

А.В. Крючков

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

РЕКОМЕНДАЦИИ ПО ПРИМЕНЕНИЮ МЕТОДОВ СИНТЕЗА СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КРУПНОЙ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

Показано, как применять методы универсализации синтеза единичных программных систем специального программного обеспечения в крупных автоматизированных системах управления предприятиями с наибольшей эффективностью.

Ключевые слова: программное обеспечение, автоматизированные системы управления предприятиями.

A.V. Kruchkov

RECOMMENDATIONS FOR USE OF SYNTHESIS METHODS OF SPECIAL SOFTWARE OF LARGE AUTOMATED CONTROL SYSTEMS OF ENTERPRISE

It shows how to apply the universalization methods of synthesis of individual software systems special software in large automated control systems of enterprise with the greatest efficiency.

Key words: software, automated enterprises control systems.

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

Основными задачами при эксплуатации автоматизированных систем управления предприятиями (АСУП) являются повышение производительности труда в промышленности и снижение трудозатрат конкретных исполнителей за счёт использования автоматизированных рабочих мест (АРМ). Для их создания используются системы программирования (СП) и языки программирования (ЯП), входящие в состав определённых информационных технологий (ИТ). Результатом использования ИТ для создания АРМ в АСУП является специальное программное обеспечение (СПО). Учитывая, что СПО для крупной АСУП состоит из СПО для отдельных АРМ, будем называть СПО, созданное для одного из АРМ, единичной программной системой (ЕПС).

Сам по себе синтез СПО как процесс, в котором участвуют программисты, менеджеры, тестировщики, технические писатели, представляет собой некий набор действий. В центре данного процесса – разработка ЕПС, которой посвящены многие источники в отечественной и иностранной технической литературе. Одной из наиболее известных моделей применения методов по управлению процессами в области синтеза СПО является интегрированная модель зрелости процессов, в английской аббревиатуре – СММ (Carcacity Maturity Model, [1-3]). Данный набор методов по управлению синтезом СПО был разработан Институтом программной инженерии при Университете Карнеги-Меллона по заказу Министерства обороны США в целях определения качества работ по созданию ЕПС на основе анализа применения определённых методов универсализации в синтезе СПО различными компаниями.

Вместе с тем, описанные в данной модели подходы к универсализации синтеза СПО касались в основном управления, а вопросы технического характера при синтезе СПО в данный проект не попали. Однако, осуществление глубокой универсализации синтеза СПО в крупных АСУП невозможно без внимательного рассмотрения процессов в труде конкретного исполнителя (программиста), реализующего АРМ конкретного исполнителя в АСУП. Для прояснения ситуации необходимо более детально остановиться на требованиях к ЕПС и их группам, которые можно объединить в классы задач автоматизации.

Иерархические требования к системе ЕПС, образующих СПО крупной АСУП, можно объединить в единый свод данных и правил их формирования. Такой объект можно назвать "паспортом информационной единицы хранения (ИЕХ)". Учитывая, что АРМ работает с базой данных (БД) или таблицей БД, необходимо ввести блок требований, описывающих состав информации для данной ИЕХ по работе с некоторой универсальной (или абстрактной) системой управления БД (СУБД).

Для работы с такой системой пользователя АРМ на конкретном рабочем месте ЕПС (приложение) должно выполнять ряд функций. Анализ различных проектов и их реализаций позволил выявить ряд сходных черт в семантике интерфейсной части АРМ в СПО крупных АСУП. В связи с этим целесообразно обобщить эти данные в виде определённого перечня или набора правил работы с ИЕХ на АРМ конкретного пользователя, с тем чтобы затем в процессе синтеза новых программных систем в рамках СПО АСУП или совершенствования их работы можно было бы заранее разрабатывать инструментарий приложений вне зависимости от его предметной области.

Семантическим элементом интерфейса (СЭИ) уместно называть то действие ПО, которое пользователь выбирает, общаясь с ПО, и которое реализует одну понятную ему функцию. Такое действие обычно совершается при нажатии кнопки, переходу по гиперссылке или при выборе пункта меню, хотя возможны и иные способы предоставления пользователю функций ввода команд приложению в рамках конкретной ЕПС.

На основании анализа данных по разработанному СПО, информацию о котором автору удалось получить лично в виде исходного кода ЕПС, был сформирован универсальный перечень СЭИ (УПСЭИ) для абстрактной ЕПС, сгруппированный по категориям. Перечень позволяет не только разрабатывать новые приложения, но и достаточно точно оценивать качество сдаваемого программного продукта заказчиками, не имеющими достаточной квалификации в области синтеза СПО. В результате применения УПСЭИ в процессе синтеза СПО возможно решение задачи преемственности в разработке. Но для корректного синтеза СПО, помимо УПСЭИ, необходимо введение понятия "системы без программирования" (СБП-системы), основанной, в том числе, и на функциях УПСЭИ.

Если рассматривать область деятельности, связанную с синтезом СПО очень крупных АСУП (несколько сотен ЕПС и более) большим количеством разных программистов с помощью различных инструментальных средств

и их версий, то в состав методов необходимо включить систематическое описание процесса деятельности программистов и их руководителей. Этот документ необходимо сопроводить подробным описанием характеристик и логической структуры данного процесса.

Условия работы в этой области требуют от программистов привлечения колоссального объёма информации, а для руководителей проектов, содержащих сотни ЕПС, её объём больше на три порядка. При этом руководители коллективов разработчиков СПО (организаций-разработчиков) вынуждены контролировать творческие процессы в головах творческих людей, имея для этого только те метрики, которые характеризуют внешнюю часть деятельности программистов. Логическая (систематическая) структура деятельности может быть определена следующим образом. Субъекты – программист, создающий ЕПС, руководители коллективов программистов, руководители организаций-разработчиков СПО. Объект – СПО АСУП. Предмет – деятельность отдельных программистов и их больших групп по синтезу СПО АСУП и руководство их работой. Два процесса в ходе осуществления деятельности. Для программиста – написание исходных текстов, их отладка и тестирование. Для руководителя – контроль этой деятельности. Средства – инструментальные средства синтеза СПО и предлагаемые их разработчиками методы работы программистов с ними. Методы деятельности субъектов по созданию объектов деятельности – написание исходных текстов программ программистами в СП инструментальных средств и их отладка, создание конфигураций ЕПС в среде CASE-средств или в среде ERP-систем, проектирование и реализация компонент и аспектов, а затем пробная совместная эксплуатация различных ЕПС в ходе испытаний перед сдачей АСУП заказчику. Помимо этого управление деятельностью программистов и её контроль. Результат деятельности – предсказуемо работающее СПО в виде большой группы ЕПС (АРМ) АСУП, удовлетворяющее требованиям пользователей.

Структура деятельности в отношении синтеза СПО крупной АСУП при применении описанного подхода может выглядеть следующим образом. Прикладные программисты (абстрактные или среднестатистические) сосредотачиваются на синтезе и доводке (отладке и доработке) инструментария в виде СБП-системы, реализованной на базе структуры данных паспорта предметной области, УПСЭИ и концепции составляющих её диалоговых компонентов. СБП-система создаётся программистами для каждого из инструментальных средств. Они вообще не работают с ТЗ на АРМ, а занимаются только данными проблемами.

Помимо этого в данную структуру деятельности следует внести ещё одно звено, называемое продвинутым пользователем. Они берут на себя большую часть работы по определению требований пользователей и их воплощению в ЕПС в ходе работы с СБП-системой. Контроль у руководителей существенно упрощается, так как не нужно теперь знать многое о программировании и работать с большим количеством творческих людей, постоянно отклоняющихся от намеченного им "маршрута".

Продвинутые пользователи после непродолжительного обучения порядку работы с СБП-системой создают на её основе реальные ЕПС, дорабатывают их в соответствии с изменениями в предметной области и осуществляют контроль над СПО АСУП в целом. Они также разрабатывают шаблоны для СБП-систем по материалам синтеза реальных АРМ по реальным ТЗ и ведут их учёт. В состав их деятельности по синтезу ЕПС включаются процессы:

- подготовки паспорта ИЕХ в соответствии с утвержденным ТЗ;
- описание в диалоговом режиме необходимых структур данных, расчётов и отчетов;
- проведение тестирования ЕПС и СБП-системы и доработка ЕПС, в случае необходимости, перед сдачей заказчику;
- осуществление доработок СПО после его сдачи в эксплуатацию: переработка ТЗ, изменения в паспорте ИЕХ, изменения в ЕПС;
- документальное сопровождение данного вида работ.

Применение данных процессов уменьшает число элементарных операций в проектах по созданию ЕПС. Для получения большего выигрыша в снижении трудозатрат необходимо двигаться по пути объединения сходных для различных ЕПС действий программистов. Такой подход отчасти использовался в прошлом и получил название "повторное использование кода" [2]. Его рекомендовали к использованию многие руководители коллективов программистов и преподаватели. Однако, ни один из методов данного направления не получил широкого применения в реальном синтезе ПО. Это было связано в основном с характером повторно используемого исходного кода. Общие принципы данного подхода были всегда очевидны, но детали реализации упирались в многочисленные трудности общечеловеческого и технологического свойства. Не только каждый программист интерпретировал по своему решение некоторой задачи автоматизации, выражая это в исходном коде, но и языки и средства программирования, меняясь от версии к версии, способствовали тому, чтобы код невозможно было использовать повторно.

Тем не менее, существует принципиальная возможность путем моделирования работы программистов создать модель повторного использования кода, не зависящую от ЯП. Для этого необходимы организационные и технические меры. К организационным мерам можно отнести:

1. Создание руководств по написанию программистами инструментария на ЯП (средстве программирования или языке СУБД), в которых будут определены основные направления их действий в этом направлении (априорного задания статически детерминированных структур для локальных моделей данных, эффективностей и ресурсов). Данные руководства должны разрабатываться для всех используемых в организациях-разработчиках ЯП.

2. Создание механизмов контроля за правильным применением данных руководств в ходе синтеза СПО. Это методики "второго" уровня для руководителей коллективов разработчиков. В них должны быть описаны принципы подготовки персонала для синтеза и его дальнейшего использования по прямому назначению, исходя из отношения к смоделированному в исследовании инст-

рументарий программиста. Те специалисты, которые будут формировать инструментарий, используя существующие средства разработки, должны признаваться специалистами высокой квалификации и требования к ним должны включать описание действий по созданию инструментария: написанию кода, его отладке, тестированию и сдаче в соответствующие, использующие его, подразделения. Данные специалисты не должны привлекаться к непосредственной работе с заказчиками. Соответственно специалисты, использующие инструментарий, должны быть ориентированными на его тестирование и работу с заказчиками. Собственно, они могут и не знать, как писать программы на соответствующем средстве разработки. Их основная задача – применение созданного инструментария.

3. Соответственно необходимая организация переподготовки персонала, разрабатывающего СПО АСУП, должна включать отдельные программы для подготовки разработчиков инструментария и для разработчиков ЕПС.

4. Во всех заинтересованных организациях необходимо провести описание предметных областей, которые могут быть подвергнуты автоматизации, в терминах предложенной модели описания предметной области. Для каждой из этих областей необходимо создание паспортов предметной области. Сводный набор паспортов предметной области вместе с датой составления и указанием документов (законов, приказов, распоряжений, инструкций), на основании которых данные области регулируются, должен храниться в организации. В подразделении, отвечающем за ИТ в организации, должен вестись протокол внесения изменений в данный документ и назначен специалист по его сопровождению.

5. В связи с тем, что данные в паспортах предметных областей необходимо регулярно обновлять, необходимо создание руководств по модификации описаний предметной области с последующим внесением в набор паспортов предметных областей организации. Поэтому необходимо определение порядка согласования и протоколирования изменений, вносимых в данные.

6. Следует отметить регламентацию ещё двух организационных процедур: определение порядка согласования и протоколирования изменений, вносимых в программные модули инструментария, а также определение порядка согласования и протоколирования изменений, вносимых в ЕПС.

К техническим мерам следует отнести:

1. Создание (разработка) для каждого из используемых в организациях-разработчиках ЯП инструментария разработки программиста по описанным выше (в предыдущих главах) правилам:

- инструментария формирования структур данных предметной области (набора программ или программных комплексов, которые бы позволили свести всю работу с данными из паспортов предметной области к внесению изменений в соответствующие поля карточек);

- инструментария остальных компонент СБП-системы;

- доработки УПСЭИ под конкретные группы предметных областей.

2. Создание файлов конфигурации, содержащих метаданные.

Данные файлы позволят хранить ряд стандартных настроек для одной или нескольких сходных предметных областей. Очевидно, при наличии развитых, определённых иерархически для ряда предметных областей, конфигурационных файлов, для работы с ними понадобится отдельный специалист. Наличие такого специалиста (консультанта по метаданным) позволит существенно ускорить процесс изначальной генерации приложений или ЕПС в СПО крупных АСУП.

Таким образом, применение описанных рекомендаций по использованию методов универсализации синтеза СПО в крупных АСУП, в дополнение к уже существующим моделям управления в ИТ, позволит:

- существенно сократить затраты на управление процессом синтеза;
- вывести процесс контроля деятельности программистов на совершенно новый уровень, доведя творческий и плохо прогнозируемый процесс синтеза ЕПС до хорошо описанного производственного процесса;
- сократить издержки при программировании;
- добиться почти абсолютной преемственности в разработке;
- получить единую структуру и правила хранения информации в различных предметных областях.

Описанный в статье набор методов по универсализации синтеза СПО, как процесс управления ИТ, может быть классифицирован по 5-уровневой модели СММ как 4-й. Развитие описанных организационных мер по многим позициям позволит оптимизировать процессы универсализации синтеза СПО и уже говорить о 5-м уровне зрелости коллектива разработчиков, применяющего данные методы.

Литература

1. **Бирюков А.Н.** Лекции о процессах управления информационными технологиями: учеб. пос. М.: интернет-университет информационных технологий: БИНОМ. Лаборатория знаний, 2010.
2. **Стив Макконнелл.** Профессиональная разработка программного обеспечения, пер. с англ. СПб.: Символ-Плюс, 2007.
3. **Вендров А.М.** Проектирование программного обеспечения экономических информационных систем. М.: Финансы и статистика, 2005.
4. **Избачков Ю., Петров В.** Информационные системы: учеб. для вузов. СПб.: Питер, 2005.
5. **Басс Л., Клементс П., Кацман Р.** Архитектура программного обеспечения на практике, пер. с англ. СПб.: Питер, 2006.