

**А.В. Крючков**

(Академия ГПС МЧС России; e-mail: hook66@list.ru)

## **ДОСТОИНСТВА И НЕДОСТАТКИ СОВРЕМЕННЫХ МЕТОДОВ СИНТЕЗА СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (часть 3)**

*Проведён анализ используемых в современной практике методов синтеза специального программного обеспечения.*

*Ключевые слова: программное обеспечение, методика разработки приложений.*

**A. V. Kruchkov**

## **ADVANTAGES AND DISADVANTAGES OF MODERN SYNTHESIS METHODS OF SPECIAL SOFTWARE (part 3)**

*The analysis of usually used synthesis methods of special software is carried out.*

*Key words: software, method of applications development.*

Статья поступила в редакцию Интернет-журнала 31 марта 2015 г.

Проблемы синтеза специального программного обеспечения (СПО), как составной части программного обеспечения (ПО) автоматизированных систем управления производством (АСУП), всегда связаны с использованием разных классов инструментальных средств разработки ПО: систем программирования (СП), языков программирования (ЯП), систем управления базами данных (БД, СУБД). Их использование необходимо для синтеза СПО автоматизированных рабочих мест (АРМ).

В статье автор попытался обобщить и кратко описать методики создания СПО, используемые в настоящее время для синтеза СПО, с которыми ему довелось познакомиться и работать. Статья является продолжением двух предыдущих статей на ту же тему и содержит обзор ряда инструментальных средств, используемых для создания исходного кода.

Для АСУП крупных предприятий синтез СПО традиционно представляет собой чрезвычайно трудную и нетривиальную задачу [1], осложнённую тем, что технологии и способы синтеза СПО непрерывно развиваются.

Автор не ставил себе целью провести критический анализ и сравнение полного списка технологий и способов реализации СПО. Поэтому в данной статье будут приведены некоторые важные направления и подходы технологий синтеза СПО, не рассмотренные в предыдущих статьях. В статье рассматриваются инструментальные средства, не относящиеся к низкоуровневым ЯП (типа ассемблер и макроассемблер), языки высокого уровня (ЯВУ) второго поколения (процедурные, декларативные и логического программирования по классификации, приведенной в [2], с. 8-15), а также языки сценариев. Подробный анализ таких средств приведён в [2, 3].

**Язык программирования Java, технология J2EE/EJB** (Java 2 Enterprise Edition/Enterprise Java Beans [4], с. 146, [6], с. 620, 642, [1], с. 455) – методика синтеза СПО, целью которой является создание распределенных переносимых объектно-ориентированных программ (компонент) для различных программных платформ, а также методика передачи данных и взаимодействия компонент (beans). Единственная в настоящий момент достойная альтернатива программным продуктам фирмы Microsoft по программированию в Интернет. Использует технику перевода программ в независимый от аппаратуры байт-код, исполняемый там, где есть инструмент работы с ним. Техника работы – клиент-сервер. Хранилище компонент (beans) используется в качестве сервера, который предоставляет клиентам (приложениям пользователей) компоненты через обращение к их интерфейсу. Обработкой запросов занимается Java Virtual Machine (JVM) через Java Server Pages (JSP). JSP – написанные на ЯП Java, как их называют, скрипты – программы на данном ЯП. Для специфицирования интерфейса компонент используется Interface Description Language (IDL). В последнее время в связи с развитием беспроводных вычислительных сетей JSP используют Wireless Application Protocol (WAP), который описывают на Wireless Markup Language (WML).

**Достоинства** (полный список достоинств данной платформы синтеза СПО подробно описан в специальной технической литературе):

- работа на большинстве известных программных платформ;
- простая расширяемая спецификация;
- использование технологии "клиент-сервер";
- наличие большого числа специальных библиотек;
- возможность создания динамических веб-страниц;
- увеличение производительности веб-страниц за счет использования техники home/remote interface;
- возможность координации транзакций с участием нескольких процессов в распределенной системе;
- возможность загрузки в и выгрузки из БД данных, а также работы с различными типами БД (JDBC), возможность создания запросов к БД с несколькими многозначными маршрутами;
- автоматическая организация EJB-сервером распределенной передачи данных, защиты от несанкционированного доступа, управление транзакциями, параллельной обработки и устойчивости прикладной части.

**Недостатки:**

- разработка и сопровождение крупных (свыше 20 АРМ) приложений требует значительных усилий, так как увеличивает число используемых компонент в десятки раз по сравнению с простыми (1, 2 АРМ) приложениями;

- уникальность проектов построения СПО с помощью данных средств и, в связи с этим, частичное отсутствие возможности повторного использования его компонент в других проектах; серьезную проблему представляет собой выбор тех компонент, которые могут использоваться повторно, а также количество повторно используемых компонент;

- зависимость объектов программ и компонент от предметной области;  
- катастрофическое увеличение числа вызовов компонент за счет отсутствия доступа к локальным вызовам сущностей в большинстве контейнеров.

**Распределённое программирование “Microsoft .NET Framework”** (технология .NET) – методика синтеза СПО, целью которой является создание повторно используемых компонент ПО на разных объектно-ориентированных ЯП (ООЯП), позволяющих создавать неоднородные распределенные системы СПО. ООЯП при этом используют общие типы данных и одинаковый набор функций. Данная технология стала ответом Microsoft на появление технологии J2EE/EJB и исповедует сходные принципы реализации СПО ([5], с. 241, [2], гл. 1).

**Достоинства** (список достоинств данной платформы синтеза СПО во многом аналогичен достоинствам технологии J2EE/EJB):

- работа с большинством известных ООЯП за счет CLI (Common Language Infrastructure) – общеязыковой инфраструктуры, – CLR (Common Language Runtime) – общеязыковой среды выполнения, – CLS (Common Type System) – общей системы типов – за счет чего библиотеки базовых классов .NET могут работать на любых ООЯП, поддерживающих ее;

- межъязыковая отладка программ;  
- использование технологии "клиент-сервер" при вызове методов;  
- наличие большого числа специальных библиотек;  
- возможность создания динамических веб-страниц и работы с XML;  
- увеличение производительности веб-страниц за счет использования техники home/remote interface;

- возможность загрузки в и выгрузки из БД данных, а также работы с различными типами БД (JDBC), возможность создания запросов к БД с несколькими многозначными маршрутами;

- возможность координации транзакций с участием нескольких процессов в распределенной системе;

- не требует изменения реестра для регистрации компонент;  
- жестко привязывает DLL-библиотеки к исполняемому файлу.

**Недостатки:**

- высокие требования к аппаратному обеспечению;  
- зависимость DLL-библиотек от предметной области;

- разработка и сопровождение крупных (свыше 20 АРМ) приложений требует значительных усилий, так как увеличивает число используемых компонент в десятки раз по сравнению с простыми (1, 2 АРМ) приложениями;
- серьезную проблему представляет собой выбор тех компонент, которые могут использоваться повторно, а также количество повторно используемых компонент;
- реализация для ограниченного числа ЯВУ;
- недостаточная устойчивость в работе по причине новизны подхода и недостаточного времени для устранения эксплуатационных недоработок.

**Распределенное программирование OMG** (и других организаций) – методика синтеза СПО, целью которой является создание повторно используемых компонент ПО, позволяющих создавать неоднородные распределенные системы СПО. Неоднородные системы – СПО АРМ, создаваемые на различных инструментальных средствах, но работающих в рамках одной АСУП. OMG (Object Management Group) – группа управления объектами – коммерческая организация, состоящая из более, чем 800 членов, выполняющая роль координатора в межоперационной совместимости программных продуктов. Эта группа создала спецификацию CORBA (Common Object Request Broker Architecture) – обобщенную архитектуру брокера объектных запросов – для достижения поставленной цели. Он предусматривает использование большого числа служб (программ поддержки) и наличие у повторно-используемых блоков (объектов) стандартизованного интерфейса, позволяющего использовать разработки разных специалистов в разных уголках Интернет, как удаленные объекты собственного программного кода.

**Достоинства:**

- широкое распространение и реализация во многих ООЯП;
- поддержка пользователей производителями ПО;
- стандартизация программирования на уровне более высоком, чем ЯВУ;
- часть реализаций в рамках данной технологии не зависят от предметной области и сходны с АОП.

**Недостатки:**

- резкое увеличение числа используемых библиотек при разработке крупных (более 20 АРМ) АСУП и как следствие затруднение разработки;
- обозначение одинаковыми названиями разных понятий в разных вариантах стандартов;
- большая и сложная спецификация (более 2000 страниц);
- зависимость объектов программ и компонент от предметной области;
- катастрофическое увеличение числа вызовов компонент за счет отсутствия доступа к локальным вызовам сущностей в большинстве контейнеров.

### ***Программирование в объектно-ориентированных СУБД (ООСУБД)***

– методика синтеза СПО, целью которой является соединение достоинств ООЯП с возможностями хранения объектов в БД.

Достоинства:

- использование одноуровневой памяти при работе с объектами;
- хранение объектов в полях БД;
- объектно-ориентированный подход к моделированию данных.

Недостатки:

- сложность изменений связанных объектов внутри БД;
- дерево объектов сильно зависит от предметной области;
- хранение ссылок на промежуточные объекты представляет проблему;
- несогласованность с "обычаями и процедурами, принятыми в коммерческих БД сегодняшнего дня" ([6], с. 99);
- объектно-ориентированный подход к моделированию данных в коммерческих СУБД, по мнению многих специалистов в этой области, не имеет смысла, так как обычно во всех программах СПО современных СУБД нет необходимости скрывать или ограничивать доступ к отдельным атрибутам объекта;
- отсутствие реализации в ООСУБД программ-инструментов, обеспечивающих хранение неактивных объектов в оперативной памяти и их последующую загрузку при необходимости использования, в то время как в существующих СУБД эти инструменты реализованы ([6], с. 747);
- в технологии СОМ модель объектов, как промышленный стандарт, пока не стабильна и изменяется ([6], с. 760);
- для написания запросов в ООСУБД необходимо использование SQL3, состоящего из традиционного SQL2 и средств объектно-ориентированной поддержки; такие программные инструменты в существующих СУБД отсутствуют ([6], с. 760);
- активное использование ООСУБД сдерживается обилием приложений, созданных при помощи традиционных СУБД и трудностями их перевода на новые технологические рельсы ([6], с. 162);
- невозможность хранения версий методов объектов из-за частичной (а не полной, как хотелось бы) поддержки объектов ООСУБД ([6], с. 770);
- ООП заставляет писать одинаковые части программного кода для разных объектов много раз; зачастую обойти это ограничение нельзя по причине необходимости явного указания ссылок на объект.

***Язык и система программирования и создания АРМ на базе продуктов компании IC*** (встроенный текстовый предметно-ориентированный ЯП) – методика синтеза СПО, целью которой является создание настраиваемых АРМ ряда заранее определенных предметных областей (бухгалтерии, кадров, материального учёта и снабжения, управления предприятием) на базе конфигурирования базового пакета программных средств. Впитала в себя некоторые черты CASE-средств и методик (технологий), описанных выше. Сходная методика используется рядом компаний, выпускающих продукцию на базе продуктов фирмы Oracle, InterSystems и др. Имеет встроенный специфический язык программирования.

### ***Достоинства:***

- удобство работы;
- уменьшение затрат на программирование и переход к описанию предметной области средствами программного продукта;
- возможность реализации семейств АРМ сходной функциональности в рамках одного проекта СПО, использующего настройки конфигурации;
- возможность быстрого создания отчетов;
- возможность описания БД сложной (нереляционной) структуры.

### ***Недостатки:***

- зависимость конфигурации от предметной области и потребностей конкретного пользователя внутри нее;
- сложность обучения технике программирования и взаимодействия с другими платформами ввиду недостаточного описания в литературе;
- ограниченные (в сравнении с XML и Java) возможности по созданию динамических веб-страниц.

Помимо этого, можно упомянуть, не раскрывая достоинств и недостатков, еще несколько методик программирования.

***Адаптивное программирование*** – методика синтеза СПО, аналогичная порождающему программированию, согласно которой одни стандартные блоки выражаются при помощи графов, а другие – ссылаются на них при помощи стратегий прохождения. Стратегией прохождения принято считать путь в графе ([7], [4], с. 255).

***Автоматическое программирование*** – методика синтеза СПО, аналогичная порождающему программированию, целью которой является высокоуровневая оптимизация. Она заключается в привлечении, хранении и использовании большого количества знаний о предметной области и технологии искусственного интеллекта для решения задач среднего размера ([4], с. 27).

***Автоматное программирование*** – методика синтеза СПО, целью которой является представление частей СПО в качестве конечных автоматов и описание их взаимодействия по правилам построения входных и выходных алфавитов, аналогичных принятым в радиоэлектронике.

***Программирование с помощью ERP-систем*** – методика синтеза СПО, целью которой является настройка выбранной к внедрению на объекте автоматизации (предприятия или организации) ERP-системы на конкретные условия её дальнейшей эксплуатации ([8], с. 19). ERP-системы (Enterprise Resource Planning) – это программные системы планирования ресурсов предприятия. Или, как сказано в [9], это – "корпоративная информационная система (КИС), предназначенная для автоматизации учёта и управления", основная функция которой – создание "единого хранилища данных, содержащего всю корпоративную бизнес-информацию и обеспечивающего одновременный доступ к ней любого необходимого количества сотрудников предприятия, наделённых соответствующими полномочиями". Здесь же отмечается, что "четкого определения

ERP-системы и внятного набора критериев для её идентификации не существует. Сегодня понятие "ERP-система" или "система класса ERP" серьезно размыто – мало того, что оно, по сути, является маркетинговым, его сейчас используют для обозначения довольно сильно разнящихся между собой интегрированных информационных систем". Даже есть коммерческие методики, позволяющие определить принадлежность того или иного программного продукта к данному классу.

Тем не менее, определённое заключение по этому поводу, в плане синтеза СПО в контексте предлагаемого исследования, сделать можно. В целом это некоторые программные объекты, имеющие крупные стандартные настраиваемые или неизменяемые компоненты, а также встроенные языки, позволяющие писать на ЯП дополнения к данным компонентам. СУБД в рамках ERP-систем используются, как правило, не "родные" (известных вендоров или общедоступные). Обычно "используемый в ERP-системах программный инструментарий позволяет проводить производственное планирование, моделировать поток заказов и оценивать возможность их реализации в службах и подразделениях предприятия, увязывая его со сбытом" [9].

#### ***Достоинства:***

- использование в рамках локальной АСУП единой интегрированной программы вместо нескольких разрозненных;
- сокращение и унификация используемых для автоматизации аппаратных средств;
- переход всех пользователей внутри ERP-системы на единую платформу хранения данных и поддержка её со стороны внедряющей организации;
- повышение эффективности взаимодействия пользователей внутри ERP-системы;
- более эффективный контроль деятельности персонала и прогнозирование затрат и доходов, связанных с основной деятельностью;
- улучшение обслуживания внешних по отношению к использующей ERP-систему организации клиентов предприятия.

#### ***Недостатки:***

- высокая первоначальная стоимость всех ERP-систем ("Как подсчитали в PanoGama Consulting, в среднем ERP-проекты занимают около 20 месяцев, а их стоимость составляет 8,5 млн USD", [9]); как следствие ERP-система может внедряться только в крупных организациях ([10]);
- низкий уровень коэффициента "эффективность-стоимость" и его размытость: "...оценить эффективность внедрения информационной системы предприятия нелегко. Все существующие подходы следует рассматривать с известной степенью приближения, так как ни один из них не дает полной картины";
- высокая стоимость владения и эксплуатации;
- сложность внедрения, в том числе из-за необходимости изменения уже используемых программных продуктов и регламентирующих документов, характер работ в рамках которых не предусмотрен во внедряемой ERP-системе;
- хранение данных в единой БД снижает живучесть системы;

- зависимость от вендоров, создающих СУБД и ЯВУ, с помощью которых разработчики создают ERP-систему;
- эффективность работы системы может быть легко нарушена одним пользователем, подразделением или партнёром [9];
- в некоторых случаях неоправданно растёт объём вводимых данных и время выполнения стандартных операций;
- "неадекватность функций системы целям и задачам предприятия";
- невозможность настройки на конкретную область деятельности во многих случаях;
- возможности индивидуальной доработки ограничены;
- "ERP может быть слишком жестким и громоздким инструментом для некрупных организаций"; "по оценкам META Group средняя мировая сумма затрат на внедрение ERP-системы составляет около \$15 млн, а средняя стоимость одного рабочего места – более \$50 тыс.";
- "фактическое отсутствие отраслевых стандартов и практик в области управления бизнес-процессами; результат: невозможность достаточно эффективно применять и развивать функциональность, заложенную в отраслевые решения";
- "недостаточный уровень компетенций в области технологий менеджмента и ИТ у многих руководителей и лиц принимающих решения".

Есть также и другие недостатки у данного класса методов автоматизации (но не синтеза СПО), например, организационные: намеренный "выбор заведомо неподходящих или неоптимальных решений в области автоматизации системы управления". Подобные практики сложились благодаря многолетнему целенаправленному выдавливаю специалистов, имеющих, как принято сейчас говорить, нужные компетенции в данном вопросе, а также созданию условий, при которых данные специалисты ни при каких обстоятельствах не могут претендовать на должности в области технологий менеджмента и ИТ, включая должности руководителей, принимающих соответствующие решения. Но на предмет исследования данные недостатки конкретной методики автоматизации существенного влияния как бы не оказывают. Поэтому сосредоточимся на существенных для изучения процесса синтеза СПО деталях.

Таким образом, методики синтеза СПО АСУП многочисленны и разнообразны. Но ни одна из них не решает все проблемы, что видно из перечня недостатков каждой из них, составленного на основе широко представленной отечественной и зарубежной технической литературы.

Большинство из методик не допускают совместного использования, наряду с ними, в создании таких АСУП других методов и методик, что особенно заметно на примере ERP-систем. Допускающие такое использование методики либо недостаточно широко представлены в существующих ООЯП, либо недостаточно проработаны. Общие недостатки, заставляющие искать новые методы оптимизации процессов при создании исходного кода СПО АСУП, можно сформулировать следующим образом:



- сложность взаимодействия во многих случаях;
- библиотеки компонентов и объектов содержат миллионы элементов со всевозможными комбинациями свойств;
- наличие ограничений в использовании, связанных с коммерческими соображениями производителей инструментальных средств, программистов и распространителей;
- зависимость реализованного проекта СПО от предметной области и потребностей конкретного пользователя внутри нее;
- сложности в сопровождении большинства систем после сдачи заказчику, в том числе связанные с быстрым обновлением версий инструментальных средств и технологий;
- как следствие предыдущего недостатка, разработка и сопровождение СПО крупных (свыше 20 АРМ) АСУП требует значительных усилий, иногда в десятки раз больших в сравнении с простыми (1, 2 АРМ) приложениями.

Использование описанных в статье средств автоматизации предприятий позволяет обеспечивать преемственность только незначительной части синтезируемого СПО.

#### Литература

1. *Басс Л., Клементс П., Кацман Р.* Архитектура программного обеспечения на практике. Пер. с англ. СПб.: Питер, 2006.
2. *Зыков С.В.* Основы современного программирования. Разработка гетерогенных систем в Интернет-ориентированной среде (на основании курса лекций по информационным системам на факультете информационной безопасности МИФИ): учеб. пособие для вузов. М.: Горячая линия – Телеком, 2006.
3. *Себеста Р.У.* Основные концепции языков программирования. Пер. с англ. М.: изд. дом "Вильямс", 2001.
4. *Чарнецки К., Айзенкер У.* Порождающее программирование. Методы, инструменты, применение, Пер. с англ. СПб.: Питер, 2005.
5. *Избачков Ю., Петров В.* Информационные системы: учеб. для вузов. СПб.: Питер, 2005.
6. *Крёнке Д.* Теория и практика построения баз данных. Пер. с англ. СПб.: Питер, 2005.
7. *Lieberherr K.* Connections between Demeter/Adaptive Programming Aspect-Oriented Programming, Informal note. <http://www.ccs.neu.edu/home/lieber/connection-to-aop.htm>.
8. SAP ERP. Построение эффективной системы управления. Пер. с англ. М.: Альпина Бизнес Букс, 2008.
9. ERP-системы. <http://www.tadviser.ru/index.php/%D1%F2%E0%F2%FC%FF:ERP-%F1%E8%F1%F2%E5%EC%FB>.
10. *Дзюба С.А.* "ERP-системы – мэйнстрим или тупиковая ветвь?". <http://www.cfin.ru/itm/kis/criticism.shtml>.