

С.С. Великодний

Методологические основы реинжиниринга систем автоматизированного проектирования

Предложены новые теоретические результаты исследований, касающиеся развития процессов реинжиниринга систем автоматизированного проектирования, основной из которых – определение основ их новой методологии перепроектирования, охватывающей фундаментальные аспекты деятельности по их созданию.

New theoretical results of studies concerning the development of processes reengineering CAD/CAM/CAE-systems are proposed. The definition of a new foundations methodology for the CAD/CAM/CAE-systems redesign, which covers fundamental aspects of their creation is elaborated.

Запропоновано нові теоретичні результати досліджень щодо розвитку процесів реінжинірингу систем автоматизованого проектування, основним з-поміж яких є визначення основ нової методології перепроєктування цих систем, що охоплює фундаментальні аспекти їх створення.

Введение. Основные цели, поставленные перед системами автоматизированного проектирования (САП) любого назначения, – сокращение сроков проектирования объекта и персонала, необходимого для его осуществления, и как следствие – себестоимости готового объекта проектирования (ОП).

В результате всестороннего изучения ОП в рамках системы автоматизации проектных работ (САПР), разрабатываются модели и алгоритмы с максимально возможной точностью фиксирующие будущие закономерности формообразования ОП, и логики, происходящей в процессах при проектировании и его изготовлении.

В соответствии с принципом взаимодействия *человек–машина*, положенного в основу построения САПР, в техническом аспекте САПР можно определить как комплекс информационного, программного, технического и другого обеспечения системы, подчиненной единой методологии машинного проектирования.

Постановка проблемы

Для корректной постановки проблемы необходимо ввести ряд основополагающих понятий предметной области исследования.

Реинжиниринг (reengineering) САП – это эволюция системы путем ее коренного изменения с целью повышения удобства ее эксплуатации, сопровождения или изменения ее функций.

Реинжиниринг (РИ) включает в себя процессы реорганизации и реструктуризации САП, перевод отдельных компонентов системы в

иной (более современный) язык программирования, а также процессы модификации или модернизации структуры и системы данных. При этом архитектура САП может оставаться неизменной [1].

Метод РИ САП – целевое средство получения нового компонента путем выполнения последовательности операций внесения изменений, модернизации или модификации, а также перепрограммирование отдельных компонентов САП; реализуется совокупностью моделей, методов и процессов, изменяющих структуру и возможности компонентов, с целью получения компонента с новыми возможностями (новые компоненты идентифицируются именами, используемыми при создании компонентных конфигураций и каркасов САП).

С технической стороны РИ – это решение проблемы эволюции САП путем изменения ее компонентов и адаптации архитектуры к новой среде, в которой компоненты размещаются согласно конфигурации операционной системы.

Причиной эволюции может быть изменение языка написания САП (например: *Fortran*, *COBOL* или даже *C* и др.) с переходом на современные языки (*Java*, *C#*, *Python* и др.).

Однако с коммерческой стороны, РИ часто считают единственным способом сохранения унаследованных систем в эксплуатации, поскольку полная («с нуля») эволюция системы, – дорогая и рискованная процедура продления времени существования САП [2].

Анализ последних исследований и публикаций

Широкое распространение видов и типов новой компьютерной техники ставит перед специалистами важные задачи по эффективному ее использованию при создании программных и прикладных систем различного применения. Анализ потребности различных отраслей хозяйства и промышленности в программных системах показывает, что на практике их нужно более 3,5 млн различного наименования, назначения и сложности [3].

В связи с этим важнейший путь развития программных систем – совершенствование технологии программирования в целях перехода на промышленные методы их создания, характеризующиеся специализированными САП по производству программных продуктов.

Основная цель технологии реализации таких САП – четкая регламентация и организация работ по проектированию, разработке и изготовлению качественных программных продуктов и высокой производительности. Условием повышения производительности труда разработчиков САП есть использование готовых производственных ресурсов, а именно, компонентов повторного использования (КПИ) со стандартными или унифицированными интерфейсами, способными вырабатывать формализованный сбор разнородных языков программирования согласно заданным требованиям заказчиков. Предложенный подход использования таких компонентов есть частным случаем РИ, а, следовательно, может быть принят в качестве составной части РИ САП в структуре его методологии.

Идею сборки как конвейерного механизма фабрик программ предлагали Глушков В.М. [4–10] и Ершов А.П. (1967 г.) [11–13]. Далее (1985–1988 гг.) формирование нового вида программирования, а именно сборочного, происходило с участием Ющенко Е.Л.

Сегодня средства сборки модулей, программ и компонентов развиваются и совершенствуются. Научно-технической поддержкой парадигмы сборочного программирования можно считать результаты кандидатских диссертаций

в 2004–2008 гг. Задорожной Н.Т., Коваль Г.И., Коротун Т.М., Слабоспицкой О.А., а также подготовленной к защите докторской диссертации Грищенко В.Н. и кандидатской диссертации Поляничко С.Л. (под руководством Лаврищевой Е.М.).

Таким образом, сделан существенный вклад в развитие и совершенствование базовых положений программной инженерии, что способствовало формированию основ индустриального производства программных продуктов высокого качества.

Анализ последних исследований и публикаций показывает, что в ближайшие годы создание и адаптация разрабатываемых САП к различным отраслевым задачам значительно улучшится с учетом широкомасштабного применения готовых программных ресурсов повторного использования, аналогично тому, как это происходит в машино-, судо-, авиа- и приборостроении при сборке крупных изделий из готовых деталей.

Цель статьи – формирование основ методологии РИ САП посредством построения фундаментальной теоретической базы составных процессов, позволяющей поставить на промышленную основу выполнение РИ САП.

Критерии рентабельности РИ САП

Главное отличие между РИ и новой разработкой САП заключается в том, что описание системной спецификации начинается не с нуля, а с рассмотрения возможностей старой унаследованной системы, благодаря КПИ; согласно данным [14] повторное использование в четыре раза дешевле, чем новая разработка САП. Однако следует заметить, что в некоторых случаях, все же действительно выгоднее применить повторную разработку.

В связи с этим одной из задач, относящихся к процессу РИ САП, – есть формализация критериев рентабельности РИ, по которым, после построения определенных сравнительных характеристик, будет приниматься однозначное решение о применении или отклонении РИ. При этом следует заметить, что ни в коем случае нельзя строить сравнительные характеристики только на условиях, охватывающих лишь инструмента-

рий САП. Представленные критерии должны базироваться на всестороннем анализе всех видов обеспечения САП.

Логическая схема РИ САП

Следующей после принятия решения о применении РИ есть задача разработки логической схемы РИ (ЛСР) – производной от логической схемы проектирования (ЛСП) [2].

ЛСР начинается целевой программой и включает в себя алгоритмический выбор компонентов РИ. Целевая программа (эксперимент) объединяет внешнюю (формирование целей) и внутреннюю (моделирование) части.

При использовании ЛСР можно выделить следующие основные этапы:

- перевод исходного кода на устаревшем языке программирования на современную версию этого или другого языка;
- анализ САП согласно документированной структуры и функциональным возможностям системы;
- модификация структуры САП для наращивания новых свойств и возможностей;
- разбивка САП на модули для их группировки и устранения избыточности;
- изменение данных, с которыми работает САП.

В таком случае САП, к которой была применена ЛСР, приобретает признаки перехода на новый качественный уровень, состоящий из процессов с неформальными процедурами и системных исследований, методологически связанных с процессами имитационного и математического моделирования на современном техническом обеспечении.

Правильно выполненная ЛСР обязательно характеризуется достижением следующих результатов:

- снижение риска возникающих ошибок при будущем обновлении САП;
- снижение себестоимости продукта при участии КПИ при разработке новой САП и сокращении трудоемкости путем практически полного исключения рутинных операций программирования многих уже идентифицированных компонентов.

Планируется, что использование ЛСР САП даст значительное повышение эффективности применения его во всех отраслях их использования.

Принцип декомпозиции САП на подсистемы

Подсистемы САП определяют зону деятельности в системе проектирования с решением тех задач, что связаны с разработкой проектов для каждой из выделенных групп ОП. Каждая подсистема САП сочетает в себе деятельность по проектированию всех ОП одного вида и относится к одной определенной области проектирования.

В свою очередь, подсистемы САП имеют свое разделение на подподсистемы, т.е. на взаимосвязанные зоны, специализирующиеся на проектировании объектов различного назначения. Границы между подсистемами САП отделяют заказ на проектирование (вход в подсистему) от задания на проектирование, т.е. начальной стадии в проектом процессе и результатом. В итоге может возникнуть заказ на проектирование объекта в другой подсистеме.

РИ видов обеспечения САП

Разработка и дальнейшее развитие элементов и подсистем САП в различных отраслях производства – одна из главных задач продвижения конкретной отрасли на современный уровень мирового промышленного производства. Благодаря мощным вычислительным средствам САП с помощью информационно-поисковых систем, содержащих банки и базы данных готовых проектно-конструкторских решений, возможно достаточно быстро внести корректировки в необходимые параметры ОП (размеры, форма, порядок обработки и так далее), т.е. гибко переориентировать весь производственный процесс.

Однако элементы и подсистемы САП, подвергнутые РИ, дают значительную эффективность в том случае, когда результаты автоматизированного проектирования используются в производстве. Эффективность в данном случае объясняется тем, что при современных темпах развития науки и техники возникает противоречие между растущим уровнем научно-техничес-

ких достижений и существующими методами и техническими средствами проектирования.

Аспекты жизненного цикла САП

Жизненный цикл (ЖЦ) САП – это период времени между началом разработки и завершением ее использования, что характеризуется несколькими процессами, отражающими закономерности при разработке и применении программных средств. Различные подходы, методы, инструментальные средства могут вносить особенности в отдельные процессы, однако содержание ЖЦ САП вполне определенное. Наиболее общие стадии сведены в диаграмму профилей (см. рис. 1).

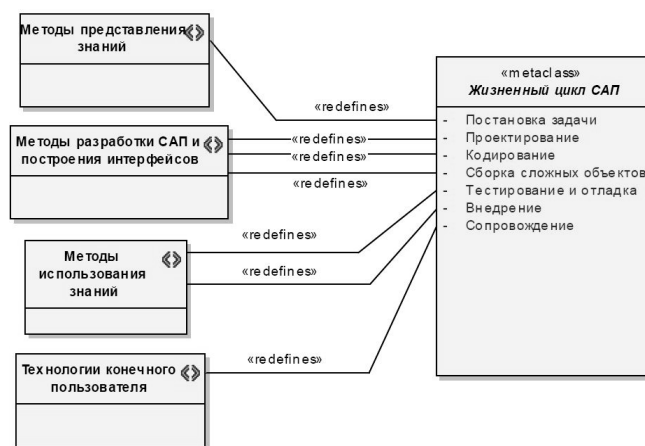


Рис. 1. Диаграмма профилей с представлением переопределения методов разработки и использования САП в рамках жизненного цикла

Главное отличие РИ от других методов повторного использования заключается в наличии *устойчивой обратной связи* (переопределения (*redefines*) – на рис. 1). Данная *обратная связь* означает, что в технологию использования разработанной САП входят методы применения этой технологии при проектировании других САП, где переопределение входит как составной элемент. Термин «устойчивая» подчеркивает, что методы применения разработанных подсистем в дальнейшей разработке САП – необходимое условие использования метода РИ.

Отметим, что методы РИ есть обобщением традиционных методов разработки, так как они дополнительно включают и методы построения интерфейсов между отдельно разработанными подсистемами.

Знания о конфигурации САП оформлены в виде абстракций общего, а также специального назначения и хранятся в активной библиотеке многоразового использования. Там же хранятся и производственные знания о способах тестирования, измерения, планирования и оценки отдельных компонентов и членов семейства САП.

Набором этих знаний определяются задачи САП и подходы к ее описанию и реализации. Каждая модель члена семейства САП представляется в некотором конкретном синтаксисе (например: *UML*, *OCL* и др.). Эта модель трансформируется от исходного к целевому представлению по следующей схеме (рис. 2):

- преобразование исходной модели к промежуточному (объектному) виду;
- преобразование исходной модели в целевую;
- превращение целевой модели в конкретный код.

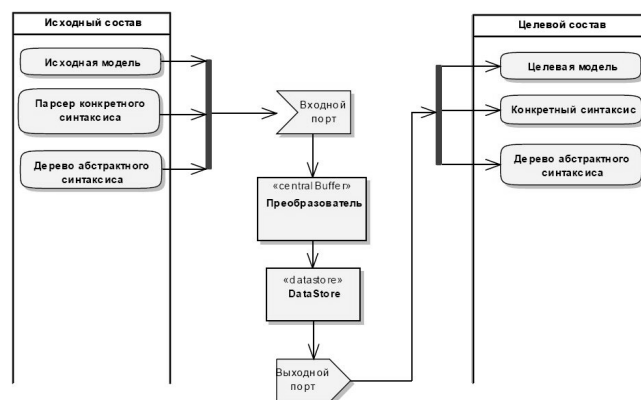


Рис. 2. Трансформация модели САП от исходного к целевому представлению

Цель генерации кода – обеспечение возможности его настройки на целевую платформу для получения исходного кода – осуществляется в два этапа. Первый этап – обеспечивает чтение конфигурации, по которой создается соответствующий генератор текущего преобразования, второй – использует созданный генератор для выполнения необходимых преобразований модели.

Последующая интеграция (сборка) исполняемого кода в САП осуществляется с помощью:

- использования шаблонов проектирования;

- сохранения сгенерированного и исправленного кода в отдельных накопителях;
- дизайна архитектуры САП, в которой определены сгенерированные артефакты;
- использования гибких методик склеивания сгенерированных и базовых частей кода.

Заключение. В методологическом плане САПР можно определить как комплекс научных методов исследования деятельности в области проектирования с целью разработки системы методик, математических моделей и алгоритмов, обеспечивающих наилучшее распределение функции между проектировщиком и машиной, при поиске оптимальных проектных решений.

САПР же выполняет ряд процедур, главным образом тех, что логически связаны между собой, и исполнителями для принятия конкретных проектных решений. САПР могут быть использованы в различных областях науки, техники и производства: для автоматизации проектирования отдельных деталей, предметов, узлов механизмов и машин, комплексов и агрегатов; гибких линий целых производственных предприятий и комплексов.

Согласно методологии системного проектирования, задача РИ заключается не только в определении его путей, но и в характере конкретной перестройки процессов проектирования и технологической переподготовки производства, что применимо и к САПР.

Главным результатом при разработке целевой программы РИ САПР станет формирование основ его методологии, образующей фундамент научного потенциала, обеспечивающий дальнейший успех всем САПР, что были перепроектированы.

1. *Пантелеймонов А.А.* Аспекты реинженерии приложений с графическим интерфейсом пользователя // Проблемы программирования. – 2001. – № 1–2. – С. 53–62.
2. *Тимченко А.А.* Основы системного проектирования та системного аналізу складних об'єктів. Кн. 1. Основи САПР та системного проектування. – К.: Либідь, 2000. – 272 с.
3. *Лаврищева Е.М., Грищенко В.Н.* Сборочное программирование. Основы индустрии программных продуктов. – К.: Наук. думка, 2009. – 372 с.
4. *Глушков В.М.* Основы безбумажной информатики. – М.: Наука, 1982. – 552 с.
5. *Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л.* Алгебра. Языки. Программирование. – Киев: Наук. думка, 1974. – 318 с.
6. *Глушков В.М.* Введение в АСУ. – Киев: Техніка, 1974. – 320 с.
7. *Глушков В.М., Вельбицкий И.В.* Технология программирования и проблемы ее автоматизации // УСиМ. – 1976. – № 6. – С. 75–93.
8. *Глушков В.М.* Фундаментальные исследования и технология программирования // Программирование. – 1980. – № 1. – С. 3–13.
9. *Система автоматизации производства программ (АПРОП) / В.М. Глушков, Е.М. Лаврищева, А.А. Стогний и др.* – Киев: Ин-т кибернетики АН УССР. – 1976. – 134 с.
10. *Глушков В.М.* Кибернетика, вычислительная техника, информатика. Избран. труды в трех томах. – Киев: Наук. думка, 1990. – 768 с.
11. *Ершов А.П.* Введение в теоретическое программирование. – М.: Наука, 1977. – 288 с.
12. *Ершов А.П.* Два облика программирования // Кибернетика. – 1982. – № 6. – С. 122–123.
13. *Ершов А.П.* Опыт интегрального подхода к актуальной проблематике программного обеспечения // Там же. – 1984. – № 3. – С. 11–21.
14. *Реинженерия программных систем* – <http://www.programsfactory.univ.kiev.ua/content/books/2/108>.

Поступила 21.01.2014

Тел. для справок: +38 048 45-4025, 097 950-0274 (Одесса)

E-mail: dotsent1981@mail.ru

© С.С. Великодный, 2014