

ИНТЕГРАЦИЯ

MasterSCADA — инструмент крупноблочной сборки АСУ ТП

Илья Аблин

К счастью для всего прогрессивного человечества — или, по крайней мере, той его части, что занимается промышленной автоматизацией, — на SCADA-системы нет стандартов, ГОСТов и прочих решеток, ограничивающих свободу фантазии творцов. Это совершенно не значит, что все SCADA реализуют абсолютно разные функции, решают разные задачи и, не изучив тщательно все существующие пакеты, нельзя правильно выбрать нужную систему. Напротив: именно потому, что базовая функциональность операторской станции всеми разработчиками понимается примерно одинаково, можно не сомневаться, что любая современная SCADA так или иначе удовлетворит основные потребности заказчиков. Но раз так, то в чем же различие этих продуктов? Когда выбрать один, а когда совсем другой? Попробую ответить на эти вопросы. На мой взгляд, есть несколько существенных моментов, обуславливающих различия между SCADA-пакетами разных производителей.

Методология разработки АСУ ТП и архитектура пользовательского интерфейса

Как ни странно, важнее всего именно то, что не поддается никаким сравнительным оценкам. От методологии разработки зависит масштабируемость проекта — не только применимость одних и тех же решений по отношению к малым и большим системам, но, главное, и сама функция зависимости трудоемкости разработки от информационной мощности создаваемой системы. Плохо, когда вместо линейного роста мы получаем экспоненту. Не секрет, что многие SCADA-решения отлично подходят для создания небольших систем, но с их ростом теряются обзорность реализуемых проектов и возможность безболезненной коррекции продукта, а также управляемость процесса

разработки.

К методологии разработки относится и концепция построения пользовательского интерфейса. Упрощенно можно сказать, что в этой области возможно и количественное сравнение продуктов: сколько раз нужно щелкнуть мышью, чтобы реализовать ту или иную задачу? В скобках замечу, что это позволяет организовать спортивные соревнования разработчиков SCADA. Если же говорить серьезно, то развитая функциональность современных SCADA-пакетов, обширный и довольно эклектичный перечень модулей и опций разного назначения существенно усложняют построение такого пользовательского интерфейса, который сам будет помогать проектировщику вести разработку в методически верной последовательности, сокращающей трудоемкость реализации проектов и при этом позволяющей не задумываться над порядком действий.

Инструментарий тиражирования проектных решений

Большая часть системных интеграторов работает в одной отрасли, годами нарабатывает проектные решения и заинтересована в максимальном упрощении их переноса из одной системы в другую. Даже в рамках одного проекта нередко бывает необходимо растиражировать отдельные части в нескольких экземплярах. Параллельное использование в технологиях аппаратов и емкостей, фильтров и бункеров ставит перед разработчиками SCADA-системы задачу обеспечить эффективный механизм тиражирования полнофункциональных фрагментов проекта, относящихся к одному технологическому элементу, как в рамках одного проекта, так и с целью обмена между различными проектами. Увы, большинство SCADA сводит эту задачу к примитивному созданию библиотечных динамических элементов мнемосхем. При

этом за бортом остается вся логика контроля и управления объектом, архивирование и документирование работы с ним.

Открытость

Это понятие включает не только поддержку расширения возможностей продукта с помощью предоставляемых пользователю внутренних программных интерфейсов, но и использование для стыковки SCADA с внешним миром исключительно стандартных интерфейсов. Может показаться, что все SCADA-пакеты вполне удовлетворяют этому критерию. Однако многие конкретные задачи не всегда легко решаются на практике без привлечения к участию в проекте АСУ ТП профессиональных программистов или приобретения вспомогательных продуктов, обычно того же производителя.

Расширенная функциональность

Все современные SCADA-пакеты имеют полный набор необходимых базовых функций, но зачастую в жизни встречаются и специфические потребности. Это может быть связано с использованием SCADA в смежных областях, например, для задач диагностики технологического оборудования, создания тренажеров операторов или моделирования технологического процесса, но может быть обусловлено и тем, что сложная технология нуждается в адекватной реализации функций контроля и управления. Разумеется, можно сказать: если есть открытые интерфейсы, то есть и свобода творчества. Однако все же хотелось бы пореже изобретать велосипеды. В такой ситуации правильность выбора SCADA-пакета во многом определяется квалифицированной формулировкой специальных требований и трезвой оценкой соответствия им возможностей выбираемого продукта.

Сформулировав постановку задачи, попробую доказать, что в нашем продукте — пакете программ MasterSCADA (www.insat.ru/products/masterscada) производства компании “ИнСАТ”

(www.insat.ru) — практическая реализация удовлетворяет теоретическим воззрениям автора.

Ньютон как-то сказал, что он видел дальше других, потому что стоял на плечах гигантов. Создавая наш пакет, мы попробовали взобраться на плечи одному из таких гигантов — и выбрали в качестве подмостков объектно-ориентированное программирование (ООП) или, в нашем случае, скорее объектно-ориентированный подход. Ведь SCADA — это, в сущности, система визуального программирования, предназначенная для предметной области промышленной автоматизации. А ООП уже давно доказал свою эффективность для любых систем программирования.

Мы попытались реализовать этот подход, сочетая программистское понимание ООП и естественное для проектировщика понимание объекта как элемента проекта, адекватно воспроизводящего контролируемый технологический объект.

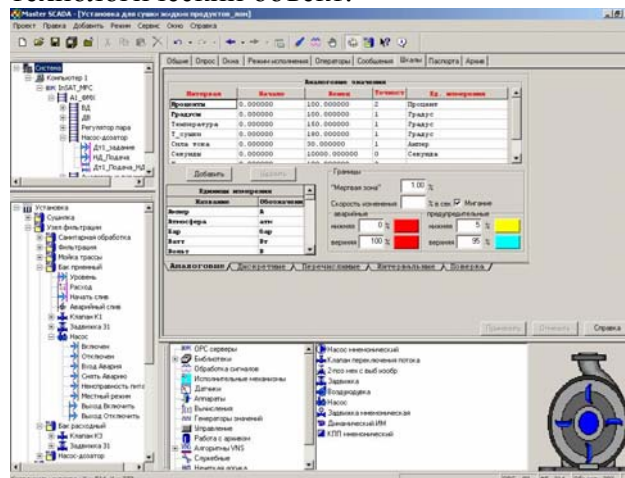


Рис. 1. Пользовательский интерфейс MasterSCADA

Посмотрим, что представляет собой проект в MasterSCADA (рис. 1). Мы создаем проект как отображение взаимосвязи модели технологического объекта, представленной в виде дерева технологической иерархии, и модели системы контроля и управления — в виде дерева иерархии технических средств системы. Приведу пример. На заводе есть цеха, цеха разбиваются на участки, на

участках имеются технологические аппараты, обвязанные исполнительными механизмами и датчиками. Это технологический объект. АСУ ТП объединяет операторские станции, к которым по каналам связи подключены контроллеры, состоящие из модулей ввода-вывода, содержащих входы и выходы. Это система. Свяжем датчик в дереве объекта с входом модуля контроллера в дереве системы и получим взаимосвязь этих моделей. Связь устанавливается простым перетаскиванием одного элемента на другой в любом направлении. Можно представить это и как взаимосвязь логического и физического уровней представления АСУ ТП. Разработчикам такая структура проекта кажется само собой разумеющейся, хотя она и не была реализована ни в одном из известных автору SCADA-пакетов.

Существенно и то, что проект разрабатывается не для отдельного рабочего места, а для всей АСУ ТП в целом, включая все операторские станции и все РС-совместимые контроллеры (функция загрузки управляющих программ в контроллеры появится в ближайшей версии — MasterSCADA v2.0). При этом никакой настройки сетевых связей как при разработке, так и при переносе системы на целевую платформу не требуется. Единственная настройка — это выбор для объекта его базового компьютера (или контроллера) в дереве системы. При старте системы все узлы находят друг друга по проектным именам, не имеющим никакого отношения к сетевым именам узлов.

Отказавшись от традиционного разделения SCADA-системы на редактор базы тегов (или каналов) и функциональные редакторы отдельных документов, мы получили массу преимуществ. Реализованная в пакете концепция “все в одном” обеспечила “бесшовное” объединение всех функциональных модулей в едином пользовательском интерфейсе, который пользователь с легкостью воспринимает

даже без тренировки или изучения документации, методом “тыка” (в том числе и буквально, поскольку все действия по созданию проекта основаны в MasterSCADA на технологии “перетаски и оставь”).

Выбор объекта в качестве центральной единицы разработки проекта позволил все переменные рассматривать как элементы его структуры, а документы — как его свойства. У нас нет нужды редактировать отдельно список тегов, отдельно — мнемосхемы, отдельно — тренды и каждый раз держать в памяти их взаимосвязи. Выбрав объект, мы можем быть уверены, что на закладках его свойств в пользовательском интерфейсе MasterSCADA есть все необходимое. Мало того, появление нового функционального модуля в составе пакета всего лишь добавляет объекту новую закладку (рис. 2). В результате мы можем манипулировать объектом как единым целым — тиражировать, помещать в библиотеку, переносить в другие проекты. При этом привязка к физическому миру в виде входов-выходов контроллеров производится в полуавтоматическом режиме с помощью специального мастера установления связей — в считанные секунды в момент тиражирования или вставки объекта в проект. Это означает, что все алгоритмы обработки данных, сообщения, мнемосхемы, тренды, рапорты и другие “свойства” библиотечного или тиражируемого объекта не требуют никакой дополнительной настройки после его добавления в дерево объектов.

Такой подход позволяет нам позиционировать MasterSCADA в качестве инструмента крупноблочной (следовательно, быстрой) сборки АСУ ТП. Разумеется, по мере работы с пакетом проектировщик накапливает собственные библиотеки и все реже создает новые объекты, а просто выбирает уже имеющиеся. Это существенное преимущество MasterSCADA дает нам возможность утверждать, что

трудоемкость разработки АСУ ТП в нашем продукте на порядок меньше, чем в других SCADA.

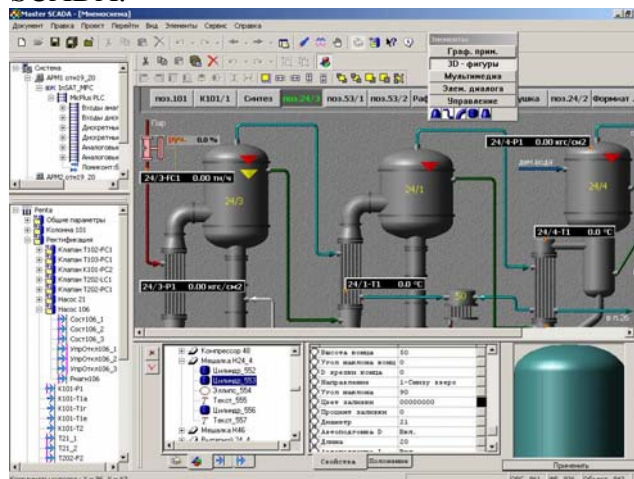


Рис. 2. Графическая подсистема.

Классическая для программистского понимания ООП триада *наследование* — *инкапсуляция* — *полиморфизм* является основой достижения описанного выше результата. Она реализуется в таких особенностях MasterSCADA, как наследование свойств в объектной иерархии проекта, возможность запрета для облегчения тиражирования использовать в документах объекта внешние по отношению к нему переменные, а также типизация (на уровне пакета в целом или для конкретного проекта) документов различных объектов, входящих в проект, и действий с ними.

Открытость в рамках системы — это не только возможность создавать на основе имеющихся программных интерфейсов новые модули и типовые объекты. В значительной степени это принципиальное решение, сводящееся к использованию для взаимодействия с внешним миром исключительно стандартных интерфейсов.

Если для многих SCADA необходимость поддержки стандарта OPC для связи с устройствами ввода-вывода представляет собой искусственную, снижающую быстродействие надстройку над созданной еще до появления этого стандарта фирменной драйверной подсистемой, то MasterSCADA изначально реализована по наиболее современной

технологии “OPC в ядре системы”. Причем не только в отношении поддерживаемого всеми стандарта на передачу данных (OPC DA). В ближайшей версии планируется поддержка использования уже существующих внутри системы клиентских интерфейсов для доступа к архивам значений технологических переменных по стандарту OPC HDA и к сообщениям по стандарту OPC A&E.

В MasterSCADA также реализован нетипичный для SCADA, но весьма гибкий подход, заключающийся в стыковке с системами верхнего уровня. Связь с базами данных АСУП реализуется не путем SQL-запросов, а вызовом хранимых процедур. С одной стороны, это позволяет разделить зоны ответственности разработчиков: процедуру реализует разработчик БД, а ее вызов производит разработчик АСУ ТП; с другой стороны — освобождает разработчика АСУ ТП от необходимости изучать язык запросов SQL и структуру имеющейся БД АСУП. Есть и чисто технические преимущества: повышение быстродействия в сравнении с прямым использованием запросов, упрощение реализации записи в БД.

Процедуры представлены в дереве системы проекта аналогично другим источникам ввода-вывода данных, и установление связи с их параметрами или выборками ничем не отличается от привязки к сигналам контроллера. Так, например, в одной из внедренных нами систем единственной требуемой функцией была запись значений нескольких сотен параметров из контроллера непосредственно в MS SQL Server. Как вы уже догадались, на ее реализацию потребовалась всего пара часов.

Еще одна возможность повысить открытость MasterSCADA — широкое использование XML как на этапе разработки (для хранения проекта), так и на этапе функционирования исполнительной системы для сохранения журналов сообщений и т. п.

Упомянутый выше критерий

расширенной функциональности не может иметь сколько-нибудь формализованного толкования. Скорее это вопрос о том, какие “вкусы” имеет продукт и нужны ли они конкретному разработчику в конкретной системе. Так что я здесь всего лишь коротко перечислю, чем мы можем похвастаться.

Прежде всего — библиотеки готовых технологических объектов. Возможность перетащить на мнемосхему не требующий настройки готовый объект, уже имеющий динамизированное изображение, сообщения, логику работы, окно управления и т. д. и есть один из главных резервов роста производительности труда разработчиков АСУ ТП. Например, библиотека изначально содержит подпрограммы индикации уровня, температуры продукта, может показать вращающуюся мешалку, выдает сообщения о превышениях и т. п. Любой объект, имеющий мнемосхему, может быть перетащен в другую мнемосхему, в которой будет находиться сжатое изображение его собственной мнемосхемы с функцией вызова любых принадлежащих ему документов.

Представление любой перетащенной из проекта на мнемосхему переменной в виде щитового прибора или сигнализатора, а групп переменных — в виде таблиц или графиков позволяет в считанные секунды создать полноценный пользовательский интерфейс. Применение стандартных и выбираемых пользователем шкал переменных, определяющих контролируемые границы и скорость изменения, единицу измерения и точность, обеспечивает унификацию отображения переменных во всех частях пользовательского интерфейса Master SCADA в режиме исполнения. На пользователей немало впечатление производят и собственные возможности графической подсистемы — объемная графика, встроенный редактор анимации со спецэффектами (движение, вращение, фокусировка и т. п.). В ближайшей версии

планируется реализовать шаблоны мнемосхем и слои изображения. И хотя мы можем без программирования и сценариев инициировать любое свойство любого ActiveX-элемента, с точки зрения идеологии пакета это путь к разработке не мнемосхем, а изображений библиотечных объектов.

Весьма развита в MasterSCADA обработка данных (подсистема Softlogic), позволяющая создавать графические схемы потоков обработки данных (рис. 3) на языке функциональных блоков. Имеется свыше сотни библиотечных блоков, включая ряд сложных алгоритмов управления — например, ПИД-регулятор с функцией оптимальной настройки и управление периодическими процессами по циклограмме. В проектах широко применяются свободные формульные вычисления с использованием разнообразных функций, в том числе такрой, как работа с архивом. Среди различных типов данных имеются такие удобные с точки зрения специфики промышленной автоматизации типы, как время, перечислимые и интервальные переменные.

Создание отчетных документов облегчается готовым шаблоном стандартного сменного или суточного рапорта со встроенными функциями вычисления почасовых и иных средних и интегральных значений для аналоговых переменных, а также счетчиков включений и моторесурса — для дискретных.

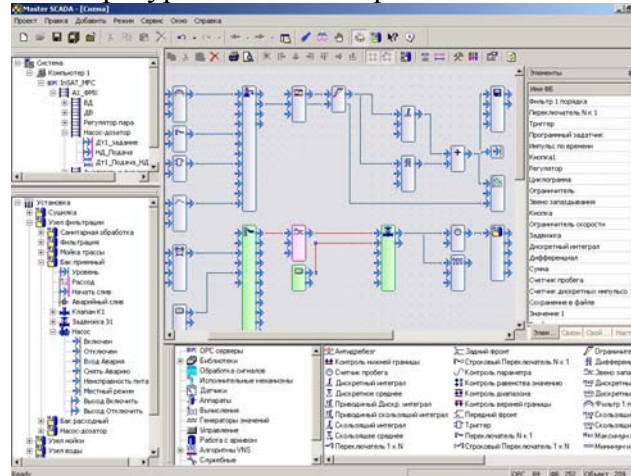


Рис. 3. Редактор схем ФБ

Любые действия с любыми документами и переменными объектов в системе могут быть выполнены по расписанию или событию.

В MasterSCADA имеются также не встречающиеся ни в одном другом SCADA-пакете опции метрологической поверки и паспортизации технологического оборудования.

Пожалуй, остановлюсь. Не потому, что нечего более сказать, а потому, что

пределы статьи ограничены. К счастью, бесплатная инструментальная система доступна любому желающему проверить достоверность вышесказанного (www.insat.ru/products/masterscada).

С автором статьи, генеральным директором компании “ИнСАТ”, можно связаться по e-mail: ablin@insat.ru