

**ERSY CONTROL: ИМПОРТОЗАМЕЩАЮЩАЯ
ПРОГРАММНАЯ ПЛАТФОРМА АСУТП.
АВТОМАТИЗАЦИЯ КРУПНЫХ ОБЪЕКТОВ АПК**

И.Е. Ермаков

ООО «Метасистемы», ООО «Базис Инжиниринг Групп»

e-mail: ie@iermakov.ru

Н. В. Плынский

ООО «Базис Инжиниринг Групп»

В статье разобраны сильные и слабые стороны доминирующего подхода к построению систем АСУТП, в контексте других видов систем управления. Представлена программная платформа ERSY Control, предназначенная для решений промышленной автоматизации (АСУТП/ICS). Отражён опыт использования Оберон-систем как основы для создания подобных прикладных платформ. Платформа включает средства разработки и исполнения для подсистемы ввода-вывода (сопряжения с ПЛК различных производителей), (под)системы управления (СУ/CS), подсистемы человеко-машинного интерфейса (ЧМИ/НМИ). В процессе разработки — подсистема сбора и архивации данных (БДРВ/RTDB). Платформа применяется для автоматизации крупных объектов агропромышленного комплекса.

Ключевые слова: промышленная автоматизация, АСУТП, SCADA, платформа, импортозамещение, автоматизация в сельском хозяйстве, элеватор. зерносушильный комплекс, Оберон, Компонентный Паскаль, Блэкбокс.

1. Системы управления и промышленная автоматизация: подходы и проблемы

Системы управления реального времени — это обширный класс задач, со своими особенностями и особыми требованиями к квалификации разработчиков и к качеству систем. На наших глазах управляющие системы из области специализированных (бортовые, промышленные) или обособленных малозначимых (простая бытовая техника предыдущих поколений) становятся действительно вездесущими и критично значимыми (массовый транспорт, коммунальные системы, здания, медицина, так называемый IoT — Internet Of Things). Любой сбой таких систем становится всё более болезненным или смертельным для окружающих, а любая уязвимость незамедлительно используется преступным миром или противником в гибридном противостоянии. В связи с очевидным началом новой Холодной войны встаёт во весь рост так называемая проблема цифрового суверенитета [1, 11].

Проблема надёжности требует совершенствования методов и инструментов разработки, о чём мы поведём речь далее. Проблема цифрового суверенитета требует решения задач импортозамещения — как в сфере оборудования, так и в сфере ПО. Если говорить об отрасли АСУТП, то на рынке присутствует не один десяток ПЛК отечественного производства, однако требуется их перевод на российскую элементную базу (которая имеется). В сфере программного обеспечения — отечественных SCADA-систем — имеется

большой выбор, но их «ахиллесовой пятой» является традиционная для АСУТП привязка к ОС Microsoft Windows.

Мир систем управления можно условно сегментировать и по типу решаемых задач, и по способу их решения. При этом интересующая нас отрасль промышленной автоматизации обладает как выраженной спецификой задач, так и устоявшимся, особым по отношению к остальной программной инженерии, способом их решения. Но сначала коснёмся других сегментов.

Сегмент ответственных встроенных и бортовых систем реального времени включает в себя широкий спектр задач (бортовое авиационное, космическое и корабельное ПО, диспетчерские системы воздушного транспорта, ПО медицинского оборудования и т. п.). Особенности: разработка оборудования и программного обеспечения под конкретное изделие; программирование с помощью универсальных языков программирования, со всеми их архитектурными подходами; управляющее ПО включает большое количество параллельных алгоритмов, богатый набор структур данных, сложные математические вычисления [2]. Языки программирования 3GL могут дополняться языками повышенного уровня — 4GL, DSL, CASE [3, 4]. Т.е. можно говорить, что этот класс задач относится к миру программной инженерии, использует её подходы и инструменты. Также в этом сегменте имеется традиция (к сожалению, не доминирующая) применения высоконадёжных, качественных языков программирования, таких как Modula-2 и Ada [5, 6], в последнее время — Oberon-07 (например, на БПЛА [7]).

Второй сегмент — бытовые встроенные системы, программирование микроконтроллеров и т. п. Эти задачи также решаются программированием. Однако ограничения дешёвых микроконтроллеров и традиции «творческого лихачества» обусловили доминирование плохих практик, характерных для раннего этапа развития ИТ (чрезмерная экономия ресурсов ценой отказа от хорошего структурирования данных и алгоритмов; применение языка C, не обеспечивающего контроль памяти и строгую типизацию; отсутствие модульности и абстракции). Увы, плохие практики этого сегмента всегда просачивались и в сферу более ответственных систем (транспорт [8], медицина). Тенденции вроде Internet Of Things «развиваются» в худших традициях этой сферы. Новой формой безответственного безумия стало проникновение в сферу встроенного ПО скриптовых языков из сферы Web-программирования (Python, JavaScript).

Перейдём, наконец, к предмету данной статьи — сфере промышленной автоматизации, АСУТП. Отдельные задачи и решения этого сегмента могут походить на первые два, но в основном, бесспорно, мир АСУТП отличается от мира программной инженерии. Решения промышленной автоматизации предполагают сборку систем из конфигурируемых аппаратных модулей (ПЛК; модули ввода-вывода) и программных модулей (группы правил на языках МЭК; OPC-серверы; сборка и конфигурирование ЧМИ в SCADA-системах). Основная часть логики управления укладывается в простой рабочий цикл «опрос входов — применение правил — вычисление выходов», асинхронно выполняемых алгоритмов управления мало или нет вообще. Исторически такая управляющая

логика исполнялась схемно, релейным способом (или на полупроводниковых логических схемах). Переход к программируемым контроллерам связан не только с ростом сложности процессов, а и с экономическими соображениями — компактностью, удешевлением, упрощением разработки, гибкостью перенастройки. Однако подход, мышление во многом перенеслись.

Возможность модульной сборки «из кубиков», конечно, упрощает процесс создания систем, повышает надёжность — и может быть засчитана «в плюс» миру АСУТП. Однако рассмотрим и вытекающие из этого проблемы:

1) Отсутствие средств формализации, абстракции-обобщения и повторного использования, которые являются центральными для программной инженерии (в частности, ООП). В традиционных инструментах АСУТП, по факту, нет нормальной возможности определить некоторую логику управления в обобщённом параметризуемом виде и использовать её многократно для каждого повторяющегося (или отличающегося параметрически) узла системы. Если объект включает, например, 60 транспортёров, то нередки случаи, когда инженер просто введёт 60 копий правил для управления ими.

Конечно, всё не так удручающе. Например, достойная российская система MasterSCADA позиционируется как объектно-ориентированная — и подразумевает унификацию и параметризацию компонентов.

Однако по мере роста сложности и требований к гибкости управления неизбежно проявляет себя закономерность п. 2.

2) За определённым порогом сложности задачи проще её запрограммировать на языке программирования, чем продолжать использовать конфигурирование и сборку на пользовательском уровне. Эта закономерность проявляет себя тем, что во всех продуктах рано или поздно появляются скриптовые языки. Однако качество эти языков удручающее (чаще всего нечто С-образное, нередко — Visual Basic for Application), что связано с тем, что разработчики SCADA-систем не являются специалистами по проектированию языков программирования. Отрицательным примером здесь может являться, например, Siemens Simatic WinCC, с его эклектикой средств, многогигабайтным размером и «дырявым» С-образным скриптовым языком.

Таким образом, наряду с перспективностью высокоуровневого, сборочного подхода мы имеем и букет непреодоленных проблем, касающихся как надёжности и безопасности, так и возможности гибкой, расширяемой разработки, интенсивного повторного использования однократно разработанной логики — как в рамках одного объекта, так и между объектами. Нам представляется, что корень этих проблем можно выразить так. Инструменты промышленной автоматизации пытаются до последнего не дать инженеру запрограммировать, и даже, когда он уже «почти программирует», — обеспечивать психотерапию: «Не волнуйтесь, это не настоящее программирование, а всего лишь написание небольшого скрипта». Хотя гораздо перспективнее для современного инженера АСУТП освоить основные принципы надёжного программирования и разработки программных систем.

На наш взгляд, пришла пора объединить лучшее из сферы АСУТП и из сферы сложных систем управления. А именно: взяв за основу некоторую

современную систему программирования и времени выполнения (с поддержкой компонентной модели и метапрограммирования в рантайме), разработать прикладной объектно-ориентированный (ОО) каркас (фрэймворк) для АСУТП, предоставить разработчику языковые и визуальные инструменты сборки системы управления из сущностей данного каркаса.

Подобные решения в отрасли вызревают — и в качестве основы часто берётся JavaVM или .NET Framework. Однако эти платформы имеют очевидные проблемы, ограничивающие их применение в ответственных системах реального времени. Альтернативным решением является использование какой-либо из Оберон-систем — как обладающей всеми свойствами JVM или .NET (и являющейся их прародителем, не будем забывать), но при этом на порядок более компактной и детерминированной. В этом качестве нами выбрана система Компонентный Паскаль/Блэкбокс (КП/ББ).

2. Платформа ERSY Control — структура и возможности

Платформа ERSY Control является прикладным решением на базе системы программирования и исполнения КП/ББ, которое реализовано как ОО-фреймворк и набор визуальных инструментов для программиста, встраиваемых в графическую среду КП/ББ [9].

В структуре платформы (естественным для отрасли АСУТП образом) можно выделить три горизонтальных слоя:

- ERSY Control IO — подсистема ввода-вывода (сопряжение с различными ПЛК; генерация простой защитной логики, исполняемой на ПЛК; пространство тегов, независимое от используемых ПЛК);

- ERSY Control Server — подсистема управления (СУ/CS), исполняемая на встраиваемом ПК под управлением ОС Unix-семейства. Разработка управляющей логики выполняется сборочно-конфигурационным программированием на основе абстракций ОО-фреймворка;

- ERSY Control HMI — подсистема человеко-машинного интерфейса (ЧМИ/HMI). Инструмент позволяет удобно разрабатывать графические элементы для визуализации, с использованием векторной независимой от разрешения устройств графики, а затем собирать мнемосхемы и формы управления, связывая их с обработчиками и событиями управляющего сервера.

Также можно выделить вертикальный компонент, связанный со всеми горизонтальными слоями, — подсистему сбора, архивации и анализа данных (БДРВ/RTDB). В настоящее время эта подсистема на стадии разработки.

Платформу ERSY Control можно отнести к классу распределённых КАСДУ/SCADA-систем, имеющих средства разработки управляющей логики (так называемые интегрированные SoftLogic SCADA). При этом мы избегаем термина SCADA, поскольку его содержание расплывчато: в нём объединены достаточно разные задачи сбора/архивирования данных и организации интерфейса с оператором, при этом неоднозначен вопрос относительно включения или не включения в это понятие управляющей логики.

В 2016-2017 годах на основе платформы нами внедрены АСУТП крупных объектов агропромышленного комплекса [10]. При этом удалось агрегировать разнородное унаследованное оборудование (разнотипные ПЛК и шкафы

управления), оперативно разворачивать и модернизировать систему в буквальном смысле слова «в поле», обеспечивать гибкую логику управления техпроцессами. Освоенный класс задач — управление потоковыми сыпучими производствами. Удалось абстрагировать данные задачи и получить единую библиотеку компонентов BigAgroLib.

3. Выводы

ERSY Control является перспективным решением, обеспечивающим надёжность, гибкость, переносимость, накопление повторно используемых обобщённых компонентов для задач АСУТП, что даёт возможность развития в сторону повышения интеллектуальности алгоритмов. Решение полностью вписывается в стратегии импортозамещения, цифрового суверенитета и цифровой экономики.

ЛИТЕРАТУРА

1. Бухарин В.В. Компоненты цифрового суверенитета Российской Федерации как техническая основа информационной безопасности. // Вестник МГИМО-Университета. - 2016. - № 6 (51) [Электронный ресурс]. Режим доступа: <http://www.vestnik.mgimo.ru/contents/6-51-2016>
2. Басс Л., Клементс П., Кацман Р. Архитектура программного обеспечения на практике. 2-е изд. - СПб.: Питер, 2006. — 575 с.
3. Калентьев А. А., Тюгашев А. А. ИПИ/CALS технологии в жизненном цикле комплексных программ управления. - Самара: СНЦ РАН, 2006. - 285 с.
4. Тюгашев А.А., Ильин И.А., Ермаков И.Е. Пути повышения надежности и качества программного обеспечения в космической отрасли // Управление большими системами. Выпуск 39. — М.: ИПУ РАН, 2012. — С. 288-299.
5. Янг С. Алгоритмические языки реального времени. Конструирование и разработка. - М.: Мир, 1985. - 400 с.
6. Антамошкин А. Н., Колташев А. А. Технологические аспекты создания бортового программного обеспечения спутников связи. // Вестник СибГАУ им. акад. М. Ф. Решетнева. Вып. 6 - Красноярск: СибГАУ, 2005.
7. Ширяев А. В. Компилятор O7 // Доклад на конференции День Оберона 2017, ИЯИ РАН. Видео и презентация доклада. [Электронный ресурс]. Режим доступа: <http://oberoncore.ru/oberonday/2017>
8. Toyota: 81 514 нарушений в коде. [Электронный ресурс]. Режим доступа: <https://habrahabr.ru/company/pvs-studio/blog/310862/>
9. Ермаков И. Е. АСУТП в КП/ББ на платформе ERSY Control // Доклад на конференции День Оберона 2017, ИЯИ РАН. Видео и презентация доклада. [Электронный ресурс]. Режим доступа: <http://oberoncore.ru/oberonday/2017>
10. Оберон и промышленная автоматизация в агропроме [Электронный ресурс] — Режим доступа: http://www.inr.ac.ru/~info21/oberon_innovation/oberonAgro.htm
11. Темиргалеев Е. Э. Блэббок-Оберон — оптимальное решение для современных задач ИТ-образования. // См. данный сборник.