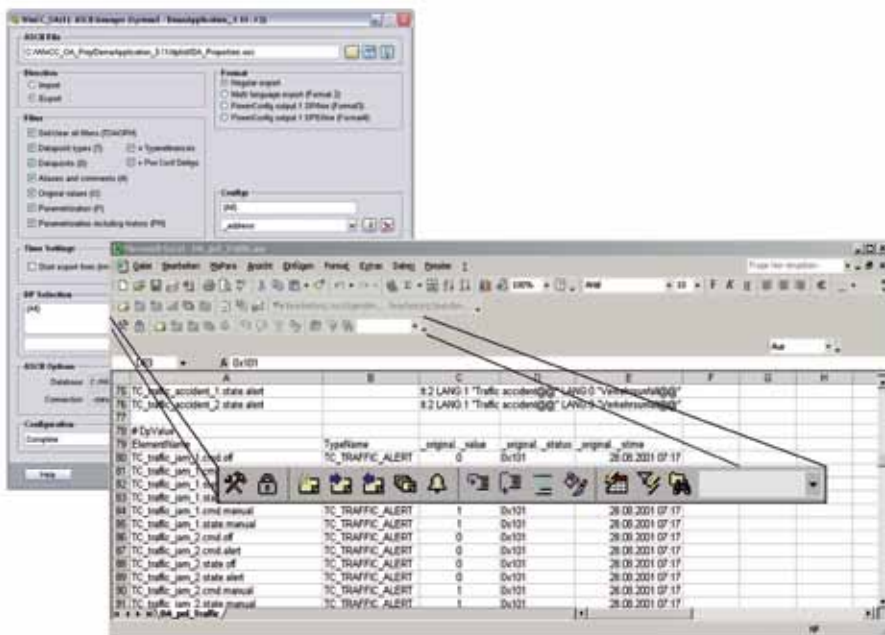


Рис. 8. Менеджер ASCII и инструменты Excel для работы с шаблонами

ные настройки будут переданы во все имеющиеся точки данных.

Менеджер ASCII (рис. 8) позволяет выполнять импорт и экспорт точек данных, типов точек данных, а также их конфигураций через ASCII-файл. Благодаря наличию специальных форматов ввода/вывода в ASCII-менеджере, с помощью программ для работы с электронными таблицами (например, Microsoft Excel) осуществляется простое массовое параметрирование. За счет создания шаблонов можно указывать, какие настройки будут впоследствии нужны для каждого экземпляра, а какие будут приниматься вне зависимости от мастер-точки данных (типа точки данных). Инструменты Excel для массового параметрирования позволяют пользователю импортировать готовые шаблоны в книгу Excel и выполнять в ней изменения точек данных и атрибутов. После сохранения книги Excel шаблон может быть обновлен или изменен, выполненные в Excel, могут быть импортированы в базу данных. Компоновка рабочего листа в Excel обеспечивает удобный просмотр импортированного шаблона ASCII. Благодаря этому можно быстро редактировать набор параметров атрибутов или создавать новые точки данных.

PARA в виде древовидной структуры (рис. 4). Узлы и конечные элементы данной структуры можно добавлять и удалять щелчком мыши. Для настройки атрибутов конфигурационных элементов предусмотрены стандартные панели. Для удобства инжиниринга все возможные альтернативы при создании задаются переключателями и флажками. Также редактор PARA поддерживает переименование точек данных (DP Rename), поиск мест использования точек данных в проекте (Cross References), проверку использования вводимых имен.

шаблонные конфигурационные элементы (Power Configs), которые автоматически передаются в каждую новую точку данных определенного типа при ее создании. Изменения мастер-точки данных также передаются в точки данных во время выполнения программы. Шаблонные конфигурационные элементы представляют собой пользовательские виртуальные конфигурационные элементы, являющиеся комбинацией или конкретизацией настроек (например, обработка алармов с тремя диапазонами). Если такой шаблонный конфигурационный элемент добавляется в элемент мастер-точки данных, создан-

Графический редактор GEDI

Редактор GEDI используется для создания и редактирования панелей как экранов оператора в целом, так и простых графических

Инструменты массового параметрирования

В системе WinCC OA базовый арсенал средств редактора PARA логичным образом дополняют инструменты массового параметрирования, позволяя осуществлять эффективный инжиниринг крупномасштабных проектов. Эти инструменты представлены мастер-точками данных и шаблонными конфигурационными элементами, а также менеджером ASCII, шаблонами и списком точек данных, инструментарием Excel. Кроме того, имеется возможность создания и удаления множества точек данных.

Ключевым элементом, обеспечивающим возможность массового параметрирования в системе WinCC OA, является *мастер-точка данных (Master Data Point)*, содержащая

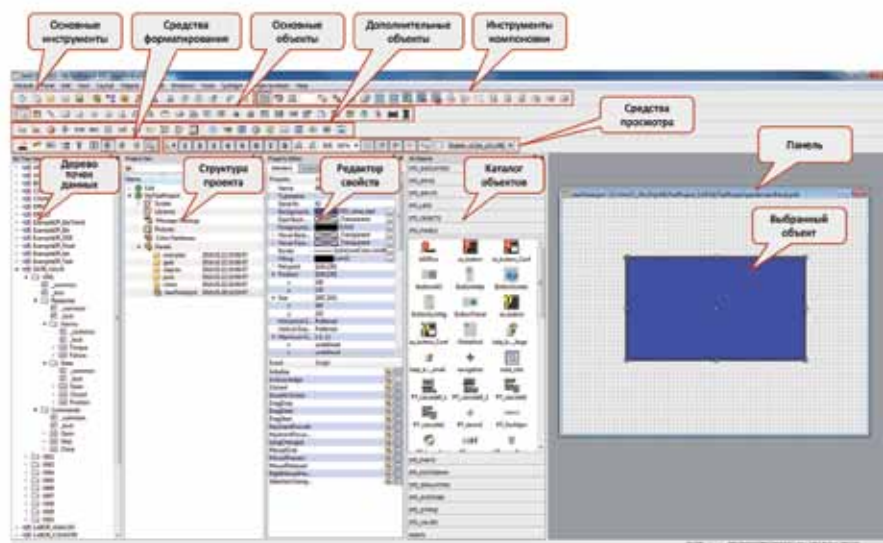


Рис. 9. Графический редактор GEDI

объектов (панелями в WinCC OA называются все графические элементы интерфейса пользователя). Редактор GEDI (рис. 9) состоит из строки меню, панелей инструментов, окна структуры проекта, каталога объектов, редактора свойств, дерева точек данных, одного или нескольких окон панелей и редактора сценариев, объединенных в интегрированную среду.

Окно структуры проекта используется для быстрой навигации по элементам структуры проекта (панелям, сценариям, библиотекам, изображениям и т. п.), отображаемой в виде наглядного иерархического дерева.

Каталог объектов содержит предопределенные графические символы и элементы разной сложности, которые пользователь может перетащить на панель (drag & drop) и применить в своем прикладном проекте. Каталог объектов включает 12 подкаталогов, в которых содержатся:

- объекты графического отображения значений (круговые и столбчатые диаграммы, индикаторы выполнения, стрелочные индикаторы, тренды и т. д.);

- объекты для пользовательского ввода значений, диапазонов значений, времени;

- кнопки;

- переключатели разного вида;

- стандартные объекты для визуализации ТП (клапаны, моторы, вентиляторы, компрессоры, счетчики и т. п.);

- имитаторы сигналов различной формы и многие другие элементы.

Возможности стандартных объектов каталога WinCC OA существенно расширяет технология внешних виджетов (External Widget Objects, EWO). Последние создаются разработчиком графических объектов и могут быть встроены в любую панель WinCC OA. В языке C++ эта функция реализуется с помощью платформонезависимого графического тулкита Qt (<http://qt-project.org/>). EWO может содержать и отображать любые данные, которые инструмент Qt позволяет поместить в объект QWidget. EWO динамически загружается в пользовательский интерфейс WinCC OA (компилируется в качестве плагина и размещается в совместно использу-

емой библиотеке.dll-файлов) и полностью встраивается в сценарии языка CTRL, при этом QWidget передает сценарию все его свойства. EWO может определять, к какому методу обращается сценарий языка CTRL, а также запускать сценарии CTRL с помощью соответствующих сигналов.

Редактор свойств используется для отображения и редактирования свойств панелей и графических объектов, а также содержит инструмент для настройки сценариев обработки событий, связанных с элементами графического интерфейса (с помощью *мастера сценариев* или редактора сценариев). При этом в случае использования мастера сценариев логика обработки событий может быть определена без написания единой строчки кода, что существенно облегчает процесс инжиниринга. В обратной ситуации, когда необходимо определить специфические пользовательские функции обработки событий, разработчик получает в свое распоряжение весь арсенал средств языка сценариев CTRL благодаря мощному *редактору сценариев*. Последний поддерживает ряд средств, повышающих эффективность работы: выделение синтаксиса с возможностью индивидуальной цветовой схемы, подстановку имен библиотечных функций, всплывающие подсказки по параметрам функций и др.

Процесс создания *пользовательских панелей* выполняется в рабочей области панели, обеспечивающей отрисовку и редактирование графических объектов. При выделении объекта (например такого, как выделенный прямоугольник на рис. 8) отображаются его параметры, которые доступны для редактирования через редактор свойств. Перенос объектов из каталога на пользовательскую панель осуществляется методом drag & drop. Редактор GEDI имеет широкие возможности для построения развитой графики в соответствии с современными требованиями (трехмерные объекты, анимация, эффекты переходов и градиенты, расширенные настройки табличных представлений и т. д.); поддерживаются все общепринятые функции визуального форматирования. В пользовательских панелях можно применять до восьми слоев

видимости. Число одновременно редактируемых панелей неограниченно.

Важно отметить, что редактор GEDI представляет собой единый инструмент создания панелей для всех поддерживаемых WinCC OA операционных систем.

Эффективность процесса создания крупномасштабных проектов командами разработчиков, помимо возможностей инструментария самой системы WinCC OA, обеспечивается также *поддержкой систем управления версиями*. В WinCC OA можно управлять проектами с помощью различных систем управления версиями — такими как Apache Subversion, также известной как SVN (<http://subversion.apache.org/>), и Concurrent Versions System, CVS (<http://www.nongnu.org/cvs/>). Функциональностью этих систем можно пользоваться и непосредственно в редакторе GEDI. Так, файлы проекта можно обновить или добавить в хранилище данных не покидая проект.

Кроме того, инжиниринговый инструментарий системы WinCC OA обеспечивает непревзойденную эффективность при создании типовых решений с последующим тиражированием. Как показывает практика, средства WinCC OA позволяют до 40% снизить трудоемкость инжиниринга при разработке второго и последующих проектов, если сравнивать с разработкой аналогичных проектов обычными средствами традиционных SCADA-систем.

Язык сценариев CONTROL (CTRL)

Программы (сценарии) на языке CTRL используются для управления системой WinCC OA. Программа осуществляет обработку точек данных, а также управление визуализацией состояний процесса и таким образом обеспечивает поддержку гибкой концепции точек данных WinCC OA в части программирования пользователем реакции системы на то или иное событие. Синтаксис языка CTRL основан на синтаксисе языка программирования C. Выполнение сценариев на языке CTRL осуществляется интерпретатором, поэтому пользователю не нужно проводить компиляцию или сборку, что дает возможность непосредственно тестировать и применять изменения. Запуск сценари-

ев CTRL может осуществляться при наступлении определенного события (например, сообщения об изменении точки данных), действий пользователя или циклически.

Структура сценариев CTRL. Каждый сценарий языка CTRL содержит функцию main(). Эта функция обрабатывается первой при запуске сценария. При этом способ вызова сценариев, относящихся к графическим элементам, зависит от типа атрибута, а именно от того, является ли он «пассивным» или «активным». Пассивными атрибутами графического элемента являются, например, цвет границы, ее видимость, шрифт текста и т. д. Они используются для визуализации состояний процесса и таким образом необязательно связаны с вводом данных пользователем. Сценарии пассивных атрибутов запускаются при открытии панели, поэтому их функция main() обрабатывается один раз на такое событие. К активным относятся атрибуты, требующие ввода данных пользователем. Примером является атрибут Clicked, который срабатывает в случае щелчка мыши на объекте, а также атрибуты для полей Command в редакторе свойств, которые отсылают текстовые входные данные или сообщения от переключателей и кнопок-флажков. Сценарии для активных атрибутов выполняются каждый раз при активации объекта пользователем, например с помощью щелчка на нем мышкой или ввода текста, и при этом при каждом запуске вызывается функция main().

Типы данных и управляющие конструкции. Язык CTRL поддерживает все основные типы данных: битовые шаблоны (bit32, bit64), целые (char, int, unsigned, uint, long, ulong), с плавающей запятой (float/double), строки (string), время (time) и другое, а также управляющие конструкции, характерные для развитых языков процедурного программирования (if-else, switch, while, for, do-while, break, continue, try-catch/finally-throw).

Функции. Язык CTRL поддерживает более 200 функций, среди которых: математические, функции точки данных, обработки алармов, передачи данных, времени, управление менеджерами, модулями и панелями, ActiveX, функция обработки ошибок и др.

Особое место в массиве функций языка CTRL занимают функции работы с атрибутами элементов точек данных и графических объектов, востребованные при создании пользовательских сценариев:

- dpGet() и dpSet() – функции чтения и записи значений атрибутов точек данных;
- dpConnect() – функция обратного вызова (callback function) при изменении передаваемых значений точки данных/атрибутов;
- getValue() и setValue() – функции чтения и записи значений атрибутов для графических объектов.

Библиотеки. Сценарии на языке CTRL могут быть оформлены в виде отдельных библиотек, доступных для использования во всех сценариях проекта. Создание пользовательских библиотек позволяет упорядочить структуру сценариев в крупномасштабных проектах и обеспечивает возможность повторного использования ранее написанного кода. Библиотека на языке CTRL может изменяться пользователем средствами редактора GEDI или в любом текстовом редакторе. Библиотека может быть изменена во время работы.

Прочие инженеринговые инструменты – палитра возможностей

Отладчик сценариев CTRL Debugger (рис. 10). В системе WinCC OA является средством отладки сценариев, написанных на языке CTRL. Основные функции отладчика:

▸ просмотр сценариев, которые выполняются средствами пользовательского интерфейса или средствами менеджера сценариев CTRL;

▸ определение точек останова, которые сообщают работающему менеджеру, что при достижении назначенной строки необходимо остановить исполнение сценария;

▸ выполнение инструкций в пошаговом режиме и установка значений переменных;

▸ просмотр и установка значений переменных во время выполнения.

Важной особенностью отладчика CTRL Debugger является возможность отладки сценариев независимо от того, где в системе WinCC OA запущен менеджер, обрабатывающий сценарии, – даже в распределенной системе.

Гибкая модель объекта. С помощью инструмента Flexible Plant Model можно создать гибкую модель объекта – иерархические структуры данных и окна просмотра на основе Common Name Service, CNS (рис. 11). Этот инструмент позволяет сопоставлять точки данных и элементы точек данных с различными представлениями, в которых одни и те же данные могут отображаться различными способами, обеспечивая тем самым выполнение специфических требований пользователей. При этом самих иерархических структур также может быть задано множество. Создание и настройка таких структур и окон просмотра осуществляются с помощью собственного редакто-

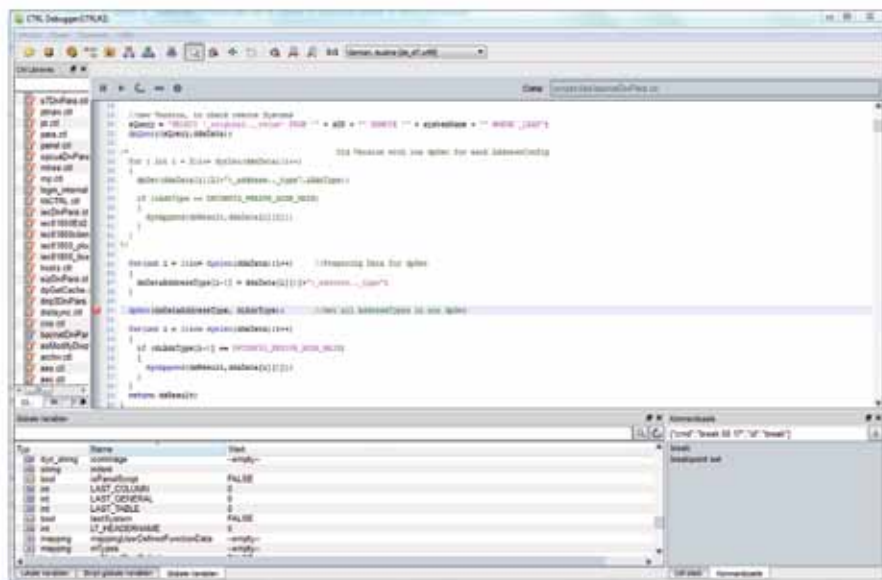


Рис. 10. Отладчик сценариев CTRL Debugger

ра инструмента Flexible Plant Model, благодаря чему обеспечивается простота и эффективность процесса инжиниринга.

Инструмент *Wizard Framework* используется для создания мастеров в едином стиле. Элементы мастера могут быть настроены в соответствии с потребностями пользователя.

Интерфейс прикладного программирования (Application Programming Interface, API) WinCC OA обеспечивает возможность дополнения системы менеджеров WinCC OA менеджерами, создаваемыми пользователем для решения специальных задач. Также WinCC OA API позволяет создавать собственные драйверы, расширения языка CTRL, расширения драйверов и графические объекты EWO. Менеджеры в терминологии WinCC OA – это модули системы, каждый из которых выполняет определенную функцию (обработка сообщений, поддержка протоколов обмена данными, интерфейс пользователя, архивирование, выполнение сценариев CTRL и др.).

Менеджеры обмениваются информацией друг с другом с помощью сообщений (WinCC OA имеет фиксированный набор сообщений, используемых для связи). Большинство функций интерфейса WinCC OA API используется для создания и отправки сообщений. Менеджеры получают сообщения с помощью функции `doReceive()`, которая применяется ко всем входящим сообщениям.

Разработка собственных менеджеров, драйверов или расширений

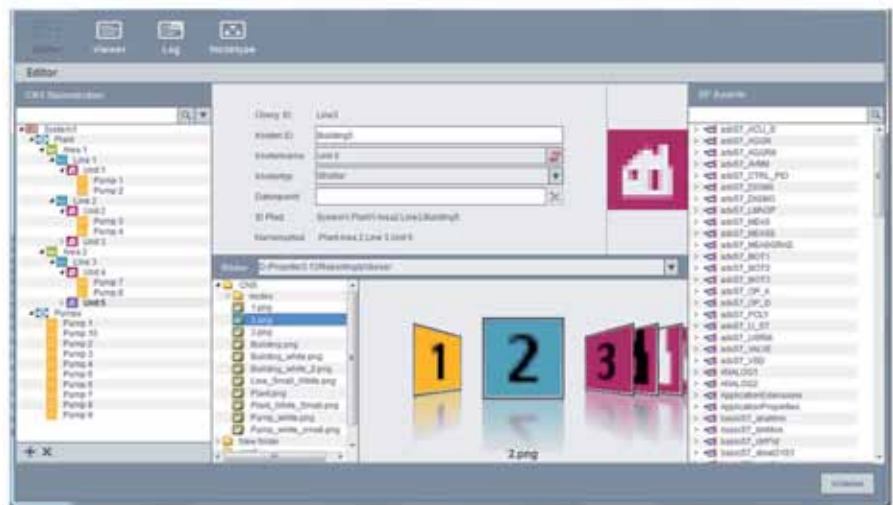


Рис. 11. Пример гибкой модели объекта на основе CNS

на базе WinCC OA API осуществляется в среде Visual C++ / Visual Studio 2013 (версии 18.00.30501 для x64) для ОС Windows и Red Hat Enterprise Linux 7.1 (сборка с gcc-4.8.2-16) для ОС Linux.

Использование API позволяет удовлетворить специфические требования заказчика и (или) обеспечить соответствие системы региональным или отраслевым требованиям путем полноценной интеграции с системой модулей собственной разработки. Этот механизм также дает возможность использовать созданные ранее программные наработки, в том числе реализованные сторонними средствами.

Средства диагностики. В дополнение к основному инжиниринговому инструментарию редактора GEDI и возможностям отладчика

сценариев на языке CTRL система WinCC OA располагает средствами общей диагностики проекта, среди которых обзор структуры менеджеров, просмотр статуса подключенный менеджера событий, алармов, статистики сессий работы и т. п. Пример средств диагностики приведен на рис. 12.

Заключение

Система WinCC OA предлагает современный подход к созданию SCADA-проектов на базе объектно ориентированной методологии и широкого использования открытых стандартов и технологий. Развитый инструментарий системы WinCC OA обеспечивает эффективность процесса инжиниринга и сокращает сроки разработки и ввода в промышленную эксплуатацию крупномасштабных систем. Модульная архитектура системы WinCC OA и гибкая концепция инжиниринга предоставляют все необходимые возможности для создания надежных прикладных решений, отвечающих самым жестким требованиям заказчиков.

Литература

1. С. Ю. Соловьев, А. С. Космин. Построение распределенных систем сбора и обработки данных на базе платформы SIMATIC WinCC Open Architecture / ИСУП. 2015. № 4.

С.Ю. Соловьёв, к. т. н.,
руководитель группы,
ООО «Сименс», г. Москва,
тел.: +7 (495) 737-2441,
e-mail: dfpd.ru@siemens.com,
www.dfpd.siemens.ru

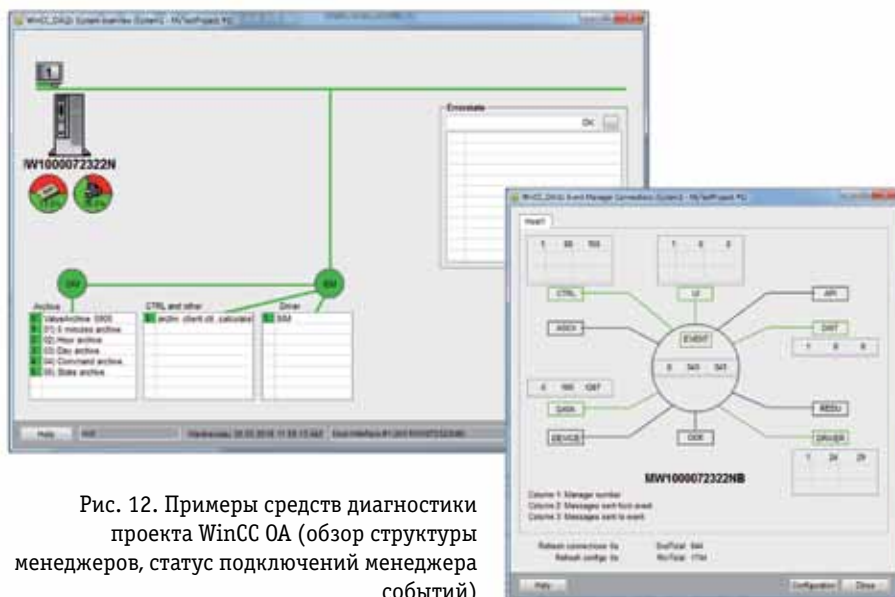


Рис. 12. Примеры средств диагностики проекта WinCC OA (обзор структуры менеджеров, статус подключений менеджера событий)