

Виконано аналіз методологій та технологій проектування програмного забезпечення систем керування. Пропонується застосування компонентного підходу що до проектування програмного забезпечення систем керування. Визначаються вимоги що до CASE засобів підтримки технології створення програмного забезпечення

Ключові слова: компонента, модель поведінки, генерація коду

Выполнен анализ методологий и технологий проектирования программного обеспечения систем управления. Предлагается применение компонентного подхода к проектированию программного обеспечения систем управления. Определяются требования к CASE средствам поддержки технологии создания программного обеспечения

Ключевые слова: компонента, модель поведения, генерация кода

The analysis of methodology and technology of the system of control software were completed. The component way of developing of the software is described. The requirements to the CASE-technology for software design are defined

Key words: the component, behavior model, code generation

ГЕНЕРАЦІЯ КОМПОНЕНТ ПРОГРАММНОГО ОБЕСПЕЧЕННЯ СИСТЕМИ УПРАВЛЕНИЯ ПО ФОРМАЛЬНОМУ ОПИСАНІЮ ІХ МОДЕЛЕЙ ПОВЕДЕННЯ І ОГРАНИЧЕНІЙ

С. Л. Харченко

Старший преподаватель

Кафедра ПО ЭВМ

Харьковский национальный университет

радиоэлектроники

пр. Ленина, 14, г. Харьков, 61166

Контактный тел.: 096-79-13-700

E-mail: khsln@yandex.ru

1. Введение

В современных условиях, при разработке любого программного продукта, преследуются несколько целей – это получение качественного программного изделия за минимальный отрезок времени и при минимальных затратах ресурсов. Такие задачи, поставленные перед разработчиками, требуют глубокого анализа при выборе методологий проектирования и средств автоматизации работ.

Если принять аксиомы, что наиболее надежное программное обеспечение для автоматизированных систем управления можно получить при автоматической генерации программного кода и что большее количество ошибок в проекте относятся к ошибкам «человеческого фактора», т.е. ошибкам, которые возникают на таких этапах как создание технического задания, технического и рабочего проекта. Эти отправные точки позволяют сформулировать некоторые требования, которые необходимо учитывать при реализации проекта.

2. Постановка проблемы

Так, сокращение ошибок «человеческого фактора» возможно только при формализации работ. Реализация такой установки требует применение формального языка описания, который используется на этапах создания технического задания и технического проекта. Кроме этого, формальный язык описания должен быть входным языком для программной системы, которая преобразует конструкции формального языка описания в конструкции одного из языков программирования.

3. Анализ состояния проблемы

Если этапы жизненного цикла, укрупнено, определить как разработка технического задания, технического и рабочего проектирования, то следует определить возможность использования доступных средств автоматизации.

Выполнив анализ возможностей CASE средства фирм Computer Associates и Oracle, можно утверждать, что эти средства автоматизации ни на одном из перечисленных этапов жизненного цикла создания программного обеспечения систем управления применить невозможно. А CASE средства фирмы IBM (Rational Rose) можно использовать как дополнительные средства визуализации в процессе проектирования.

Подводя итоги анализа возможности применения доступных CASE систем на этапе формирования технического задания можно утверждать, что отсутствуют CASE системы, которые специализировались бы на решении задачи создания технического задания для программных систем управления.

4. Требования к созданию системы проектирования

Анализ состояния проблемы позволил сформулировать перечень требований к гипотетической CASE системе, которая бы обеспечила автоматизацию работ на этапе формирования технического задания на программное обеспечение системы управления:

- все задачи, цели, требования и ограничения и другие компоненты проекта записываются на формальном языке в виде графического изображения и его текстового описания;
- возможность проектирования программной системы по ниспадающей методологии и декомпозиции слоя проектирования на компоненты;
- возможность декомпозиции основных целей, задач и поведения системы на цели задачи и поведение подсистем и компонентов;
- возможность декомпозиции основных требований и ограничений к системе на требования и ограничения для подсистем и компонентов;
- текстографическая запись формального языка должна быть пригодна для хранения в базе данных проекта и достаточна, чтобы по формальной записи в базе данных построить текстографическую форму на экране дисплея;
- на каждом слое декомпозиции выполняется соответствующее уточняющее описание всех элементов слоя декомпозиции проекта;
- «сборка» системы из подсистем и компонентов выполняется путем подстановки элементов описания нижележащего слоя декомпозиции в вышележащий слой;
- процесс декомпозиции завершаться достижением неделимой компоненты проекта или компоненты, которая уже создана и соответствует требованиям проекта;
- слои декомпозиции реализуются параллельно независимыми группами экспертов, которые специализируются в некоторой проблемной области;
- на каждом слое декомпозиции проверяется возможность достижения поставленной цели (достижение точки выхода из компонента);
- контроль объединения компонентов проекта выполняется автоматически исходя из описания интерфейсов взаимодействия и условий применения компонентов.

5. Этапы реализации проекта

Так как этап жизненного цикла – эскизный проект, исходя из особенностей выполняемого проекта, может выполняться как на этапе технического задания, так и на этапе технического проекта и основное его назначение это проверка правильности выбранных технических решений. Это позволяет этап эскизного проектирования исключить из рассмотрения.

Следующим шагом анализа - рассмотрим проблемы, которые возникают на этапе технического проекта. Исходя из практики, можно утверждать, что техническое задание на проект занимает большой объем текстового материала и содержит в приложении алгоритмы функционирования подсистем и программ. Так как это текстовый документ, который сформирован человеком, то можно предположить, что неавторское прочтение и интерпретация такого документа, на этапе технического проекта, вызовет разночтение отдельных его положений. Это, в свою очередь, приведет к тому, что в проекте будут реализовываться ошибочные концепции. Следует учитывать и то, что проведение верификации результирующего документа полученного на этапе технического проекта весьма сложное и неформальное действие. При этом особую озабоченность вызывает правильность формирования интерфейсов подсистем и программных компонент.

Если сгруппировать статистику ошибок из различных проектов, то можно утверждать, что большинство ошибок, которые будут выявлены на последующих этапах, будут отнесены к ошибкам «человеческого фактора», которые возникли как на этапе создания технического задания, так и на этапе технического проекта.

Анализ проблемы показывает, что эти работы автоматизировать невозможно, учитывая уровень современного развития систем искусственного интеллекта, так как основная работа выполняется за счет интеллекта разработчика. Все ошибки на этих этапах относятся к ошибкам разработчика. Сокращение ошибок возможно только за счет сокращения числа неоднозначно трактуемых элементов входной информации, что предполагает наличие формальной формы описания, представленной в тексто-графическом виде, с одной стороны, и повышения квалификации специалистов участвующих в работе над проектом.

Традиционно на этапе рабочего проекта формируется код реализации системы, создаются модели поведения и системы ограничений, которые необходимы при тестировании полученного кода, проверяется достижимость поставленных целей и работоспособность созданного программного обеспечения в заданном диапазоне ограничений. Традиционно большая часть работ, на этом этапе, выполняются человеком. А качество работ зависит от квалификации персонала, а сроки исполнения от его количества.

6. Автоматизация проектирования

В тоже время, современные CASE системы предполагают возможность генерации конечного кода по формальному описанию. Так, в ERwin имеется возможность выполнить генерацию SQL кода по фор-

мальному описанию структуры базы данных. В CASE средствах корпорации Oracle возможна генерация SQL кода и кода позволяющего реализовывать разнообразные интерфейсы пользователя (JDeveloper, FORMS и REPORTS Developer).

Имеется механизм генерации кода и в CASE средствах фирмы Rational Rose.

Из этого можно сделать вывод, что механизм генерации программного кода по формальному описанию имеется, но недостаточно распространен. И в этом случае основной вопрос заключается в том - достаточна ли форма и объем формального описания технического задания для его трансформации в код и возможна ли такая трансформация формального описания без дополнительных описаний и механизмов. На эти вопросы каждый разработчик отвечает исходя из созданных условий.

7. Методология проектирования

Сегодня, в силу ряда причин, возник дефицит в специалистах, которые способны качественно выполнять работы по созданию программных систем управления. В этом случае выполнение работ по традиционной методологии проблематично или требует увеличения сроков исполнения работ, что отрицательно сказывается на конкурентоспособности. Такой вывод объясняется пересмотреть методологию проектирования в следующих аспектах:

- на этапе технического задания выполняются работы как по созданию ТЗ, так и по проектированию системы, без привязки к вычислительной и программной платформе. Результатом работы должны стать развернутые модели поведения и ограничений системы, на которых можно проверить возможность достижения поставленных целей. Данные моделей представляются в виде, пригодном для машинного хранения и обработки. Дополнительно формируется описание, что соответствует требованиям действующего стандарта на разработку системы. Вся информация должна храниться в базе данных проекта;

- завершением работ на этапе технического задания необходимо считать момент, когда все построенные модели докажут свою функциональность, т.е. возможность решения поставленных задач и достижимость конечных элементов моделей на каждом слое декомпозиции. Это будет возможно, так как имеется возможность оперативно преобразовать формальное описание в программный код, на котором будет проверена модель. Это может заменить работы по тестированию конечного программного продукта, так как все реакции на ограничения будут проверены, на соответствующих моделях;

- на начальных стадиях создания технического задания определяется, какой будет среда функционирования ПО, что позволяет на ранних этапах проектирования начать работу по созданию средств преобразования формального описания модели поведения и ограничений в программный код;

- так как система проектируется «сверху-вниз», как некоторый набор взаимодействующих компонент, то это позволяет распараллелить работу по нескольким направлениям. В конечном итоге будет произ-

ведена сборка высокоотлаженных, согласованных по интерфейсам и назначению компонент в один проект;

- наличие модели ограничений позволит использовать средства автоматической генерации тестов для системы тестирования, что обеспечит получение заключения о готовности программной системы.

8. Оценка эффективности проектирования

Теперь определим положительные моменты нововведения.

С точки зрения применения компонентной методологии проектирования и повторного использования компонент:

- компонента проекта хранится в базе данных как формальная запись поведения и ограничений;

- наличие множества претрансляторов позволяют получать эквивалентные программные модули в разных языковых средах из исходного описания компоненты;

- так как компонента хранится как набор элементов описания поведения и ограничений имеется возможность организовать хранения в базе данных вариантов исполнения компонента;

- возможно создание базы компонентов для их повторного использования в иных проектах.

С точки зрения организации работ:

- разработчики технического задания дополнительно берут на себя функцию создание моделей, описывающих поведение и ограничения в проектируемой системе, определяют правила декомпозиции, интерфейсы и другие элементы проекта;

- разработчики технического проекта выполняют работы по проектированию недостающих компонент и моделей, реализуют механизмы преобразования формальных описаний в соответствующий программный код, т.е. создают претрансляторы и генераторы тестов;

- разработчики программного кода выполняют генерацию программного кода проекта и тестовых заданий, проводят тестирование системы на работоспособность.

9. Вывод

Реализация такого подхода к проектированию программного обеспечения систем управления позволит сократить количество персонала участвующего в проекте, сократить сроки исполнения, повысится надежность программного кода, так как он получается путем генерации по формальному описанию, с соблюдением формальных правил преобразования при автоматическом контроле интерфейсов системы и правил подстановки.

Литература

1. В.В. Липаев, Проектирование математического обеспечения АСУ (системотехника, архитектура, технология) [Текст]. -М.: «Советское радио», 1977. – 400 с.
2. В.В. Липаев, Л.А. Серебровский, П.Г. Гаганов, Ф.А. Каганов, М.А. Минаев, А.А. Штрик, Технология проектирования комплексов программ АСУ [Текст]. -М.: «Радио и связь», 1983. – 264 с.

3. С.В. Маклаков, ВРwin и Erwin. CASE-средства разработки информационных систем [Текст]. – М.: «ДИАЛОГ-МИФИ», 2000. – 256с.
4. С.В. Маклаков, Создание информационных систем с ALLFusion Modeling Suite [Текст]. – 2-е изд., испр. и дополн. – М.: «ДИАЛОГ-МИФИ», 2007 – 400 с.
5. Питер Колетски, Д-р Поль Дорен Oracle Designer Настольная книга пользователя [Текст]. – 2-е изд., - М.: «ЛОРИ», 1999. – 592 с.
6. Нирва Мориссо-Леруа, Мартин К. Соломон, Джеральд Момплезир, Oracle 9i Программирование на языке SQLJ [Текст]. - М.: «ЛОРИ», 2003. – 592 с.
7. Майкл Армстронг-Смит, Дарлен Армстронг-Смит, Oracle Discoverer Разработка специальных запросов и анализ данных [Текст]. - М.: «ЛОРИ», 2002. – 483 с.
8. Грейди Буч, Джеймс Рамбо, Айвар Джекобсон Язык UML Руководство пользователя [Текст]. -М. Издательство ДМК, 2001. – 429с.
9. Дэвид А. Марка и Клемент МакГоуэн "Методология структурного анализа и проектирования" [Электронный ресурс]. Электронная книга опубликована на <http://lib.perm.ru>.
- 10 Орлов С.А. Технология разработки программного обеспечения [Текст], – СПб.: Питер, 2002. – 464 с.

УДК 656.212.5

Розглянуті існуючі методики розрахунку числа колій. Надано аналіз колійного розвитку парку приймання Південної системи сортувальної станції Основа. Розрахована економічна ефективність від оптимізації колійного розвитку парку приймання

Ключові слова: методики розрахунку числа колій, колійний розвиток, експлуатаційні витрати

Рассмотрены существующие методики расчета числа путей. Дан анализ путевого развития парка приема Южной системы сортировочной станции Основа. Рассчитана экономическая эффективность от оптимизации путевого развития парка приема

Ключевые слова: методики расчета числа путей, путевое развитие, эксплуатационные расходы

They are considered existing methods of the calculation of the number of the ways. It is given analysis of the travel development parka receiving the south system to marshalling yard Osnova. The calculated cost-performance from optimization of the travel development parka acceptance

Key words: methods of the calculation of the number of the ways, travel development, working expenses

ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ ОТ ОПТИМИЗАЦИИ ПУТЕВОГО РАЗВИТИЯ СОРТИРОВОЧНОЙ СТАНЦИИ

К. В. Таратушка

Ассистент

Кафедра "Железнодорожные станции и узлы"

Украинская государственная академия

железнодорожного транспорта

пл. Фейербаха, 7, г. Харьков, Украина

Контактный тел.: 372-21-72, 063-419-00-09

E-mail: kostiktar@rambler.ru

Введение

Приведение основных фондов, трудовых и материальных ресурсов в соответствие с объемами работы, и к существующим потребностями экономики и на-

селения в перевозках - одна из основных задач 1 этапа Программы реструктуризации железнодорожного транспорта Украины [13]. Общее падение объемов перевозок привело к перераспределению их между железнодорожным и автомобильным видами транс-