



SCADA-системы: проблема выбора

Владимир Бунин, Валентин Анопенко, Алексей Ильин, Ольга Салова,
Наталия Чибисова, Алексей Якушев

Введение

Фирма, в которой мы работаем, — АО «Система-Сервис» — имеет достаточно богатый опыт в области создания систем АСУ ТП для объектов магистральных газопроводов, в первую очередь, для газокompрессорных станций. В прошлом применяемое нами программное обеспечение как нижнего, так и верхнего уровня (АРМ сменного инженера) являлось продуктом собственной разработки. Но время показало, что дублирование своими силами результатов деятельности специализированных фирм-разработчиков SCADA-систем — дорогое удовольствие, отвлекающее к тому же ограниченные ресурсы наших программистов от решения насущных проектных задач. Поэтому был поставлен вопрос о выборе стандартной SCADA-системы для решения, по крайней мере, задач верхнего уровня АСУ ТП, включая ПО операторского интерфейса.

Работы проводилась в три этапа. На первом этапе по заявленным для различных пакетов технико-коммерческим характеристикам проводился первичный отбор пакетов, подлежащих тестированию. На втором этапе дистрибьюторы и разработчики SCADA-систем были приглашены для проведения презентации своих продуктов. Для уточнения характеристик двух систем, оказавшихся по итогам первых этапов наиболее перспективными (iFIX и GENESIS32), была разработана и реализована программа и методика тестирования.

Хотя работа проводилась для собственных нужд, мы сочли полученные результаты интересными и для других специалистов в области промышленной автоматизации.

Мы не ставили своей задачей выбрать абсолютного победителя, тем более что в такой быстро развивающейся области, как SCADA-системы, такое звание может быть весьма кратковременным и эфемерным. К тому же в различных проектах АСУ ТП могут оказаться наиболее важными совершенно различные характеристики SCADA-пакета. Думаем, читатель, ознакомившись с результатами испытаний и нашими, возможно, субъективными комментариями, сам сможет сделать необходимые выводы. В основном нас интересовали те характеристики пакетов, которые влияют на создание эффективного человеко-машинного интерфейса и на решение других задач, характерных для верхнего уровня АСУ ТП. Так как на нижнем уровне мы по-прежнему собираемся использовать программное обеспечение собственной разработки, то инструментальные средства для программирования на языках МЭК-61131-3 не рассматривались.

Требования к SCADA

Нас, в первую очередь, интересовали следующие характеристики SCADA-систем:

- качество документации;
- техническая поддержка в России;
- открытость и масштабируемость;
- полнофункциональность;
- надежность;
- эффективность;
- цена.

Мы считаем качество сопроводительной документации одной из ключевых характеристик, так как наша фирма работает не по типовой двухступенчатой схеме «системный интегратор — конечный пользователь», а по трехступенчатой.

То есть мы делаем множество проектов в различных регионах России, сопровождение и изменение которых может выполняться специалистами заказчика и региональных сервисных фирм. Низкое качество документации (например, ее неполнота или англоязычность) приведет к фактическому ухудшению качества сопровождения.

Техническая поддержка определяет, сколько времени и сил придется затратить системному интегратору на освоение всех возможностей системы. При ее отсутствии зачастую оказывается, что проще и быстрее написать программу самим, нежели разобрататься, как пользоваться готовой.

Вопрос масштабируемости значим потому, что мы, как и любой другой системный интегратор, имеем проекты разного масштаба (от сотен сигналов до десятков тысяч). Иметь SCADA-пакет, применяемый либо только в малых, либо только в больших системах, непрактично.

Требование открытости имеет несколько основных аспектов.

- Во-первых, это возможность сопряжения данной SCADA с различными продуктами других фирм-производителей (ПО технологических контроллеров, СУБД, другие SCADA).
- Во-вторых, это наличие мощного и универсального скриптового языка.
- В-третьих, это возможности встраивания в SCADA готовых компонентов (в первую очередь — ActiveX).

Под полнофункциональностью понимается способность пакета решать весь комплекс задач промышленной автоматизации, выдвигаемых перед программным обеспечением на верхнем

уровне АСУ ТП, а не некоторое их подмножество.

Надежность — это не только отсутствие ошибок в программном коде самого пакета, но и его устойчивость к ошибкам во внешних компонентах и к некорректным действиям обслуживающего персонала.

Эффективность SCADA-системы в первом приближении сводится к тому, насколько мощный компьютер требуется для реализации с ее помощью конкретных проектов. Так как в составе реальных проектов одновременно используется множество функций SCADA-системы, желательнее чтобы каждая из ее подсистем (графическая, ввода-вывода, архивирования и т.п.) обладала необходимой функциональностью и быстродействием.

Цена — это и цена лицензий на исполнительные модули пакета (run-time-модули), и зависимость этой цены от числа тегов и запрошенного набора функций, а также стоимость документации и обучения.

Какие SCADA-пакеты мы рассматривали?

Общее количество доступных SCADA-продуктов весьма велико, поэтому мы должны были как-то ограничить круг рассмотрения. Нами не рассматривались системы, не имеющие официальных дистрибьюторов в России. Мы брали в расчет только Windows-программы. Продукты, работающие под управлением ОС UNIX и ее клонов (Linux, QNX), не тестировались. Мы учитывали только те программы, которые используются в сетевых версиях и допускают работу с десятками тысяч тегов. Отечественный пакет Trace Mode подробно не рассматривался, в связи с негативными откликами его пользователей.

После обзора литературных источников и консультаций со специалистами для предварительного ознакомления были избраны следующие пакеты:

- iFIX фирмы Intellution;
- InTouch фирмы Wonderware;
- GENESIS32 фирмы Iconics;
- WinCC фирмы Siemens.

Популярный пакет InTouch был нами отвергнут из-за слабой технической поддержки дистрибьюторами в России, использования специального скриптового языка и некоторых недостатков графической подсистемы. WinCC предъявляет максимальные требования к аппаратным ресурсам ПЭВМ и ощутимо оптимизирован именно для контроллеров SIMATIC фирмы Siemens.

Наилучшее впечатление по совокупности всех факторов на нас произвели

iFIX и GENESIS32. Именно эти продукты и были избраны для более тщательного тестирования.

Тестировался пакет GENESIS32 Version 5.2 Build 54 и iFIX Version 2.1 Build 1121 совместно с OPC Power Tool 7.12.

Тестирование проводилось на компьютере следующей конфигурации: CPU Celeron-300; RAM 64 Мбайт; Video S3 Virge DX (размер видеопамати — 4 Мбайт), разрешение экрана 800×600 при 65535 цветах; сетевой адаптер Ethernet 10 Мбит/с.

Использовалась операционная система Windows NT Workstation 4.0 (Service Pack 3). Отдельно проверялась работоспособность GENESIS32 в среде Windows 95 (iFIX работает только на платформе Windows NT).

Первые впечатления

Установка

Как театр начинается с вешалки, так и использование ПО начинается с его установки на компьютер. Процесс установки и GENESIS32, и iFIX не вызвал у нас каких-либо проблем, а последовательность необходимых действий в обоих случаях была интуитивно понятной.

Ресурсоемкость

Требования к аппаратным ресурсам целевого компьютера, согласно информации фирм-производителей, следующие:

iFIX

- Процессор Pentium II 300 МГц, 96 Мбайт ОЗУ, 120 Мбайт на НЖМД.

GENESIS32

- Процессор Pentium 233 МГц, 64 Мбайт ОЗУ, 200 Мбайт НЖМД (для приложений средней и большой величины).
- Процессор Pentium 166 МГц, 32 Мбайт ОЗУ, 200 Мбайт НЖМД (для небольших приложений).

Если GENESIS32 может функционировать под управлением операционных систем Windows 95/98 и Windows NT, то iFIX работает только с Windows NT. Хотя Windows NT потребляет больше ресурсов и стоит дороже, чем Windows 95/98, для ответственных приложений мы рекомендуем, опираясь на наш опыт работы, всегда использовать более надежную Windows NT.

В то же время для операторских станций, стабильность функционирования которых не является критической, с точки зрения технологического процесса, GENESIS32 предоставляет возможность воспользоваться более дешевой и неприятной Windows 95/98.

Компонентность

Анализ компонентности проводился с целью определить возможность закупки и использования не SCADA-системы в целом, а отдельных ее компонентов.

iFIX имеет в своем составе один основной компонент, называемый Workspace, который в режиме исполнения реализует практически все функции SCADA-системы.

GENESIS32 включает в себя три основных компонента (GraphWorX32, TrendWorX32 и AlarmWorX32), каждый из которых способен работать автономно и выполняет свою часть функций, в том числе в части интерфейса с оператором:

- GraphWorX32 предназначен для разработки и исполнения графических мнемосхем;
- TrendWorX32 предназначен для построения графических зависимостей контролируемых параметров и архивации;
- AlarmWorX32 предназначен для обнаружения, фильтрации и представления информации об аварийных событиях.

Таким образом, GENESIS32, в отличие от более «монолитного» iFIX, обеспечивает возможность экономии средств при лицензировании своих программных компонентов, количество и состав которых определяются прикладной задачей.

Кроме того, установка только необходимых компонентов позволяет уменьшить требования к аппаратным ресурсам компьютера.

Стоимость

Для малобюджетных проектов вопрос стоимости зачастую является определяющим. Компонентная архитектура GENESIS32 позволяет приобретать только нужные компоненты и на необходимое число точек ввода-вывода.

Ценовую информацию по системе GENESIS32 можно получить по адресу www.prosoft.ru. Информация по стоимости iFIX в России не является общедоступной, а высылается по запросу. Тем не менее, в соответствии с нашими расчетами для нескольких типовых конфигураций SCADA-системы, произведенными на основе данных, полученных от дистрибьютора Intellution в России, iFIX во всех случаях оказался существенно дороже, чем GENESIS32.

Графическая подсистема

Одной из обязательных функций, реализуемых любой SCADA-системой, является представление оператору на экране наглядной информации о ходе

технологического процесса. В связи с этим вопросы разработки и реализации операторского интерфейса представляют непосредственный интерес для системных интеграторов.

Удобство разработки

Как GENESIS32, так и iFIX имеют в своем составе мощные и гибкие средства, обеспечивающие пользователю эффективное создание экранных форм с минимальными затратами труда и времени. Оба приложения содержат наборы графических примитивов для рисования и наборы типовых графических объектов. Набор графических объектов для повторного использования (в терминологии iFIX — альбом динамо, в терминологии GENESIS32 — библиотека символов) может быть расширен пользователем. При этом объект, созданный пользователем, может быть сохранен в библиотеке вместе с установленными динамическими соединениями и VBA-скриптами.

Тем не менее, каждый пакет имеет некоторые особенности, которые следует отметить.

- В GENESIS32 по сравнению с iFIX более удобно для разработчика реализована операция Duplicate (дублировать). Это выражается в том, что после выполнения данной операции один раз и перетаскивания объекта в нужное положение (на определенном расстоянии от созданного ранее объекта) последующее выполнение операции дублирования избавляет разработчика от необходимости каждый раз перемещать объект в требуемое место (то есть операция дублирования не просто создает копию объекта, но и размещает новый объект на том же расстоянии от предыдущего, что и исходный объект). В iFIX все копируемые объекты появляются с одинаковым предопределенным смещением от оригинала, и требуется вручную перетаскивать их в нужное положение.

- В GENESIS32 имеются шаблоны экранных форм и объектов, отсутствующие в iFIX. Шаблоны экранных форм аналогичны шаблонам документов Microsoft Office. Для создания нескольких экранов, содержащих однотипные элементы, можно просто создать шаблон, в который следует включить повторяющиеся объекты, и применять его при создании каждого экрана. Шаблон объекта резервирует место для вставки объекта определенного типа (bmp, wmf или OLE). Это может быть полезно, если сам объект импортируется в экранную форму позднее. Пример использования шаблона показан на рис. 1.

- При создании нового элементарного графического объекта в GENESIS32 изначально у него присутствуют только статические свойства (например цвет, тип контурной линии). Если какое-либо свойство объекта (изменение размера, цвета, положения и т.п.) анимируется, то оно появляется как вкладка окна свойств объекта. Окно свойств при этом содержит только необходимую информацию, отражающую состояние данного объекта. Напротив, в iFIX создаваемый объект обладает сразу всеми свойствами даже при отсутствии их анимации, как показано на рис. 2. Это перегружает окно свойств объекта излишними деталями.

- В GENESIS32 графические объекты, предназначенные для повторного использования, сохраняются в библиотеке символов, которая является отдельным приложением и имеет интерфейс, аналогичный проводнику Windows, как показано на рис. 1. Можно настраивать корневой каталог библиотеки и таким образом иметь несколько отличающихся библиотек, что удобно при разработке проектов в различных предметных областях. И наоборот, при редактировании мнемосхемы iFIX предоставляет интерфейс системного дерева, аналогичный проводнику Windows (на рис. 2), а GENESIS32 — нет. Системное дерево iFIX отображает в иерархическом виде файлы и папки, содержащие все элементы системы (средства создания и модификации базы данных, документы, альбомы динамо, мнемосхемы, тревоги и др.).

- В iFIX доступ к объектам, входящим в состав группы, может быть получен без

разгруппировки через системное дерево, в то время как в GENESIS32 эта операция выполняется несколько менее прозрачным, но более быстрым способом (щелчком правой клавишей мыши на символе при одновременном удержании клавиши <Shift>).

- В отличие от GENESIS32, в iFIX кнопки разных панелей инструментов могут быть объединены в Toolbox (инструментарий). Это позволяет не выводить на экран несколько различных панелей инструментов, а ограничиться одной панелью Toolbox с часто используемыми кнопками. За счет этого, во-первых, экономится место на экране, а во-вторых, быстрее осуществляется доступ к отдельным операциям (рис. 2).

- В iFIX более просто и удобно для разработчика реализованы операции создания таких графических примитивов, как закругленный прямоугольник, многоугольник, хорда и сектор. Для этого существуют отдельные кнопки на панели инструментов. В GENESIS32 создание подобных объектов также возможно, но осуществляется это несколько более трудоемким способом. Например, закругленный прямоугольник можно создать из простого прямоугольника, настроив соответствующим образом его свойства. Правда, следует отметить, что при таком способе реализации пользователь имеет возможность настроить параметры объекта (например радиус скругления), что в iFIX невозможно.

- В iFIX, в отличие от GENESIS32, действия разработчиков, создающих экранные формы, протоколируются, что может оказаться полезным, например, в

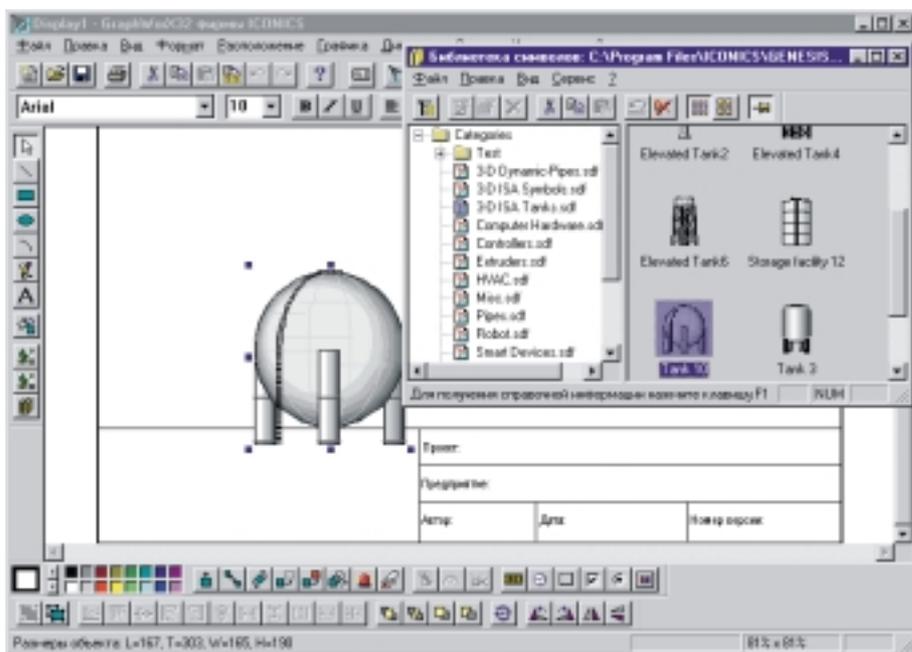


Рис. 1. Библиотека символов и шаблон стандартной рамки, примененный к экранной форме GraphWorX32

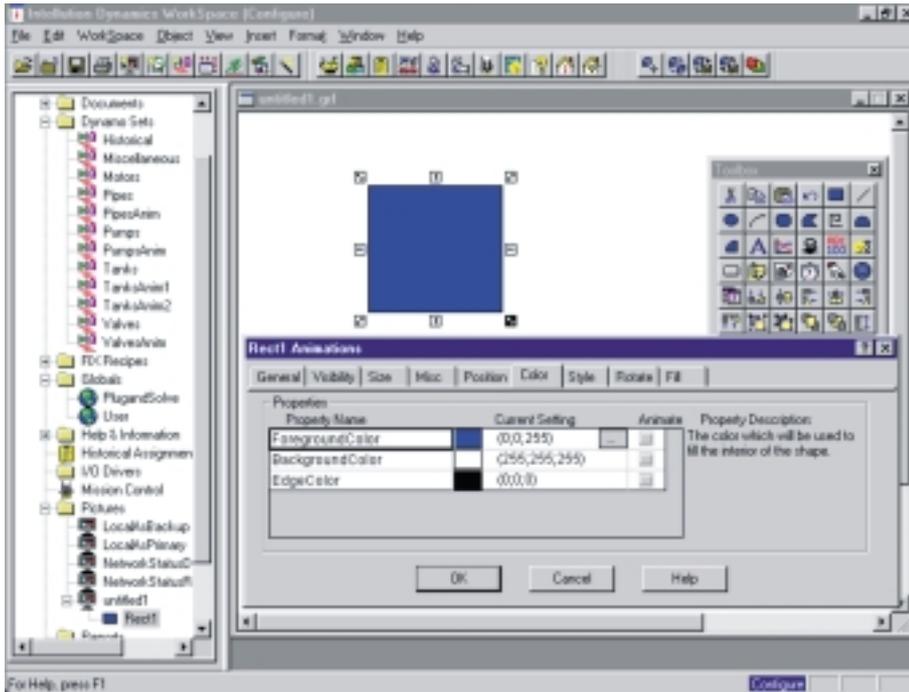


Рис. 2. Диалоговая панель свойств графического объекта iFIX. Слева находится системное дерево, справа виден набор инструментов Toolbox

случае, когда над проектом работают несколько человек и нужно найти источник ошибок.

● Оба пакета предоставляют пользователю богатые возможности по анимированию мнемосхемы. GENESIS32 имеет в своем арсенале такие оригинальные способы анимации, как параметрическая «заливка» контура произвольной формы методом усечения и движение объекта по заданной траектории (рис. 3).

Быстродействие графической подсистемы

Качественно выполненная технологическая мнемосхема обычно содержит большое количество динамических объектов. Такой объект представляет собой совокупность элементарных графических примитивов (линий, многоугольников, эллипсов и т.д.), объединенных в группы. Изменение параметров технологического объекта, состояние которого отображает мнемосхема, может привести к массовому перевыводу графических объектов. Поэтому большое значение имеет быстродействие экранной формы. Положение, при котором экранная форма отображает состояние объекта с запаздыванием, является нежелательным. Загрузка формы в режиме исполнения (run-time) также может рассматриваться как действие, критичное к затратам времени на выполнение.

При оценке приемлемости быстродействия графической подсистемы следует иметь в виду, что в составе SCADA она является далеко не единственным

ранный форма, содержащая однотипные объекты (прямоугольники). У данных объектов анимировалось одно из свойств (например изменение размера) путем привязки к дискретному тегу OPC-сервера. Затем измерялась периодичность обновления экранной формы в режиме run-time в обоих пакетах путем определения временного интервала, в течение которого объекты в экранной форме изменяли свое состояние 100 раз.

Если экранная форма не содержала сгруппированных объектов, то быстродействие графики обоих пакетов было одинаковым, то есть 100 изменений состояния объектов занимали одно и то же время. При объединении же объектов в группы быстродействие экранных форм GENESIS32 оставалось на прежнем уровне, тогда как iFIX значительно увеличивал период изменения состояния объектов. Большое влияние на быстродействие экранных форм iFIX оказывал уровень вложенности сгруппиро-

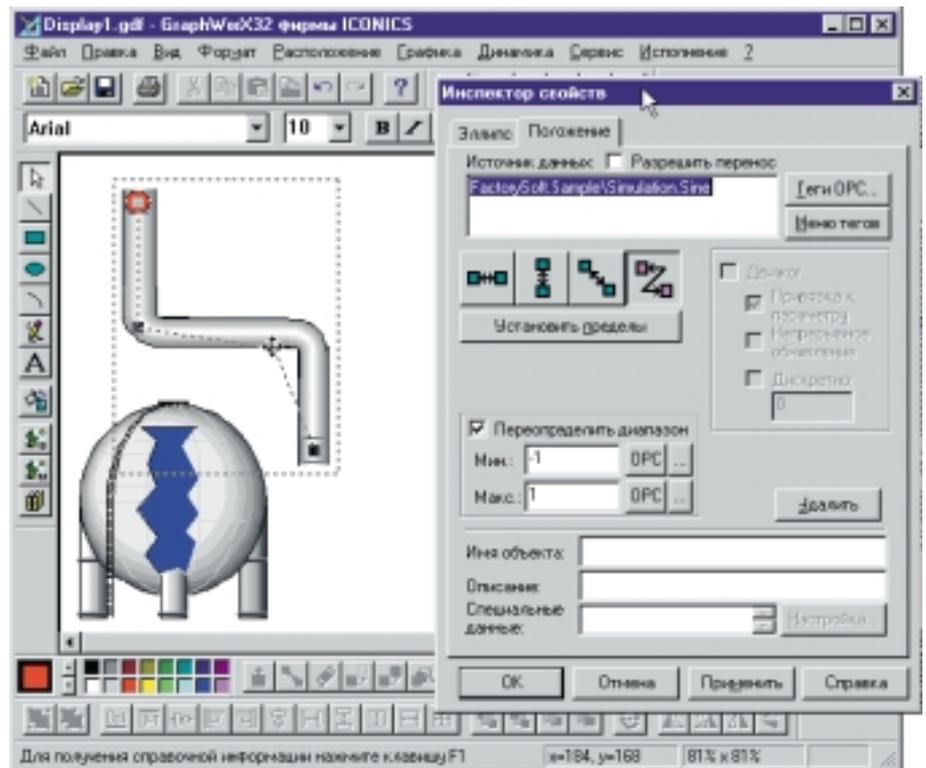


Рис. 3. Настройка траектории движения объекта в GraphWorX32

потребителем ресурсов. Солидным аппетитом отличаются системы архивирования, сетевого и межзадачного взаимодействия. Поэтому графическая подсистема должна не просто решать свои задачи, но и оставлять достаточно ресурсов компьютера для работы своих «партнеров».

Тестирование графических подсистем велось так: средствами рассматриваемых SCADA-пакетов создавалась эк-

ванных объектов. Если, например, в одной экранной форме мы имели 4 группы из 16 объектов, причем в каждой из групп было 4 уровня вложенности, а в другой — одну группу из 64 объектов с шестью уровнями вложенности, то последняя работала значительно медленнее первой (более чем в два раза).

Кроме этого, при большом количестве несгруппированных объектов (около 5000) iFIX, в отличие от GENESIS32, не



успевал осуществлять перевод всех объектов в пределах периода изменения привязанного сигнала. Тестирование показало также, что у iFIX время загрузки новой экранной формы в режиме исполнения в 1,5-2 раза больше, чем у GENESIS32 (при равной сложности экранных форм). Особенно явно данный эффект проявляется, если экранная форма содержит импортированную растровую графику в формате Windows Bitmap.

Ограничения на количество подобъектов и число уровней вложенности

Задачей данного теста было выявить, насколько глубокое вложение графических подобъектов допускают рассматриваемые SCADA-системы. Для этого создавалась экранная форма, содержащая однотипные объекты, анимировалось их свойство (в данном случае — изменение цвета) и форма переводилась в режим исполнения. Объекты помещались на экранную форму следующим образом: два объекта группировались, затем составлялась группа из двух таких объектов, и процесс рекурсивно повторялся. Было протестировано несколько экранных форм с разным числом объектов, максимальное из которых составило 2048. Согласно описанной методике тестирования, это соответствовало 10 уровням вложенности, что является вполне достаточным для мнемосхем сложных технологических объектов.

Каких-либо ограничений на количество подобъектов и число уровней вложенности ни для GENESIS32, ни для iFIX выявить не удалось. Оба пакета позволяют реализовать экранную форму с наибольшей степенью вложенности объектов. Тем не менее,

как уже было отмечено, число уровней вложенности оказывает значительное влияние на быстрдействие экранных форм iFIX.

Выводы

Оба рассматриваемых пакета имеют в своем составе мощные средства разработки операторского интерфейса, позволяющие отображать на экране достаточно сложные технологические объекты с высокой степенью адекватности.

Оба пакета поддерживают уровень вложенности объектов, достаточный для представления сложных автоматизируемых систем.

Быстрдействие GENESIS32 по сравнению с iFIX для сложных объектов (при наличии группировки, а также при большом количестве элементов) заметно выше.

Подсистема ввода-вывода

Наиболее перспективным и популярным на сегодняшний день способом интеграции подсистем разных изготовителей в составе крупномасштабной АСУ ТП является использование стандарта OPC.

В настоящее время практически все изготовители контроллеров и других средств промышленной автоматизации поставляют для своей продукции специальные программные драйверы, соответствующие спецификациям OPC.

Такой драйвер называется сервером OPC и является источником данных для клиентов OPC, таких, например, как графическая подсистема или подсистема архивирования (подробнее об OPC см. «СТА» 3/99). Естественно, нас интересовала полнота и эффективность поддержки технологии OPC рассматриваемыми пакетами.

Время установления связи с сервером

Это время является достаточно важным показателем, поскольку оно, например, определяет, насколько быстро созданная экранная форма отобразит реальные значения параметров. Со стороны GENESIS32 тестировался OPC-клиент GraphWorX32, а со стороны iFIX — OPC-клиент OPCDRV, являющийся посредником между OPC-серверами и базой данных iFIX.

Тестирование показало, что время установления связи с локальным (то есть работающим на том же компьютере, что и SCADA-система) сервером невелико и составляет не более 1 секунды для обоих пакетов.

При тестировании связи с удаленным (работающим на другом компьютере)

сервером мы запускали форму на исполнение непосредственно после перезагрузки Windows NT, а также повторно, и оценивали время появления реальных показаний.

В обоих случаях после перезагрузки ОС время появления реальных показаний достаточно велико и составляет десятки секунд. При повторном запуске формы реальные данные отображаются на ней по прошествии малого времени (того же, что и при связывании с локальным сервером).

Чтобы убедиться в том, что большое количество тегов OPC-сервера, к которому подключается SCADA, не приводит к существенным накладным расходам на поиск используемых тегов при установлении OPC-связей, запускалась экранная форма с 10 объектами, привязанными к десяти разным аналоговым тегам сервера, содержавшего 10 групп по 2539 тегов в каждой. Для обоих SCADA-пакетов время появления реальных показаний и в этом случае составляло не более 1 секунды.

Поддержка диагностической информации OPC

Как известно, спецификации OPC предусматривают в дополнение к соб-

ственно значению параметра и его метке времени передачу признака качества. По этому признаку SCADA-пакет как клиент OPC может обнаружить неисправность аппаратных средств ввода-вывода и даже получить от сервера более развернутую диагностическую информацию.

Для проверки того, как система визуализации реагирует на неисправность аппаратных средств, мы использовали ту же самую конфигурацию, что и в предыдущем случае, но в процессе работы физически разрывали связь с удаленным компьютером путем отсоединения сетевого кабеля от сетевого адаптера, а затем восстанавливали связь.

В случае GraphWorX32 сразу же после разрыва связи прекращается изменение показаний, а обрыв связи индицируется «потерей качества» тега через 20-30 секунд. При восстановлении связи «качество» тега восстанавливается также в среднем через 20-30 секунд. iFIX же обрыв связи никак (ждали 15 минут) не индицирует, хотя именно для этого в экранную форму был помещен объект, предназначенный для такой индикации. Наблюдается лишь замирание показаний. Если после таких манипуляций со связью (обрыв на десятки минут) по-

пытаться завершить выполнение iFIX, то иногда его компонент OPCDRV перестает отвечать на системные запросы, и в такой ситуации завершить его работу можно лишь путем снятия задачи диспетчером задач Windows NT.

Эффективность обработки большого числа параметров

Целью этого раздела программы тестирования было выяснить, не приводит ли большое количество (несколько сотен) установленных OPC-связей к значительным затратам вычислительных ресурсов.

Со стороны GENESIS32 тестировался GraphWorX32, со стороны iFIX — оба имеющихся в составе системы OPC-клиента (OPCDRV и встроенный в Workspace клиент), в качестве сервера использовался Fastwel UniOPC Server в режиме «console» (этот режим позволяет наблюдать за количеством и составом OPC-транзакций между сервером и клиентом). Экранная форма содержала объект, привязанный к изменяющему свое значение аналоговому тегу сервера. Объект обновлялся максимально быстро. Мы запускали форму на исполнение, меняли интенсивность изменения значения тега, пользуясь бланком



свойств тега, и наблюдали на консоли сервера за тем, сколько OPC-вызовов происходит в единицу времени и какие это вызовы (прямые или обратные).

Результаты экспериментов отражены в таблице 1.

Итак, OPC-клиент GraphWorX32 пользуется обратными вызовами, то есть получает новые данные «по подписке» лишь в том случае, когда они изменились. Интенсивность транзакций прямо пропорциональна динамике сигнала. Поэтому, если сигналов много, но их динамика мала, то интенсивность транзакций будет низкой и, соответственно, накладные расходы на OPC-взаимодействие окажутся незначительными.

Встроенный в Workspace клиент использует прямые вызовы. Интенсив-

ность их от динамики сигнала не зависит, а определяется лишь периодом обновления экранной формы. Таким образом, увеличение количества сигналов приведет к пропорциональному увеличению накладных расходов на OPC-связь.

Ситуация еще больше ухудшается, если OPC-сервер работает на удаленном компьютере. Прямые запросы большого числа неизменяемых параметров могут привести к загрузке сети бесполезными данными. Эту ситуацию иллюстрирую графики, приведенные на рис. 4.

В iFIX имеется возможность частично решить указанную проблему, задавая для разных частей экранной формы индивидуальные частоты обновления. Можно также отказаться от использова-

ния встроенного в Workspace клиента OPC, применяя входящий в состав iFIX драйвер OPCDRV, который позволяет получать данные от OPC сервера «по подписке». При этом практически решаются проблемы по неэффективному использованию сети, однако могут резко возрасти затраты вычислительных ресурсов (рис. 4) в связи с тем, что графическая подсистема iFIX получает данные от OPCDRV опосредованно — через базу данных. Кроме того, OPCDRV накладывает определенные ограничения на задаваемый пользователем период обновления данных OPC. Этот период может быть либо очень коротким (10 мс), либо достаточно длинным (>1 с).

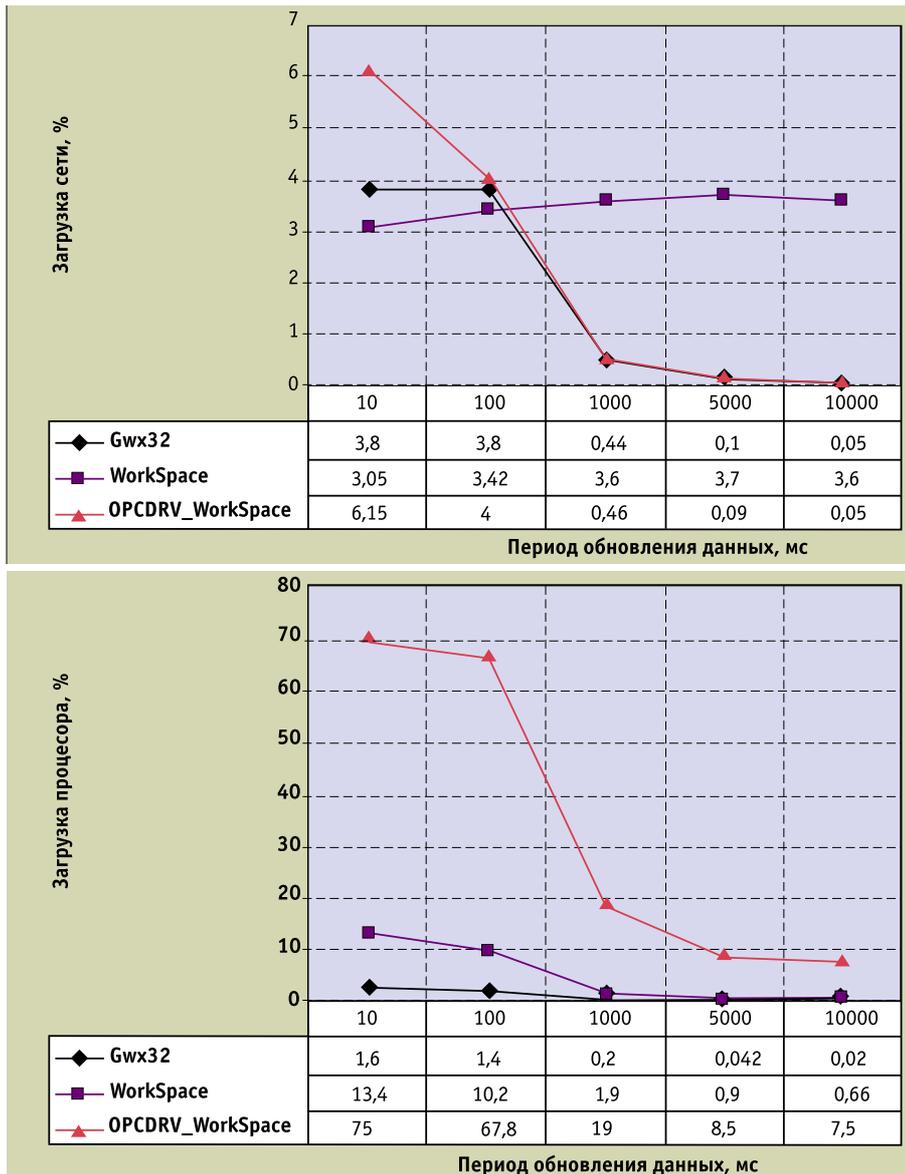
Следует отметить, что наличие общей базы данных является удобным решением для успешной разработки и сопровождения больших проектов. Поэтому для реализации таких проектов с помощью iFIX затраты ресурсов на прохождение данных через посредника — базу данных — могут рассматриваться во многих случаях как «необходимое зло».

Возможность работы с несколькими серверами

При решении задач комплексной автоматизации возникает необходимость

Таблица 1. Принципы взаимодействия с OPC-клиентами iFIX и Genesis32

Клиент	Количество OPC-транзакций за секунду			
	Постоянный сигнал		Случайный сигнал	
	Прямые обмены	Обмены «по подписке»	Прямые обмены	Обмены «по подписке»
GraphWorX32	0	0	0	20
Workspace	17,2	0	13,2	0
OPCDRV	0	1	0	100



Экранная форма содержит 75 элементов отображения типа «Параметр», соединенных с 75 элементами данных на удаленном сервере OPC. Приложения свернуты в пиктограмму. Период обновления экранной формы GraphWorX32 — 100 мс. Период обновления экранной формы iFIX WorkSpace, получающей данные от 75 тегов базы Database Manager iFIX, — 100 мс. OPCDRV подписан на теги UniOPC с запрошенным периодом обновления группы 10 мс. Database Manager получает данные от OPCDRV методом Exception Based.

Рис. 4. Зависимости загрузки сети и процессора от периода обновления данных на удаленном OPC-сервере

во взаимодействии SCADA-пакета с несколькими разнородными технологическими подсистемами. SCADA-пакет должен обеспечивать возможность одновременного подключения к OPC-серверам этих подсистем.

Мы запускали экранную форму, содержащую 6 объектов, каждый из которых был привязан к тегу одного из 6 разных OPC-серверов. И для GENESIS32, и для iFIX время появления показаний было невелико, системы вели себя нормально, сбоев не наблюдалось.

Дополнительная информация по результатам тестирования OPC-клиентов

- При старте iFIX запускаются требуемые OPC-серверы, а после его завершения они остаются запущенными и при попытке завершить их работу вручную выводят

сообщение о том, что к ним присоединен OPC-клиент. Это объясняется тем, что OPC-клиент WorkSpace при установлении связи с OPC-сервером, помимо основной OPC-группы с запрашиваемым интервалом обновления 10 мс, создает пустую группу с интервалом обновления 1000 мс, которая по какой-то причине не удаляется при отключении от сервера.

- GENESIS32 имеет OPC-браузер (рис. 5), позволяющий добраться до любого тега любого OPC-сервера, как на локальном компьютере, так и на любой рабочей станции в сети Microsoft. iFIX имеет подобный браузер, однако он работает только в пределах локального компьютера, и пользоваться можно только теми тегами, наименования которых не содержат кириллицы и пробелов. Если же нужно связаться с тегом удаленного сервера или если наименование тега содержит «запрещенные» символы, то использовать его можно, но значительно более сложным способом, через OPC-драйвер (OPCDRV). База данных этого драйвера содержит описания «виртуальных тегов», которые ссылаются на конкретные OPC-теги. Запись в БД iFIX, описывающая тег iFIX, должна ссылаться на источник данных — тег какого-либо драйвера, в случае же OPC она ссылается на «виртуальный тег» OPC-драйвера. И, наконец, какое-либо свойство объекта экранной формы может быть привязано к тегу iFIX из его БД. Таким образом, имеем три ссылки вместо одной, в связи с чем существенно возрастает объем работ по привязке SCADA-пакета к источникам данных.

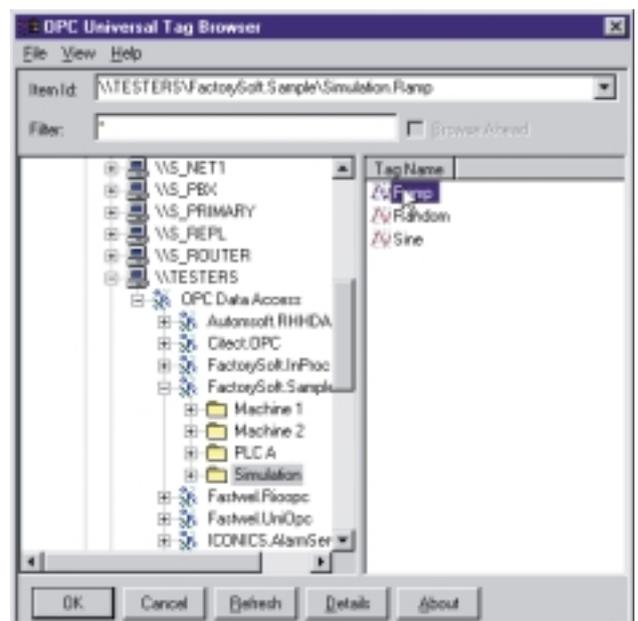


Рис. 5. Универсальный навигатор тегов OPC Universal Tag Browser, входящий в GENESIS32. Осуществляется связывание с тегом сервера OPC, расположенного на сетевом узле TESTERS



Выводы

Реализация OPC-клиента в GENESIS32 выглядит предпочтительнее, чем в iFIX, по следующим причинам:

- GENESIS32, в отличие от iFIX, диагностирует физический обрыв связи с компьютером, на котором работает удаленный OPC-сервер, и оповещает оператора об этом;
- интерфейс браузера OPC-тегов значительно более удобен и «прозрачен», за счет чего уменьшается объем рутинной работы по привязке SCADA-системы к источникам данных;
- более качественно реализована процедура отключения от OPC-сервера при завершении работы системы;
- GENESIS32, реализуя OPC-взаимодействие, потребляет несколько меньше ресурсов компьютера.

Встроенная среда разработки и исполнения сценарных процедур

Как показал опыт, результативность разработки любого более или менее сложного проекта с использованием SCADA-системы в конечном итоге в достаточной степени зависит от функциональных возможностей встроенного в SCADA скриптового языка. Среда программирования должна позволять разработчику выполнить то, что невозможно или нецелесообразно делать, используя типовой набор инструментальных средств SCADA. Столкнувшись с отсутствием стандартного механизма решения некоторой задачи, пользователь естественным образом обращает свой взор на возможности скриптового языка.

Производители GENESIS32 и iFIX для решения своих задач велосипед изобретать не стали, а воспользовались существующим решением — в качестве скриптового языка в обеих SCADA используется Microsoft Visual Basic for Applications (MS VBA).

Представление проекта в целом, а также всех его внутренних элементов в виде объектной модели, на работу с которой ориентирован MS VBA, дает пользователю гибкий и эффективный механизм доступа практически к любым ресурсам SCADA. Интегрированная среда разработки Visual Basic-приложений (и в iFIX, и в GENESIS32 используется тот же VBA, что и в Microsoft Office), всплывающие подсказки, демонстрирующие синтаксис вызова процедур, списки доступных свойств объектов, система просмотра библиотек объектов (Object Browser), встроенная справочная система и отладчик — все это значительно повышает удобство написания программ.

Немаловажным фактором является и то, что Basic традиционно является од-

ним из наиболее легких в освоении языком программирования.

Общая оценка возможностей

В результате тестирования мы убедились, что в обеих SCADA доступны все стандартные средства MS VBA:

- редактор программного кода VBA;
- система просмотра объектов и их свойств;
- контекстная справочная система по VBA (iFIX предоставляет также контекстную справку по использованию своих внутренних объектов);
- возможность отладки сценариев с использованием точек останова, трассировки программы, инспектора свойств объектов;
- возможность импорта/экспорта программных модулей;
- возможность добавления в VBA-проект пользовательских форм.

Кроме основного своего назначения — реализации пользовательских сценарных процедур, когда SCADA находится в режиме исполнения, средства разработки обеих систем позволяют с помощью скрипт-кодов автоматизировать разработку проекта в режиме конфигурирования. VBA-процедура, реализующая набор типовых операций по конфигурированию какого-либо элемента экранной формы, написанная пользователем, может существенно упростить работу над проектом. iFIX для этих целей дает возможность создания пользовательских панелей инструментов, к кнопкам которых разработчик может привязать собственный скрипт-код. В GENESIS32 эта задача решается несколько менее очевидным способом: кнопка с привязанной к ней процедурой помещается в библиотеку символов и в дальнейшем используется при разработке других проектов. Из этого следует, что, помимо «программы минимум» — предоставления пользователю возможности реализовать с помощью скриптов все то, что не позволяют реализовать другие средства разработки, системы разработки и исполнения сценарных процедур GENESIS32 и iFIX выполняют и более глобальную задачу — дают возможность заменить VBA-кодом стандартные операции по конфигурированию системы.

Одной из наиболее распространенных задач, встречающихся при разработке проекта, является реакция на действия пользователя, связанные с каким-либо внутренним объектом экранной формы, например, нажатие виртуальной кнопки. Обе SCADA поддерживают обработку таких событий, как одиночный и двойной щелчок мыши или нажатие клавиши на клавиатуре, iFIX, кроме

того, обрабатывает перемещения курсора мыши и переход в режим редактирования (конфигурирования) объекта.

В iFIX реакция на событие всегда задается в виде VBA-скрипта. GENESIS32 позволяет либо сделать то же самое, либо без создания скрипт-программы привязать к событию одну из стандартных реакций (таких, например, как загрузка экранной формы, запуск внешнего приложения, перепривязка псевдонимов и т.п.) путем выбора из списка.

Для привязки обработчика события к объекту экранной формы в iFIX достаточно нажать над этим объектом правую кнопку мыши и в появившемся меню выбрать пункт «Edit Script» («Редактировать VBA-скрипт»). При этом автоматически запускается редактор Visual Basic, а в списке доступных объектов появляется имя объекта, с которым производится действия. Необходимое событие, обработку которого требуется произвести из VBA-кода, можно выбрать из выпадающего списка методов данного объекта. Последовательность перечисленных операций иллюстрируется рис. 6–7.

В GENESIS32 для привязки скрипта к объекту выполняются аналогичные операции, как показано на рис. 8–9. Однако код, связанный с объектом, помещается в отдельный модуль и не содержит predeterminedных обработчиков событий, происходящих с объектом.

Для привязки скрипта к событию, связанному с графическим элементом, необходимо выбрать элемент панели инструментов «Pick» («Указание и щелчок»), в появившемся бланке указать необходимое событие, вид реакции на него определить как «Run VBA Script» («Выполнить сценарий VBA») и выбрать или задать имя процедуры-обработчика, после чего соответствующая процедура становится доступной в редакторе VBA.

Итак, создание скриптового обработчика и привязка его к событию в iFIX осуществляется более простым и удобным способом, чем в GENESIS32. С другой стороны, GENESIS32 позволяет реализовывать типовые функции SCADA-системы при щелчке мышью над графическим объектом специалистами, вообще не знакомыми с языками программирования. Набор predeterminedных функций обработки события, генерируемого системой при щелчке мышью над графическим объектом, показан на рис. 10.

Быстродействие исполняющей системы

Оба SCADA-пакета тестировались на скорость исполнения скриптов с целью выяснить, насколько быстро выполняется VBA-код в режиме исполнения. При тес-

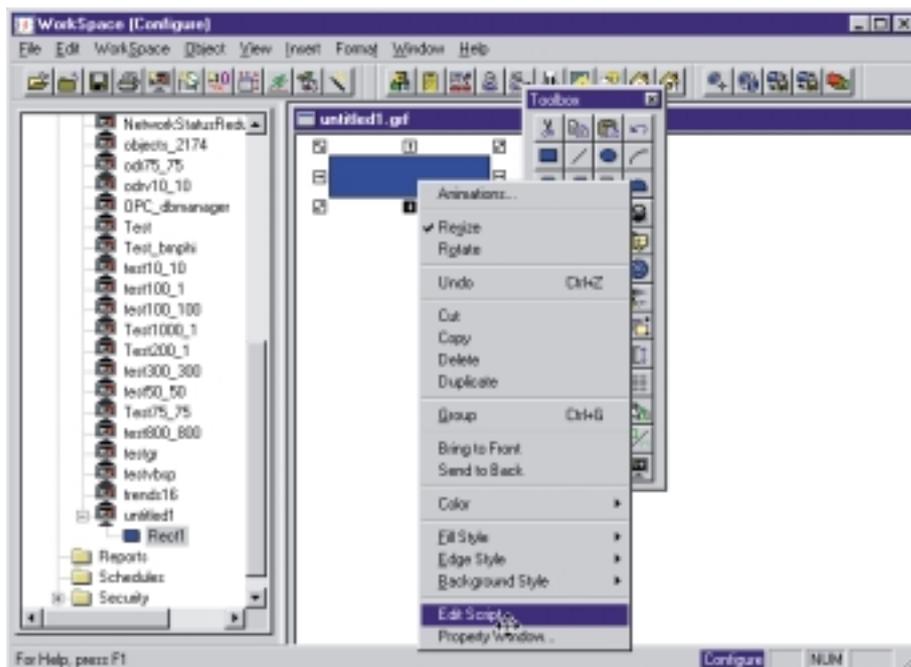


Рис. 6. Переход в IDE VBA для редактирования сценариев, связанных с выделенным объектом в iFIX

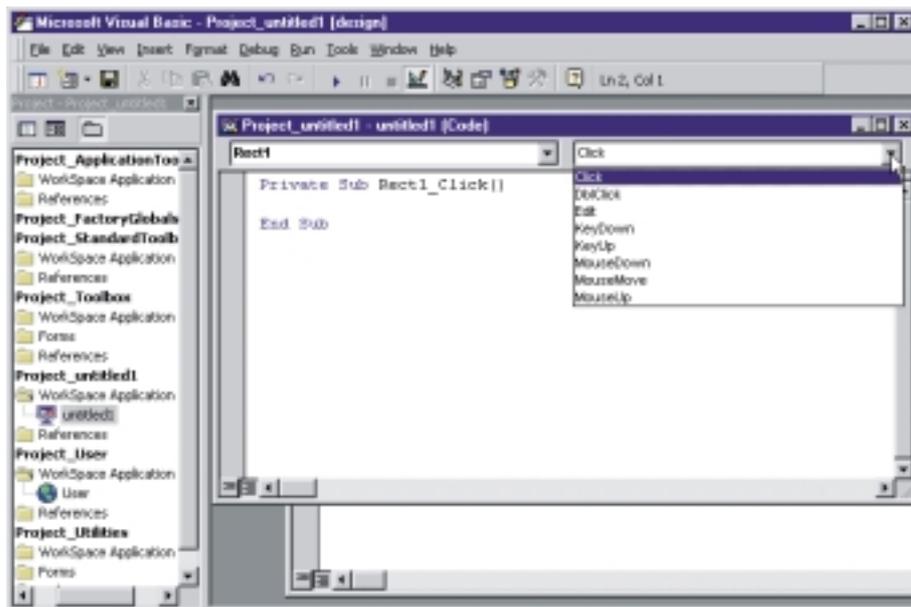


Рис. 7. Среда IDE VBA в iFIX. События, связанные с выбранным графическим объектом, доступны в выпадающем списке событий

тировании решено было использовать только стандартные операции Visual Basic, поскольку взаимодействие с внутренними элементами SCADA, а также с внешними приложениями было вынесено в отдельные пункты программы тестирования. На данном же этапе определялась исключительно «скорострельность» кода, использующего только стандартные языковые конструкции. Эксперимент, состоящий в измерении времени сортировки методом «пузырька» числового массива из 20000 элементов, предварительно упорядоченного в обратном порядке, показал, что быстродействие скриптов в обеих системах одинаково. Абсолютным

же показателем явилось время работы скрипта, составившее в обоих случаях 47 секунд. С учетом того, что количество элементарных операций, требующихся для сортировки массива, пропорционально квадрату его размерности, результаты теста показывают вполне достаточную для решения большинства задач скорость исполнения скрипт-кода.

Доступность внутренних объектов приложения и объектов ActiveX

Эта проверка производилась с целью определения удобства, полноты и эффективности (с точки зрения скорости)

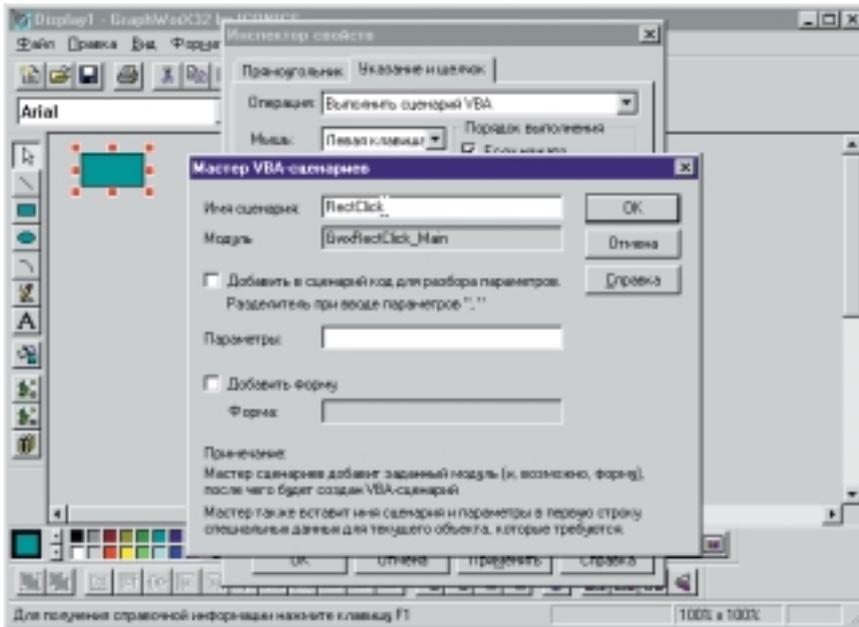


Рис. 8. Создание VBA-сценария, связанного с выделенным объектом в GraphWorX32

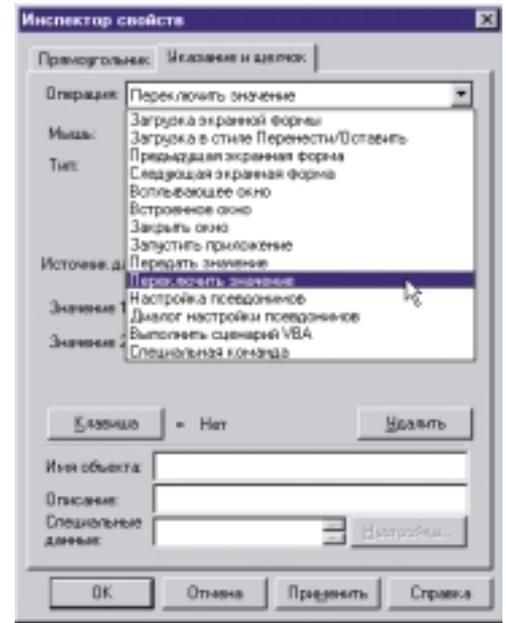


Рис. 10. Предопределенные обработчики события «Указание и щелчок»

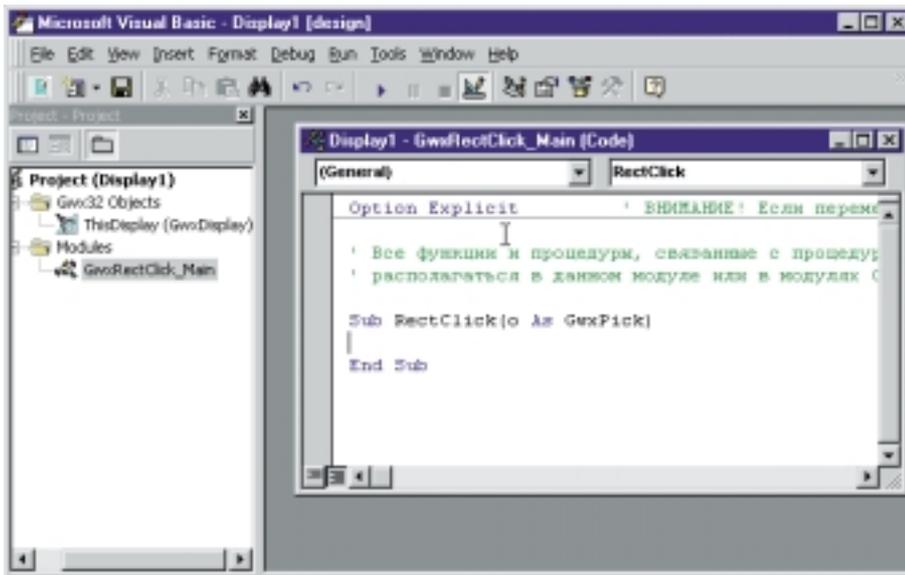


Рис. 9. Среда IDE VBA в GraphWorX32. Код помещен в отдельный модуль

реализации доступа сценарной процедуры к элементам экранной формы.

В iFIX для разрешения видимости в скриптовых процедурах стандартных внутренних элементов, таких как графические примитивы, элементы просмотра данных, графики и т.д., необходимо выбрать пункт «Edit Script» в контекстном меню, доступном по нажатию правой клавиши мыши над соответствующим объектом. Возможно также разрешение видимости сразу для целой группы объектов аналогичным образом. После выполнения данной процедуры становится возможным доступ к объекту из VBA-процедур непосредственно по имени. Поскольку iFIX именуется все объекты автоматически («Rect1», «Rect2», «Chart1», «Chart2» и т.д.), то ника-

ких дополнительных усилий по разрешению доступа к объекту из VBA от пользователя не требуется. При изменении пользователем имени объекта автоматически обновляются ссылки на него в скриптах (кроме некоторых случаев, когда система не может распознать обращение к данному объекту).

Таким образом, iFIX обеспечивает для VBA-скриптов доступ к любому внутреннему объекту, не требуя от разработчика каких-либо дополнительных действий, однако для этого iFIX регистрирует класс каждого своего графического объекта в реестре Windows. В результате при запуске iFIX происходит загрузка большого количества библиотек динамической компоновки, содержащих интерфейсы диспетчеризации к

графическим объектам всех классов (экран, прямоугольник, дуга, овал, линия и т.д.), что повышает требования к ресурсам системы.

GENESIS32 предоставляет другой механизм доступа к внутренним элементам экранной формы (графики и «тревоги» не входят в это число, поскольку являются объектами ActiveX). Для получения доступа к объекту из VBA-сценария его необходимо предварительно поименовать внутри диалогового окна свойств объекта, доступного из контекстного меню. Автоматическое именование одноименных объектов в GENESIS32 осуществляется путем задания одному из объектов шаблонного имени в формате NAME_#, где NAME - имя объекта. При последующем тиражировании объектов данного вида к имени добавляется порядковый номер («давление_котла_1», «давление_котла_2» и т.д.). Ссылку на поименованный объект из VBA-кода можно получить посредством вызова одной из следующих функций: GetVisibleObjectFromName, GetDynamicObjectFromName, GetPointObjectFromName. Функция GetVisibleObjectFromIndex позволяет получить доступ к объекту по индексу, что

удобно при унифицированной обработке множества элементов экранной формы.

На первый взгляд кажется, что дополнительная работа по именованию объектов создает для разработчика определенные неудобства. Однако делать это нужно только для объектов, к которым необходим доступ из VBA-скриптов (а это, как правило, только часть объектов экранной формы), к тому же, при этом разработчик присваивает объекту определен-



ное смысловое значение, которое облегчает в дальнейшем разработку скрипта. Ведь очевидно, что при программировании гораздо труднее ошибиться, оперируя такими понятиями как «индикатор перегрева» и «уровень в норме», чем безликими «Rect1» и «Rect2».

Доступ к внутренним элементам из VBA-кода в iFIX осуществляется непосредственно по имени объекта. В GENESIS32 необходимо сначала получить ссылку на объект, а затем уже работать с ним через эту ссылку.

Как показали тесты, совместное выполнение получения ссылки и доступа к объекту посредством этой ссылки в GENESIS32 занимают столько же времени, сколько обращение по имени в iFIX (на компьютере с процессором Celeron-300 оно составило 15 мкс).

За счет обращений к объекту по имени текст программы в iFIX, как правило, более нагляден. Но доступ к объекту в GENESIS32 по уже существующей, например, полученной при инициализации ссылке выполняется примерно в 5 раз быстрее, чем доступ по имени в iFIX.

Доступ к объектам ActiveX в обеих SCADA реализован одинаково, — по автоматически генерируемому системой имени объекта. ActiveX-объекты становятся «видны» в VBA-процедурах сразу после вставки в экранную форму и спе-

циального разрешения видимости не требуют.

Обмен с внешними приложениями

Целью одного из этапов тестирования была проверка взаимодействия приложения SCADA-системы с внешним приложением с помощью стандартных функций VBA GetObject и CreateObject. Данные функции позволяют посредством механизма OLE получить доступ к внешнему приложению, которое представляет собой объект ActiveX, сохраненный в файле. Такими приложениями являются, например, продукты семейства Microsoft Office: Word, Excel, Access.

Ссылка на объект-приложение, полученная с помощью функций GetObject/CreateObject, позволяет работать с внешним приложением, как с обычным объектом VBA. Таким образом пользователь имеет возможность, например, наладить импорт/экспорт данных в MS Excel или же получение/передачу текстовой информации в MS Word. Опыт разработки проектов автоматизации показывает, что во многих случаях подобные задачи бывают достаточно актуальны для конечного пользователя.

На этапе проверки возможности доступа из VBA-скрипта к объектам внеш-

них приложений (в качестве таковых выступали Word и Excel) никаких особенностей реализации данной функции ни в iFIX, ни в GENESIS32 отмечено не было. Обе системы позволяют осуществить взаимодействие с программными продуктами, поддерживающими данную технологию, включая загрузку/выгрузку приложения, получение доступа к уже запущенному приложению, создание нового или открытие существующего документа (или таблицы Excel), передачу и получение данных, сохранение измененных документов на диске и т.п.

Вторая часть эксперимента была направлена на решение обратной задачи: получение из скрипт-кода внешнего приложения доступа к объекту, представляющему собой SCADA-проект. Необходимость такого рода действий может возникнуть в том случае, если требуется осуществить доступ к приложению SCADA-системы из программы, написанной на другом языке программирования, например Visual C. Кроме того, подобный тест позволяет более качественно оценить полноту и корректность реализации механизма обмена с внешними приложениями, поддерживающими технологию OLE. Проверка показала, что в обоих продуктах функции GetObject/CreateObject корректно работают и «в обратную сторону» — они позволяют получить доступ ко всем объ-

ектам экранной формы из внешнего приложения. Единственным нюансом явилась невозможность запуска iFIX и из внешнего VBA-кода, что обусловлено, по всей видимости, тяжеловесностью iFIX.

Проведенный анализ возможностей среды разработки и исполнения сценарных процедур подтвердил, что встроенный в iFIX и GENESIS32 скриптовый язык, каковым является MS VBA, предоставляет разработчику гибкие и удобные средства для написания и отладки сценариев, позволяет быстро и эффективно получать доступ из сценарных процедур как к внутренним элементам экранной формы, так и к объектам внешних приложений. Обе SCADA-системы имеют свои плюсы и минусы в реализации некоторых свойств среды программирования, которые были отмечены ранее, однако механизм поддержки скриптов в обоих продуктах следует признать мощным и отвечающим современным направлениям развития программного обеспечения.

Взаимодействие с СУБД и другими внешними приложениями

АСУ предприятия — сложный и многоплановый комплекс, в котором SCADA — лишь один из компонентов. Одной из важнейших характеристик, определяющих применимость конкретной SCADA в больших проектах, является возможность их интеграции с продуктами смежного назначения (СУБД, генераторами отчетов, бухгалтерскими и складскими программами, электронными таблицами, текстовыми процессорами и т.д.).

Взаимодействие с внешними СУБД

Объект управления обслуживают разные категории специалистов. Если сменный персонал работает, главным образом, с текущими значениями параметров, то инженера-исследователя зачастую интересуют данные, накопленные за некоторый период времени. Очевидно, что невыгодно занимать ресурсы приложения реального времени для задач детального анализа исторических данных, кроме того, для анализа исторических данных лучше всего пользоваться приложениями, имеющими встроенные средства статистической обработки и представления данных в виде графиков и диаграмм. Взаимодействие с внешними СУБД нас интересовало, в первую очередь, с точки зрения возможности хранения данных в форматах, удобных для дальнейшей работы с ними сторонних приложений.

Для реализации проектов с большим количеством архивируемых параметров в состав пакета GENESIS32 входит сервер архивации и доступа к историческим данным TrendWorX32 SQL Data Logger, который позволяет сохранять информацию в базах данных MS Access и MS SQL Server. Данные для архивирования описываются по группам аналогично описанию их в сервере фоновой буферизации Background Trending, который применяется в проектах с небольшой или средней информационной емкостью. Группа SQL Data Logger представляет собой объединение контролируемых параметров (сигналов), для которого выполняется общий алгоритм накопления и архивации данных. Для ускорения записи и обеспечения целостности хранения данные сохраняются в таблицах в особом формате, предназначенном не для непосредственной работы, а главным образом для последующего просмотра данных в TrendWorX32 с использованием режима истории, а также преобразования их с помощью компонента TrendWorX32 Reporting или элемента ActiveX SQLTool Control в таблицы, более удобные для дальнейшей работы.

Программный продукт iFIX ODBC обеспечивает коммуникационный об-

мен между реляционными базами данных и базой данных iFIX.

iFIX ODBC состоит из

- задачи SQL,
- блока базы данных SQL Trigger (SQT),
- блока базы данных SQL Data (SQD).

Задача SQL выполняет следующие функции:

- исполняет блоки SQT, которые запускают ODBC-обмен,
- получает данные о процессе от блоков SQD и вводит эти данные в реляционную базу данных,
- выбирает данные из реляционной базы данных и записывает эти данные обратно в базу данных iFIX,
- производит резервирование данных в случае отказа сети с последующим их восстановлением.

Блоки SQD и SQT реализуют вызов команды SQL с передачей ей конкретных данных.

Поддержка многих реляционных баз данных позволяет пользователю, знакомому с языком SQL, к примеру, хранить все команды SQL в одной базе данных и исполнять эти команды в нескольких различных базах данных.

Помимо команд SQL, iFIX позволяет выполнять хранимые процедуры реляционных баз данных. Процедуры явля-

ются более быстродействующими, чем команды SQL.

Таким образом, и GENESIS32, и iFIX поддерживают взаимодействие с внешними СУБД на уровне, достаточном для решения большинства задач АСУ ТП (главным образом, для хранения исторических данных). При этом для взаимодействия с внешними базами данных пользователь iFIX должен, как минимум, обладать квалификацией, достаточной для написания SQL-скриптов, выполняющих сохранение данных с заданными интервалами, управление таблицами и извлечение данных. Зато это позволит ему сформировать таблицу в удобном для себя виде. Пользователь GENESIS32 может обойтись без написания SQL-скриптов, но при этом ему придется довольствоваться теми форматами таблиц представления архивных данных, которые реализует TrendWorX32 Reporting (в большинстве случаев этого достаточно), либо писать код на VBA для элемента ActiveX SQLTool Control, используя его свойства и методы по извлечению данных. Последний способ извлечения данных, сохраняемых в реляционных базах сервером TrendWorX32 SQL DataLogger, состоит в разработке программы на VB/VBA или Visual C, использующей подключение к объекту доступа к данным через OLE DB.

Взаимодействие с другими классами приложений

Взаимодействие SCADA-систем со сторонними приложениями служит, главным образом, для решения задач анализа, документирования информации об объекте управления, а также для обеспечения производственно-хозяйственной деятельности.

В обеих тестируемых SCADA доступ к внешним приложениям удалось получить средствами VBA. Кроме того, в GENESIS32 генерация и печать отчетов может автоматически выполняться с помощью компонента TrendWorX32 Reporting, который позволяет извлекать данные из баз, созданных TrendWorX32 SQL DataLogger, и представлять их в отчетах табличной формы MS Excel и базах MS Access и MS SQL Server.

iFIX, помимо использования возможностей VBA, имеет набор собственных средств планирования и документирования (планировщик, генератор отчетов, служба регламентов).

Кроме этого, iFIX Workspace позволяет интегрировать ActiveX-документы Word и Excel в рабочую зону. При открытии этих документов отображаются соответствующие меню и панели инструментов.

Таким образом, и GENESIS32, и iFIX обеспечивают возможность эффективного взаимодействия с другими классами приложений.

Поддержка ActiveX

Поддержка ActiveX-технологии позволяет создавать экранные формы, состоящие из ранее разработанных компонентов, как из кубиков. В настоящее время формируется открытый рынок ActiveX-компонентов и появляются фирмы, специализирующиеся на выпуске подобной продукции. Это позволяет системному интегратору использовать в рамках современной базовой SCADA-системы дополнительные ActiveX-компоненты, разработанные самостоятельно или приобретенные у независимых поставщиков.

Обе SCADA-системы поддерживают технологию ActiveX, однако поскольку ActiveX-компонент является достаточно сложным программным модулем, в его коде могут присутствовать ошибки. Мы ставили перед собой задачу установить, как каким последствиям для системы в целом приводят ошибки во время выполнения вставленного ActiveX-компонента.

Со стороны GENESIS32 тестировался GraphWorX32, со стороны iFIX — Workspace, в качестве генератора ошибок использовался ActiveX-компонент собственной разработки. Экранная форма содержала вставленный генератор ошибок. Мы запускали форму на исполнение и наблюдали за реакцией системы на запись по нулевому адресу, на запись по случайному адресу, на порчу указателя стека и на зацикливание.

При генерации записи по нулевому адресу, генерации записи по случайному адресу и порче указателя стека GraphWorX32 выполнял недопустимую операцию и завершался аварийно, Workspace же выдавал сообщение о сбое в работе компонента, блокировал его и продолжал работать.

Однако существует ситуация, когда и Workspace не справляется с ошибкой, возникающей в ActiveX-компоненте. Это выяснилось, когда в экранную форму Workspace был вставлен ActiveX-компонент, выполняющий запись данных по произвольным адресам приложения-контейнера. Возникающее при этом исключение обрабатывалось тем же самым ActiveX-компонентом, но обработчик не содержал каких-либо осмысленных действий, а продолжал «портить» память контейнера. В результате нажатие кнопки, реализующей интерфейс пользователя для этого ActiveX-компонента, приводило к про-

паданию приложения-контейнера (оно исчезало из памяти и с экрана, оставляя все связанные с ним объекты «неотпущенными»).

При зацикливании и GraphWorX32, и Workspace переставали на что-либо реагировать, их состояние диспетчер задач обозначал как «Не отвечает», и завершить их можно было только из диспетчера задач.

Итак, с точки зрения устойчивости к ошибкам в реализации ActiveX-компонентов, iFIX выглядит предпочтительнее, однако необходимо помнить, что и в случае GENESIS32, и в случае iFIX для возобновления нормальной работы ActiveX-компонента необходимо полностью перезапускать приложение. Кроме того, технология ActiveX, в том числе технология «безопасного контейнера» (safe container), не позволяет гарантированно защитить контейнер от краха в случае сбоя во вставленном ActiveX-компоненте. Поэтому мы настоятельно рекомендуем тщательно тестировать ActiveX-компоненты перед их использованием на «живом» объекте.

Управление доступом

Известно, что если с системой работают пользователи различных категорий, то неизбежно встает вопрос об организации контроля и управления доступом к объектам системы. Наличие системы контроля и управления доступом позволяет не только ограничивать доступ пользователей к объектам системы, но и отслеживать сделанные пользователем изменения, заставляя его регистрироваться для работы в системе. Поэтому в процессе тестирования нами рассматривались такие возможности SCADA-пакетов в области контроля доступа, как введение различных категорий доступа, контроль лиц, осуществлявших доступ, протоколирование внесенных изменений, а также средства защиты протокола и паролей от фальсификации.

Результаты тестирования показали, что оба рассматриваемых продукта включают в себя систему управления доступом, которая позволяет вводить различные категории доступа и ограничивать доступ к функциям пакета для пользователей, зарегистрированных в системе. С помощью специальной программы конфигурации защиты можно назначать права пользователей и групп пользователей, имена входа и пароли (рис. 11). Права пользователя определяются его персональным уровнем привилегий, а также уровнем привилегий, назначенным группе, в которую он входит.

Так, каждое приложение GENESIS32 предоставляет список операций, доступ

к выполнению которых может быть ограничен (рис. 12). Например, можно запретить оператору добавлять новые пerryя в график, завершать работу приложения или переключаться из режима исполнения в режим разработки.

Имеется возможность запретить доступ к отдельным файлам и группам файлов, к записи значений в теги OPC.

Защита пакета iFIX позволяет управлять доступом к приложениям и файлам iFIX, к критичным для процесса программным функциям (например к перезагрузке базы данных), к записи в базу данных. Существует возможность ограничить доступ к редактору Visual Basic. Отличительной особенностью защиты iFIX является возможность определить имя и пароль пользователя iFIX такими же, как входное имя и пароль пользователя Windows NT (рис. 13). В этом случае iFIX использует в качестве списка входа список учетных записей пользователей Windows NT.

Задача контроля лиц, осуществлявших доступ к объектам системы, решена в тестируемых продуктах по-разному. В пакете GENESIS32 вход пользователя в систему с регистрацией осуществляется с помощью утилиты Login. Эта утилита позволяет также получить информацию о тех пользователях, которые зарегистрированы в системе в данный момент времени. Анализ фактов входа и выхода из системы осуществляется путем просмотра системного журнала Windows NT и/или базы данных MS Access, которая может вестись регистратором событий AlarmWorX32 Logger.

Что касается пакета iFIX, пользователи могут входить в систему не только вручную (с помощью программы входа Login), но и автоматически (если была создана конфигурация автоматического входа). Контроль же осуществляется с помощью собственного средства iFIX — контрольного журнала защиты. В файле контрольного журнала хранится информация о входах и выходах из системы, о неудачных попытках входа в систему, о попытках получить несанкционированный доступ.

iFIX предоставляет возможность регистрировать действия оператора и в качестве тревог рассылать сообщения об этих действиях адресатам тревог. Протоколирование изменений, вносимых

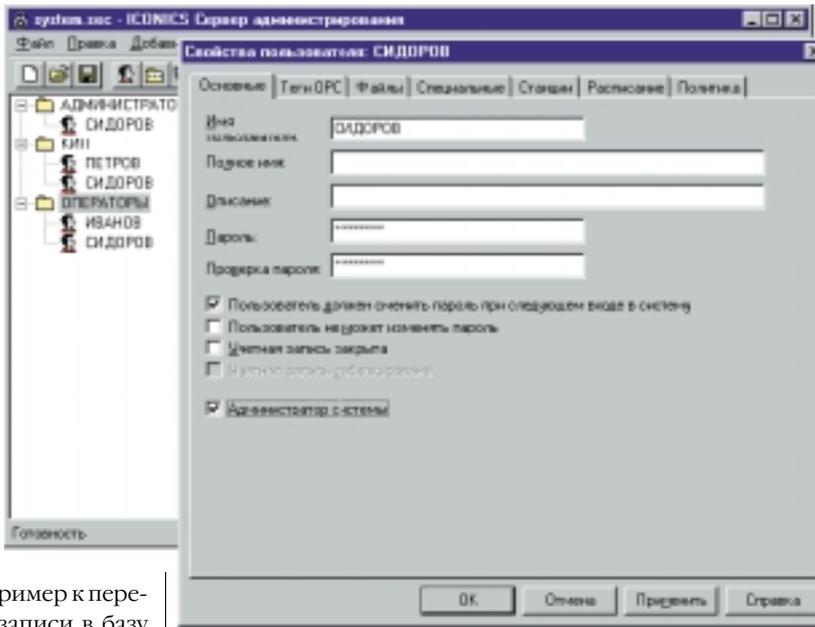


Рис. 11. Настройка прав пользователя в GENESIS32

оператором, осуществляется с помощью сервиса тревог Alarms ODBC Service. Все соответствующие тревоги и сообщения записываются в указанную в процессе конфигурирования сервиса базу дан-

ных. В дальнейшем информация из этой базы данных доступна для просмотра. Защитить протокол, то есть базу данных, в которую записываются тревоги и сообщения, можно средствами соответствующей СУБД. В пакете GENESIS32 изменения значений параметров в серверах OPC, произведенные зарегистрированным в системе пользователем, могут просматриваться в журнале событий и тревог AlarmWorX32 Viewer ActiveX, который позволяет организовать подписку на события, рассылаемые

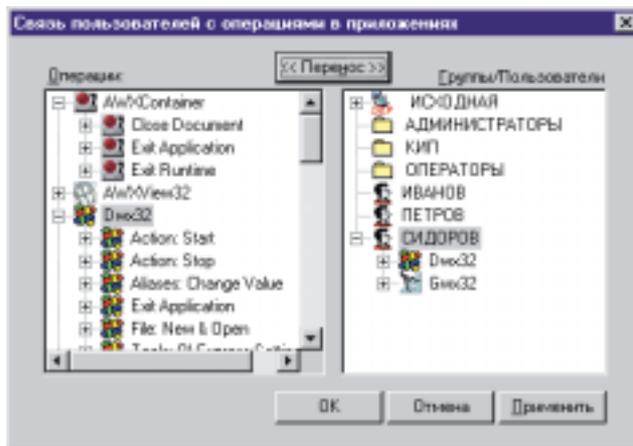


Рис. 12. Настройка доступа к операциям в приложениях GENESIS32

сервером событий ICONICS Event Server. Эти же события могут автоматически регистрироваться в базах данных MS Access и/или MS SQL SERVER при помощи регистратора AlarmWorX32 Logger. Что же касается политики в отношении паролей, то для защиты паролей пользователей от фальсификации подсистема контроля доступа GENESIS32 позволяет установить некоторые правила использования паролей. Например, можно запретить изменять пароль средствами утилиты Login. Также возможно установить временные ограничения на продолжительность

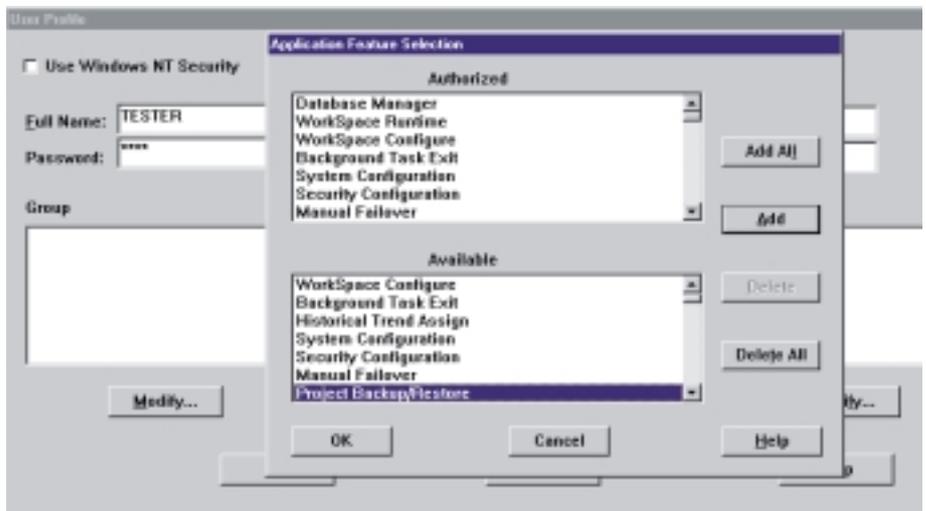


Рис. 13. Настройка доступа к операциям в iFIX

действия пароля, задать минимальное допустимое число символов в пароле. Если говорить о защите паролей в iFIX, то разработчики этой системы сделали акцент на встроенные в Windows NT средства организации парольной защиты.

Еще одним важным аспектом защиты от несанкционированного доступа является противодействие некорректным действиям оператора на уровне операционной системы.

Обычно для устойчивой работы АРМ оператора необходимо запретить как завершение самого SCADA-пакета, так и запуск других приложений.

Как правило, основная обязанность оператора состоит в наблюдении за технологическим процессом и принятии необходимых мер в случае отклонения хода процесса от нормы, о чем, согласно заданным на стадии разработки инструкциям, его уведомляет система. От оператора требуется, чтобы он обладал минимальными навыками работы с компьютером, но он отнюдь не обязан быть «продвинутым» пользователем. Следовательно, большое значение приобретает контроль за действиями оператора, дабы он не смог совершить необратимых действий, ведущих к неприятным последствиям для автоматизируемого объекта. Как показывает практика, помимо того что оператор может нажать не на ту кнопку, очень часто компьютер с запущенной на нем пультовой программой используется не по назначению (например для игр). При этом возможна ситуация, когда оператор увлечется игрой настолько, что пропустит сообщение о неисправности или аварийной ситуации.

GENESIS32 позволяет запретить конкретному пользователю переключаться из режима исполнения (run-time) в режим разработки (configure) и завершать приложение при помощи кнопки закрытия окна, системного меню и комбинации клавиш <Alt+F4>.

Если оператор не владеет клавиатурным интерфейсом, а использует только мышь для получения доступа к меню «Пуск» и переключения между задачами, то в случае настройки приложения для работы в полноэкранном режиме он не сможет запускать какие-либо посторонние задачи и переключаться между работающими приложениями.

Однако в общем случае внутренними средствами GENESIS32 не обеспечивается запрет запуска приложения через меню «Пуск» (<Ctrl+Esc>), а также переключение на другие приложения с помощью комбинации клавиш <Alt+Tab> и диспетчера задач.

Если для Windows NT 4.0 установлен Service Pack не ниже 4, то в дополнение к сказанному iFIX позволяет внутренними средствами блокировать доступ оператора к диспетчеру задач Windows NT с использованием комбинации клавиш <Ctrl+Alt+Del> и, следовательно, оператор не имеет возможности закрыть приложение с помощью диспетчера задач.

Кроме того, iFIX позволяет заблокировать запуск посторонних приложений или переключение на них с помощью комбинаций <Ctrl+Esc> и <Alt+Tab>.

Следует отметить, что почти все дополнительные «запреты» (кроме запрета вызова диспетчера задач по <Ctrl+Alt+Del>), которые удобным для разработчика способом накладываются утилитой конфигурирования защиты iFIX, достаточно легко могут быть наложены и без ее использования путем ручного редактирования реестра Windows NT (Service Pack 3). Если этим способом запретить все, что возможно, то по <Ctrl+Alt+Del> диспетчер задач вызвать можно, однако в этом случае использовать его для запуска других приложений нельзя, можно лишь завершить работу уже запущенных приложений или же перезагрузить компьютер (операционную систему), что явно предполагает злой умысел. Представляется, что для обеспечения безопасности программным способом этого достаточно, так как человек со злым умыслом с тем же успехом может нажать кнопку аппаратного сброса или отключить электропитание, и меры защиты для этого случая должны быть скорее административными.

Выводы

В рамках обоих пакетов можно обеспечить требуемый уровень доступа для отдельного пользователя и группы пользователей.

В рамках применения обеих SCADA-систем можно обеспечить надлежащий уровень безопасности. Однако iFIX предоставляет разработчику более удобные и полные средства для конфигурирования защиты.

Локализация

Вопросы локализации были включены в программу тестирования не случайно. Англоязычный интерфейс и отсутствие документации на русском языке могут значительно затруднить процесс освоения системы. Причем здесь нужно говорить не только о трудностях, с которыми столкнется разработчик проекта, но также и о сложностях при

сопровождении системы. Именно поэтому наряду с локализацией справочной системы и документации нами проверялась локализация интерфейса оператора.

На сегодняшний день для пакета GENESIS32 русифицирован пользовательский интерфейс практически для всех основных компонентов:

- GraphWorX32 — среда разработки и просмотра мнемосхем;
- Symbol Library — библиотека символов;
- TrendWorX32 Viewer ActiveX — элемент просмотра графиков;
- AlarmWorX32 Viewer ActiveX — элемент просмотра событий и тревог;
- Persistent Trending — сервер фоновой буферизации;
- Security Configurator — конфигуратор подсистемы безопасности (администрирования);
- License Monitor (сервер регистрации событий и контроля лицензии).

Из важных компонентов не локализованным остался, пожалуй, только конфигуратор подсистемы обработки тревог (Alarm Server Configurator). К сожалению, встроенная справочная система GENESIS32 осталась нерусифицированной.

Локализованной версии пакета iFIX в настоящее время не существует.

Анализ имеющейся в нашем распоряжении документации показал, что из всего комплекта документов системы GENESIS32 на русский язык переведено около 50 процентов. Это, в первую очередь, руководство пользователя по GraphWorX32, которое также содержит разделы, посвященные работе с элементом просмотра графиков (TrendWorX32 ActiveX), сервером системного администрирования (Security Configurator), сервером фоновой буферизации данных (Persistent Trending), Visual Basic для приложений. Документация по VBA-классам представлена для следующих программных компонентов: GraphWorX32, TrendWorX32 Viewer ActiveX (элемент просмотра графиков), Persistent Background Trending (сервер фоновой буферизации). Также на русский язык переведен документ «Начальные сведения», который содержит информацию об архитектуре, порядке установки и настройки GENESIS32. Эти документы представляют собой файлы в pdf-формате, ориентированные на самостоятельное изготовление пользователем твердых копий. Файлы документации могут быть свободно получены пользователем с ftp-сервера дистрибьютора фирмы Iconics.

Документация на пакет iFIX переведена на русский язык полностью и поставляется только в бумажном виде, поэтому ее общедоступной электронной версии нет.

Что касается использования русского языка в экранных формах, то в обеих системах к этому нет никаких препятствий.

Текстовые объекты, надписи на кнопках и графиках, описания параметров можно вводить на русском языке, а использование скриптов позволяет определить свои собственные диалоги на русском языке (например диалог для дооблавления пера в график).

Единственная обнаруженная нами проблема с русским языком - это невозможность в iFIX использовать кириллицу в именах тегов при работе со встроенным в Workspace OPC-клиентом.

Документация

При анализе документации, кроме вопросов ее качества, особое внимание нами было уделено таким аспектам, как наличие учебника (методических указаний) по построению проекта, описания и способов устранения типовых ошибок, а также примеров проектов, демонстрирующих возможности пакета.

Электронная справочная система пакета GENESIS32 представляет собой совокупность файлов в форматах hlp и html. Ссылки, имеющиеся в этой справке, позволяют возвращаться к предыдущему разделу справки и переходить к ее содержанию. С помощью содержания можно перейти к любому разделу справки. К сожалению, практически отсутствуют перекрестные ссылки на связанные разделы документации.

Справочная система позволяет также искать информацию, используя возможность поиска полного текста или предметный указатель. Кроме того, из диалоговых окон с помощью кнопки Help может быть вызвана контекстно-зависимая справка, предоставляющая информацию об элементах диалогового окна и пошаговые инструкции по выполнению конкретных задач.

В состав электронной справочной системы входит руководство Training Manual, представляющее собой методические указания по построению проекта. Training Manual содержит большое число пошаговых инструкций, облегчающих процесс освоения основных приемов работы с пакетом.

Информационная система iFIX включает в себя электронные книги и оперативную справку. В состав электронных книг входит так называемая карта документации, определяющая основные разделы документации и их краткое содержание. Карта помогает быстро сори-

ентироваться в большом количестве справочной информации и найти интересующий раздел.

Электронные книги содержат концептуальную информацию об iFIX (архитектура и функции системы, конфигурирование iFIX, основные принципы работы, управление системой, организация защиты). Книги представляют собой хорошо структурированные электронные документы с большим количеством перекрестных ссылок на связанные разделы.

Кроме непосредственного просмотра текста и иллюстраций, электронные книги позволяют искать информацию, используя алфавитный указатель или возможность поиска полного текста. Наличие полного содержания комплекта документов в виде дерева разделов позволяет быстро переходить к нужному разделу книги. Кроме того, структура электронных книг делает возможным переход от концептуальной информации к соответствующей информации в оперативной справке. К сожалению, в электронных книгах практически полностью отсутствуют копии экранов.

Оперативная справка содержит подробную рабочую информацию, в частности, большое количество пошаговых инструкций для выполнения конкретных задач, описания элементов диалоговых окон. Эта система доступна из любого приложения iFIX и выдает контекстно-зависимую информацию типа «What's This?» («Что это?») и «How-To» («Как сделать?»).

Кроме того, в состав электронной информационной системы входит глоссарий с возможностью поиска интересующего термина.

Электронные книги содержат руководство «Введение в iFIX», дающее общее представление о системе iFIX, и руководство «Quick Start» («Быстрый старт»), которое помогает быстро начать работу с iFIX. В книгах можно найти также пример разработки базы данных процесса.

Документация обоих тестируемых продуктов включает описание и способы устранения типовых ошибок. В упомянутом документе GENESIS32 «Начальные сведения» есть раздел «Дополнительные сведения и сообщения об ошибках», который содержит описание и способы устранения типовых ошибок, возникающих при установке и настройке системы. В документации iFIX есть информация о типичных проблемах, с которыми можно столкнуться при работе с iFIX, и рекомендации по их решению, а также различные советы по оптимизации работы с iFIX.

И, наконец, несколько слов о примерах проектов. На компакт-диске, поставляемом фирмой Iconics, представлено большое количество примеров, демонстрирующих возможности пакета. Среди них можно найти довольно сложные примеры, приближенные к реальным условиям (например, с достаточно большим числом аналоговых параметров).

На компакт-диске, поставляемом фирмой Intellution, подобные примеры отсутствуют, но могут быть получены у дистрибьютора Intellution.

Говоря о документации в целом, можно отметить, что на компакт-диске, поставляемом фирмой Iconics, содержится достаточно полная информация о пакете GENESIS32. Однако плохая навигация по справочной системе значительно затрудняет работу с документацией и, как следствие, замедляет процесс освоения возможностей пакета.

Документация же пакета iFIX в этом плане более предпочтительна. Она содержит хорошо структурированную, полную и удобную в использовании информацию, позволяющую быстро и просто освоить основные возможности системы и получить ответы на интересующие пользователя вопросы.

Заключение

Тестирование само по себе — сродни ремонту: его можно прекратить, но нельзя закончить. Наверняка многие существенные обстоятельства остались за кадром, наверняка много вещей изменится с выходом в свет новых версий продуктов. Тем не менее, полученных сведений достаточно, чтобы констатировать, что на базе любого из рассмотренных продуктов можно сделать работоспособный проект для всякой типовой задачи АСУ ТП.

К сильным сторонам GENESIS32 можно отнести более удачную реализацию графической подсистемы и интерфейса OPC, компонентность, относительно низкую стоимость, возможность работы под управлением Windows 95/98, локализацию основных компонентов.

В то же время iFIX обладает большей устойчивостью к ошибкам в компонентах ActiveX, имеет более простую реализацию программного доступа к своим графическим объектам из VBA-скриптов, предоставляет больше возможностей для SQL-программирования и взаимодействия с базами данных, имеет более продвинутую систему управления доступом и протоколирования изменений, а сопроводительная документация на iFIX, на наш взгляд, является образцовой. ●