

SCADA НОВОГО ПОКОЛЕНИЯ. ЭВОЛЮЦИЯ ТЕХНОЛОГИЙ – РЕВОЛЮЦИЯ СИСТЕМОСТРОЕНИЯ

И.Г. ВАРЛАМОВ (Компания ИнСАТ)



В статье рассматривается вклад современных технологий в направление развития нового поколения SCADA-систем и пример применения этих технологий в MasterSCADA 4D.

Ключевые слова: стандарт HTML5, протокол OPC UA, АСУ ТП, системы программирования SoftLogic, SCADA системы, MasterSCADA 4D, стандарт МЭК 61131-3.

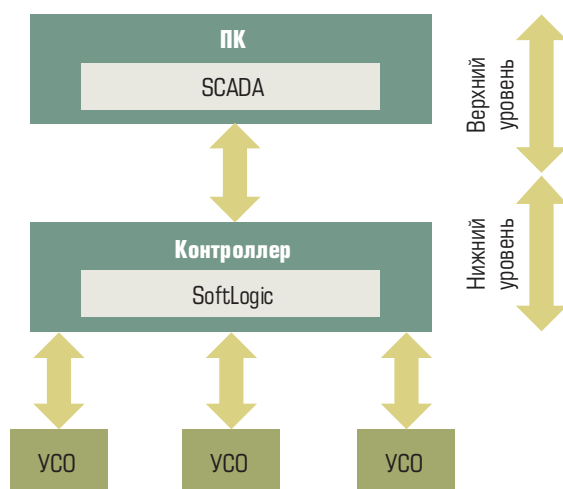
НА РУБЕЖЕ ЭПОХ

Мы сегодня живём на рубеже смены эпох в области IT технологий. Эпоха персональных компьютеров, безраздельно главенствующая на протяжении последних трёх десятилетий, неминуемо сдаёт позиции. Вместе с этим, мы вступаем в эпоху “после ПК”. Мобильные решения и облачные технологии – далеко не полный список нового поколения информационных технологий. “Интернет вещей”, веб доступ к самым разным системам от бытовых приборов до сервисов государственных услуг – неотъемлемые атрибуты современной повседневной жизни.

Что принесла новая эпоха и новые IT технологии в системы автоматизации? Системы визуализации, основанные на стандарте HTML5, позволяют отображать информацию

не только на привычных мониторах, но и на планшетах, мобильных телефонах и других устройствах. Всё большее распространение находит новый протокол OPC UA, обеспечивающий транспорт данных в гетерогенных IP сетях. Исполнительные системы становятся кроссплатформенными, что обеспечивает горизонтальную переносимость прикладных программных решений.

На первый взгляд, всё перечисленное лишь расширяет прежние возможности. Благодаря HTML5 увеличился спектр устройств, которые можно использовать для визуализации. OPC UA – ещё один новый протокол. Кроссплатформенность облегчает перенос программных решений с одного устройства на другое. В чём же здесь революционность? Что прорывного в этих технологиях? Сами по себе новые технологии ещё не являются революцией, но они служат основой для систем нового поколения. Совокупность этих технологий даёт возможность построения систем, обладающих не только горизонтальной, но и свободной вертикальной переносимостью программных компонентов проекта, чего не было ранее.



▲ Рис. 1. Пример структуры “классической” системы контроля и управления

“КЛАССИЧЕСКАЯ” СТРУКТУРА АСУ ТП

Рассмотрим пример структуры “классической” системы контроля и управления, какой мы её привыкли видеть последние десятилетия (рис. 1). Полевое оборудование связано с модулями УСО. Те, в свою очередь, связаны с контроллером, который обеспечивает сбор, первичную обработку данных, реализацию логики управления и выработку управляющих сигналов. Контроллер представляет собой ниж-

ний уровень АСУ ТП. Далее, контроллер связан с персональным компьютером, который производит окончательную обработку данных и обеспечивает взаимодействие системы управления с человеком, что представляет собой верхний уровень АСУ ТП. Для нижнего уровня – контроллеров – используются системы программирования SoftLogic. Для верхнего уровня – персонального компьютера – применяются SCADA системы. И эти понятия стали практически синонимами. Говорим “нижний уровень” – подразумеваем контроллеры и SoftLogic. Говорим “верхний уровень” – имеем в виду персональный компьютер и SCADA.

Более сложные распределённые системы построены по принципу клиент-серверной архитектуры, где несколько контроллеров и несколько клиентских АРМ соединены с сервером. Тем не менее, это та же самая двухуровневая система.

Существующие вертикально-интегрированные системы программирования, позволяющие в одной среде разрабатывать прикладные программы как верхнего, так и нижнего уровня не отменяют разделения на верхний и нижний уровень.

СКВОЗЬ ЛИНИЮ РАЗГРАНИЧЕНИЙ

Современные технологии позволили совершить прорыв сквозь ещё недавно кажущуюся незыблемой жесткую линию разграничения между верхним и нижним уровнем. Кроссплатформенность всех составляющих компонентов системы позволяет “отвязать” программную реализацию различных её частей от уровня в целом и от конкретных устройств в частности. Другими словами, программная часть становится переносимой не только по горизонтали (между устройствами одного уровня, работающими на различных платформах), но и по вертикали – между уровнями.

Рассмотрим различные возможные варианты расположения программных компонентов системы. Систему сбора и обработки данных назовём сервером данных. Он включает в себя протоколы связи, средства хранения и хорошо знакомый нам SoftLogic. Система визуализации, обеспечивающая отображение и взаимодействие с человеком, состоит из веб-сервера и веб-клиента (веб-браузера). А линию разграничения верхнего и нижнего уровня оставим применительно к аппаратной части.

Начнём с варианта, когда все программные компоненты размещаются в контроллере

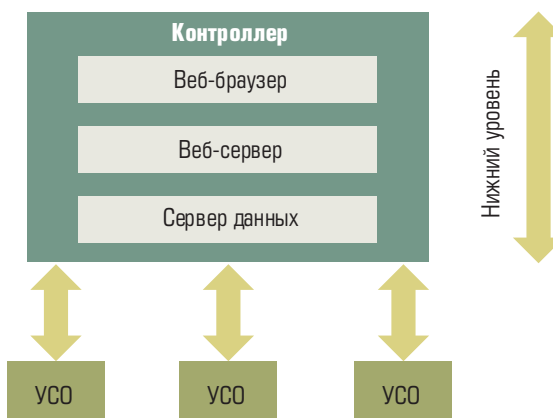


Рис. 2. Вариант размещения всех программных компонентов в контроллере

(рис. 2). Такое решение может быть, например, при использовании контроллера с собственным встроенным или внешним дисплеем. При таком решении все программные компоненты, включая систему визуализации, расположены в контроллере – под линией, прежде разграничивающей верхний и нижний уровни. Другими словами, компоненты, являющиеся ранее непременной составляющей верхнего уровня, теперь находятся на тех устройствах, которые ещё недавно считались принадлежностью сугубо нижнего уровня.

Если система визуализации построена на основе веб-сервера, то достаточно подключения контроллера к сети, чтобы часть системы отображения, представляющая собой браузер, могла свободно переместиться за пределы нашего “нижнего” уровня и перейти на уровень выше (рис. 3). В самом простом применении

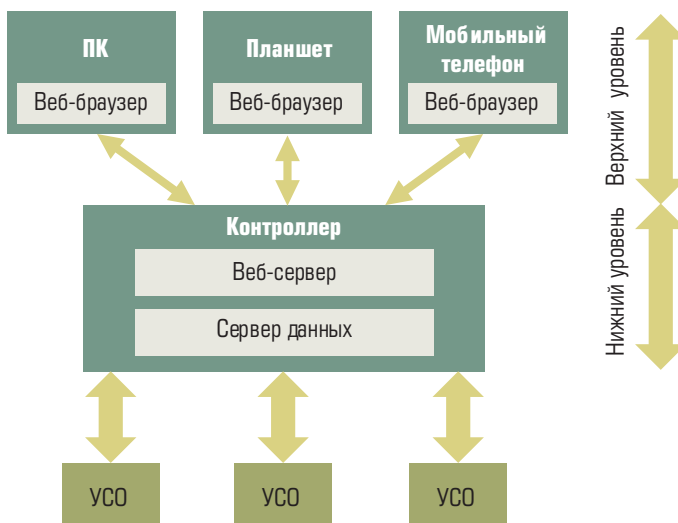


Рис. 3

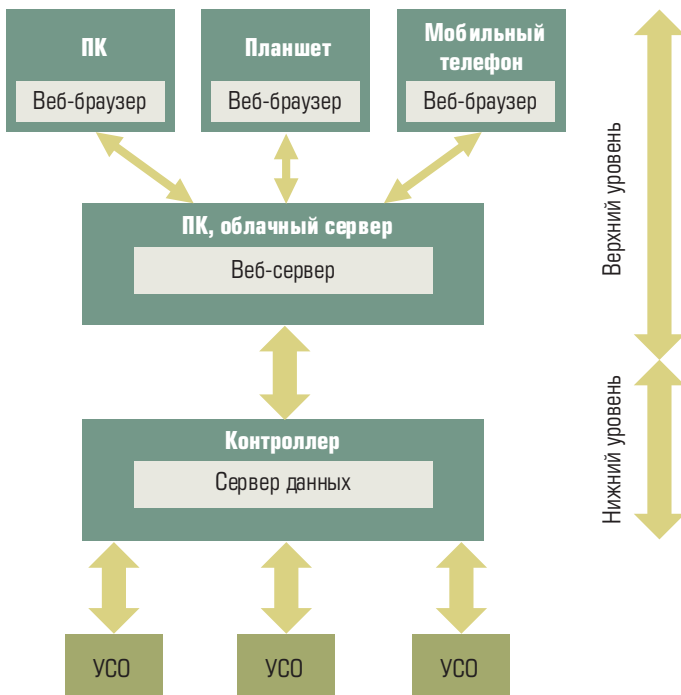


Рис. 4

это может быть тот же самый персональный компьютер, о котором мы уже упоминали, когда говорили о “классической” структуре системы управления. В таком случае на контроллере располагается сервер данных и веб-сервер, а на

компьютере – веб-браузер. Особенность такого решения заключается в том, что верхней частью системы отображения не обязательно должен являться конкретный ПК, а любое устройство, на котором можно запустить веб-браузер, имеющее доступ по сети к нашему контроллеру.

Следующий вариант конфигурации системы – сервер данных остаётся в контроллере, а система отображения, включая веб-сервер, переезжает на уровень выше. Здесь следует остановиться подробнее – что собой сегодня представляет этот самый уровень. Его роль может выполнять уже хорошо нам знакомый персональный компьютер (рис. 4). Но лишь этим дело не ограничивается. Как таковой “верхний” уровень не привязан к конкретному железу и может располагаться, в том числе, и на облачном сервере. Это позволяет построить систему визуализации не только для нескольких локальных контроллеров, но и для контроллеров, расположенных на значительном удалении друг от друга.

Завершающий этап “миграции” системы управления наверх – перемещение туда же и сервера данных (рис. 5). В этом случае вся логика управления располагается на верхнем уровне, а необходимость в контроллере, в его привычном представлении – утрачивается. Для связи с модулями УСО достаточно применение недорогого коммуникационного устройства.

Разумеется, в рамках одного проекта никто не будет хаотично перемещать различные компоненты системы, и такая гибкость может показаться избыточной. Для чего же нужны все эти возможности? Во-первых, достаточно одной программной среды для создания систем управления самой разной конфигурации, топологии и сложности. Во-вторых, в системах управления, как правило, решается большое количество типовых задач. И самое ценное в работе инженера – это его наработки и готовые решения. Среда программирования, обладающая вышеперечисленными свойствами переносимости позволяет использовать одни и те же типовые наработки независимо от места их применения и конфигурации конкретной системы, что сокращает время на разработку и удешевляет её стоимость.

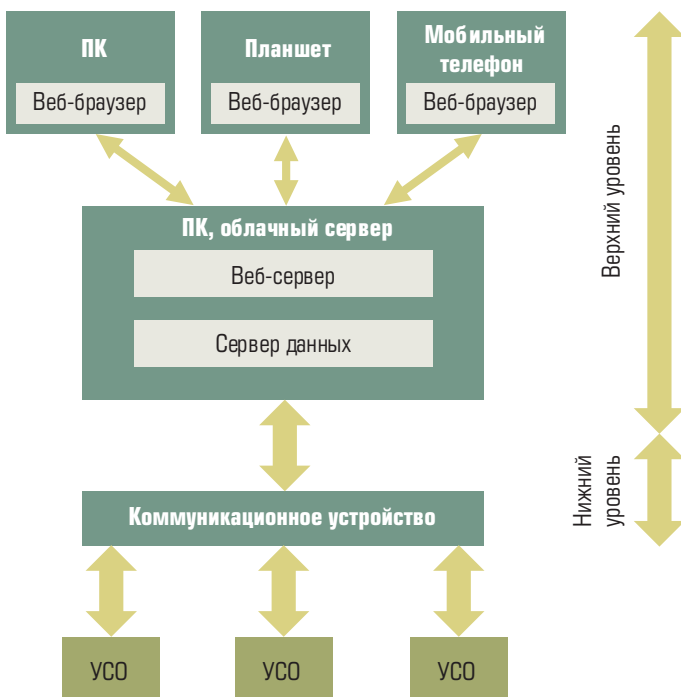


Рис. 5

MasterSCADA 4D – СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В ДЕЙСТВИИ

Примером SCADA-системы нового поколения является MasterSCADA 4D, разрабатываемая компанией ИнСАТ.

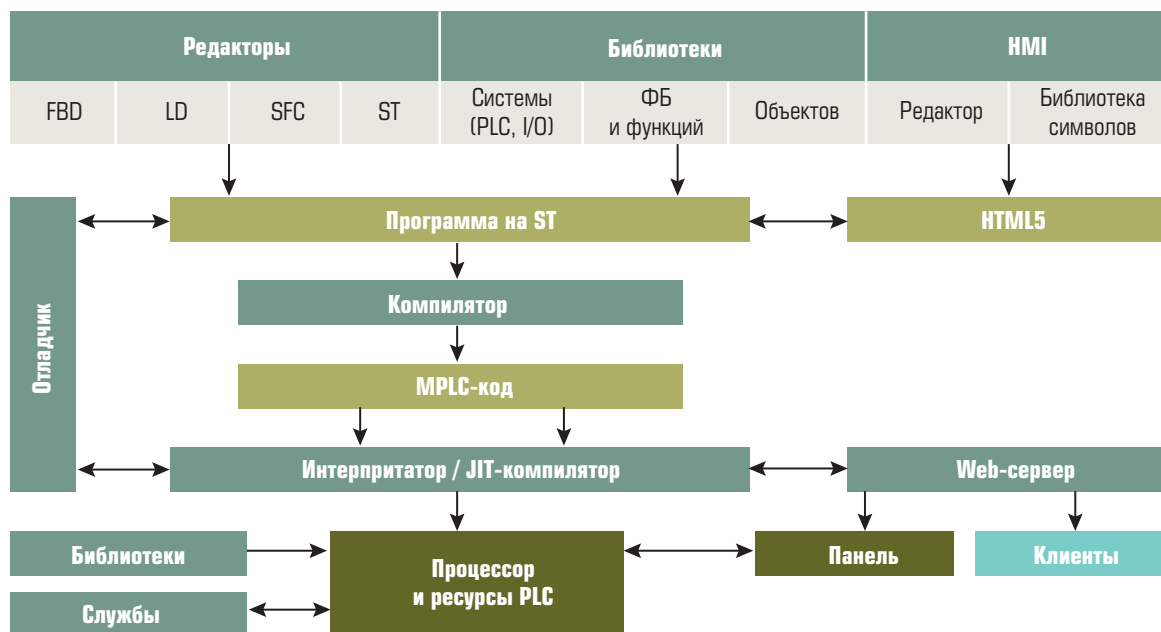


Рис. 6

MasterSCADA 4D состоит из среды разработки и исполнительной системы. Среда разработки, как и прежде, является Windows-приложением и предназначена для работы на персональном компьютере. Исполнительная система – кроссплатформенное приложение, предназначенное для исполнения созданных в среде разработки прикладных программ, а также обеспечения человеко-машинного интерфейса. Таким образом, код прикладных программ независимо от места его последующего исполнения остаётся всегда один и тот же, а исполнительная система для каждой программно-аппаратной платформы различная. На сегодняшний день она поддерживает операционные системы Windows, Linux, Android и Эльбрус.

СТРУКТУРА MasterSCADA 4D

Внутренняя структура MasterSCADA 4D изображена на рис. 6.

Прикладные программы разрабатываются на хорошо знакомых по системам SoftLogic языках стандарта МЭК 61131-3, каковыми являются ST (язык структурного текста), FBD (язык функциональных блоков), SFC (язык функциональных последовательностей) и LD (язык релейной логики). Программы, созданные на графических языках (FBD, SFC и LD) предварительно компилируются в код ST. Затем

вместе с кодом, изначально написанном на ST, компилируется MPLC-код, который в зависимости от конкретной системы, либо интерпретируется непосредственно исполнительной системой, либо компилируется с помощью JIT-компилятора.

Система визуализации разрабатывается в этой же среде с помощью редактора мнемосхем, поддерживающего векторную SVG графику с возможностью пользовательской параметризации различных частей изображения (т.е. пользователь может самостоятельно сделать динамически управляемыми различные части изображения). Исполнительная часть системы визуализации представляет собой кроссплатформенный веб-сервер, который может физически располагаться как вместе с исполнительной системой сервера данных, так и отдельно, на другом устройстве или в облаке. Обмен данными между сервером визуализации и сервером данных осуществляется посредством веб сервисов.

СРЕДА РАЗРАБОТКИ

В среде разработки реализован подход, зарекомендовавший себя среди широкого круга пользователей MasterSCADA предыдущих поколений, и заключающийся в раздельном построении структуры физической реализации системы (дерево системы) и системы

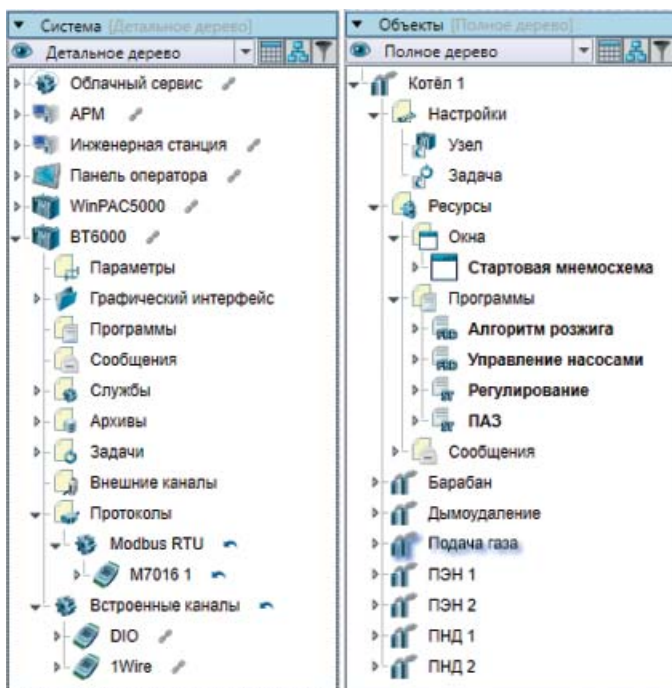


Рис. 7

абстрактной модели объекта (дерево объектов) – рис. 7.

В структуре дерева Системы задаются узлы системы, место их физического расположения, протоколы связи. В качестве узла может служить как физическое (контроллер, персональный компьютер), так и виртуальное устройство (в том числе – облачный сервис). Межузловое взаимодействие реализовано на основе протокола OPC UA, что обеспечивает “бесшовное” (не заметное для пользователя) соединение любых компонентов в рамках одного проекта – различные компоненты прикладной программы могут располагаться на разных узлах системы, при этом пользователю не требуется специально создавать линии связи между узлами или назначать протоколы обмена данными между ними.

В структуре дерева Объектов строится иерархия прикладных программ, данных и средств визуализации.

Такой подход позволяет разрабатывать логику управления без привязки к конкретному “железу”, а уже потом привязать параметры структуры объекта к конкретным входным и выходным каналам, заданным в структуре системы. При изменении конфигурации си-

стемы потребуется переделать только дерево системы. А всё дерево объектов, которое включает прикладные программы, данные и визуализацию – останется нетронутым. За счёт этого достигается гибкость применения одних и тех же решений в системах различной конфигурации.

Ещё одну степень свободы даёт возможность любой программный компонент (как объект, так и просто отдельный функциональный блок) назначить для исполнения в определённом узле системы. Таким образом можно перераспределять нагрузку в зависимости от вычислительной мощности того или иного оборудования или от специфики выполнения той или иной задачи. Например, логику аварийных блокировок, ввиду высоких требований по быстродействию, назначить для выполнения в контроллере, а ведение долгосрочных архивов и их визуализацию – делегировать облачному сервису.

ВЗГЛЯД В БУДУЩЕЕ

Новые технологии принесли в область системостроения горизонтальную и вертикальную переносимость программных компонентов, бесшовные соединения между узлами, гибкое перераспределение ресурсов. Всё это позволяет создавать системы отвечающие современным запросам потребителей, а труд инженера делает более плодотворным и эффективным.

Подробнее о новом продукте MasterSCADA 4D и о технологиях, лежащих в его основе, а также о перспективах развития можно будет узнать на специальной конференции, организуемой компанией ИнСАТ, которая состоится 16 марта 2013 года в ИПУ РАН. Приглашаем всех желающих. Будем рады встрече с Вами.

Список литературы

1. *Аблин И.Е.* “Заметка о вкусном и здоровом контроллере”. “ИСУП”, № 3(51), 2014.
2. *Аблин И.Е.* “MasterPLC. Контроллер как открытая платформа”. “Автоматизация производства”, № 2, 2011.
3. *Богданов Н., Киселёва О.* “OPC UA: изменения в популярной технологии информационных обменов с точки зрения инженера”. “СТА”, № 3, 2010.

Варламов Илья Геннадьевич – ведущий специалист компании ИнСАТ.