

РАЗРАБОТКА КУРСА «УНИФИЦИРОВАННЫЙ ЯЗЫК МОДЕЛИРОВАНИЯ UML» В УЧЕБНО-МЕТОДИЧЕСКОМ КОМПЛЕКСЕ «ПРОГРАММИРОВАНИЕ»

А. П. Побегайло

Белорусский государственный университет, кафедра технологии программирования
пр. Независимости, 4, г. Минск, Беларусь
телефон: + (172) 209-52-73; e-mail: pobegailo@bsu.by
web: www.bsu.by

В докладе предлагается подход к изложению материала по курсу «Унифицированный язык моделирования UML». Изложенный подход используется при чтении специального курса «UML и шаблоны проектирования» на факультете прикладной математики Белорусского государственного университета для студентов кафедры технологии программирования, специальности прикладная математика. Материалы по разработанному курсу представлены в Учебно-методическом комплексе «Программирование», разработанном на факультете прикладной математики Белорусского государственного университета.

Ключевые слова – разработка программного обеспечения, моделирование программных систем, шаблоны проектирования, унифицированный язык моделирования UML.

1 ПОСТАНОВКА ЗАДАЧИ

В настоящее время общепризнано, что разработка программного обеспечения переходит на промышленную основу. Поэтому практически во всех ведущих университетах мира студентам, специализирующимся в области информационных технологий, читаются как общие, так и специальные курсы по разработке программного обеспечения. Как правило, общие курсы называются «Software Engineering», что можно перевести как «Разработка программного обеспечения». В свою очередь названия специальных курсов обычно более специализированы, например, «Моделирование программных систем», «UML и шаблоны проектирования» или «Разработка требований к программным системам». Общим для всех этих курсов является изучения языка моделирования систем UML, как основного средства моделирования и документирования программных систем.

Из вышесказанного следует, что разработка учебных и методических материалов для поддержки курсов по разработке программного обеспечения является актуальной задачей. В данном докладе рассмотрен подход к разработке материалов по языку моделирования систем UML, которые могли бы использоваться как в общих, так и специальных курсах по разработке программного обеспечения. Предлагаемый подход к изучению UML ориентирован на начальное ознакомление с предметом. Есте-

ственно, что студенты не разрабатывали программные системы. Поэтому представленный курс предполагает ознакомление студентов с основными концепциями языка моделирования систем UML и разработку простой программной системы, которая включает как модель системы, так и её программную реализацию.

2 ОБЗОР ПРОГРАММ СУЩЕСТВУЮЩИХ КУРСОВ

Рассмотрим несколько программ по курсам разработки программного обеспечения, которые приняты в американских и европейских университетах. Но прежде скажем, что важность таких курсов стала понятна при успехе у разработчиков программного обеспечения фирмы Rational Rose, которая предложила разработчикам мощный инструмент для моделирования программных систем, на основе языка моделирования систем UML. В 2001 году в журнале Computer даже была опубликована статья, посвященная проблемам обучения разработке программного обеспечения в академических структурах [1]. Для примера рассмотрим программы курсов по разработке программного обеспечения двух университетов США и одного европейского университета, которые занимают верхние места в мировом рейтинге университетов.

Американский университет University of California Berkley предлагает своим студентам курс «CS 169. Software Engineering». Вот темы, которые рассматриваются в данном курсе:

- Software engineering processes.
- Object-oriented processed.
- Agile processes (Extreme programming).
- Gathering requirements.
- Specifications.
- Unified Modeling Language.
- Design.
- Unified Modeling Language.
- Design patterns.
- Version control.
- Testing.
- Regression testing.
- Integration testing.
- Test generation.
- Debugging.

General principles.
Delta debugging.
Runtime monitoring.
Purify.
Cooperative Bug Isolation.
Deduce.
Eraser (data race debugging).
Software security.
Verification.
Refactoring.
Software reuse.
Programming with Exceptions.

Другой американский университет Cornell University также предлагает своим студентам курс «CS 501. Software Engineering». Вот краткая программа данного курса:

Introduction to Software Engineering.
Software Processes.
Source Code Management.
Project Management.
Legal Aspects of Software Engineering.
Requirements.
Usability.
System Architecture and Design.
Object Oriented Design.
Reliability.
Delivering the System.
Performance of Computer Systems.
Risk in Software Development.

В Европе в качестве примера можно привести курс по разработке программного обеспечения «Software Engineering with Objects and Components», который предлагается студентам университета University of Edinburgh в Великобритании. Вот перечень названий лекций, читаемых в данном курсе:

Intro to Software Engineering.
Requirements Engineering.
Use Cases.
Software Design.
Class Diagrams.
CRC Cards.
Project Management.
Package Diagrams.
Composite Structures.
Component Diagrams.
Deployment Diagrams.
Sequence Diagrams.
Communication Diagrams.
Activity Diagrams.
Statechart Diagrams.
Software Construction.
Software Testing.
Software Maintenance and Evolution.
Reuse and Components.
Software Quality.
Engineering High-Dependability Systems.

В представленных программах виден общий подход, принятый в университетах США и Европы к преподаванию дисциплины «Разработка программного обеспечения». Так как программы ведущих американских и евро-

пейских университетов являются примером для университетов всего мира, то можно сказать, что данный подход к преподаванию рассматриваемого курса используется во всех университетах мира.

Из анализа представленных программ общих курсов по разработке программного обеспечения можно сделать следующие выводы:

- курсы по разработке программного обеспечения охватывают полный спектр вопросов, связанных с разработкой программного обеспечения;
- так как охват вопросов очень широк, то, по-видимому, в данных курсах многие вопросы рассматриваются на ознакомительном уровне, а их детальное изучение предполагается в других специальных курсах;
- для моделирования программных систем используется язык UML, причем в некоторых курсах он изучается довольно подробно;
- очень большое внимание уделяется разработке требований к программным системам.

Теперь можно провести краткий анализ специальных курсов, касающихся тематики разработки программного обеспечения. Все эти специальные курсы можно разбить на следующие три категории:

- специальные курсы, посвященные формальным средствам спецификации программных систем;
- специальные курсы, посвященные формальным средствам описания поведения или работы программных систем;
- специальные курсы, посвященные тестированию и оценке качества, разработанных программных систем.

Язык моделирования систем UML является графическим языком, позволяющим разработчику построить графические диаграммы, которые моделируют как структуру, так и поведение программной системы. Поэтому часто специальные курсы, включают углубленное изучение этого языка. Здесь следует сказать, что на содержание специальных курсов по разработке программного обеспечения сильное влияние оказали книги [2, 3]. Поэтому сравнительно недавно многие специальные курсы, касающиеся разработки программного обеспечения назывались «UML и шаблоны проектирования». Но с течением времени, этот материал перешел из категории новаторских работ в категорию классических работ. Поэтому в настоящее время шаблоны проектирования, как правило, изучаются в курсах по объектно-ориентированному проектированию систем, причем классические шаблоны проектирования разработаны практически для всех языков программирования. В любом случае для спецификации шаблонов проектирования также используется язык UML.

В заключение данного раздела можно сказать, что знание языка моделирования UML является необходимо-стью для специалиста любого уровня, связанного с разработкой программного обеспечения. Это обусловлено

тем, этот язык используется для исчерпывающего моделирования программных систем, начиная от спецификации функциональных требований к программной системе, и заканчивая спецификацией интерфейсов классов и компонент программной системы, а также оборудования, на котором будет работать эта система. Основным пособием по изучению языка моделирования UML следует считать книгу [4].

3 ПРЕДЛАГАЕМЫЙ ПОДХОД К ИЗЛОЖЕНИЮ МАТЕРИАЛА

Рассмотрим подход к изучению языка моделирования систем UML, предложенный в Учебно-методическом комплексе «Программирование», разработанном в Белорусском государственном университете на факультете прикладной математики. Прежде чем излагать основные концептуальные методические идеи, используемые при изложении данного курса, приведем программу этого курса:

Программные системы.

Понятие программной системы.

Модель программной системы.

Структурная организация и динамическое поведение программной системы.

Архитектура программной системы.

Этапы разработки программных систем.

Анализ решаемой задачи: изучение предметной области и определение функциональных требований к программной системе.

Проектирование программной системы: модель системы, документация на систему.

Реализация программной системы: программирование и отладка.

Внедрение программной системы: опытная эксплуатация системы и исправление ошибок.

UML унифицированный язык моделирования систем.

Назначение языка моделирования систем UML.

Виды или представления модели программной системы.

Диаграммы, используемые для моделирования структуры системы.

Диаграммы, используемые для моделирования динамического поведения программной системы.

Моделирование функциональных требований к системе.

Моделирование функциональных требований к программной системе.

Диаграммы вариантов использования.

Оформление документации на диаграммы вариантов использования.

Глоссарий.

Моделирование поведения системы на функциональном уровне.

Диаграммы активности.

Диаграммы последовательности.

Диаграммы состояний.

Моделирование логической структуры программной системы.

Диаграммы классов.

Классы и их спецификация.

Атрибуты класса и их спецификация.

Операции класса и их спецификация.

Отношения между классами и их спецификация.

Моделирование поведения системы на логическом уровне.

Диаграммы активности.

Диаграммы последовательности.

Диаграммы состояний.

Диаграммы взаимодействия.

Моделирование физической организации системы.

Диаграммы компонент.

Диаграммы развертывания.

Модель физической организации программной системы.

Процесс разработки программной системы.

Последовательный и итеративный процессы разработки.

Фазы итеративного процесса разработки.

Классическое изложение материала по изучению языка моделирования UML обычно включает более или менее подробное изучение графических элементов этого языка и диаграмм, которые используются при моделировании программной системы. Опыт преподавания данного курса показал, что, как правило, изучение элементов языка и диаграмм не представляет для студентов значительных трудностей. Однако возникают другие проблемы, которые связанные с использованием изученных средств при моделировании программной системы. Кратко перечислим основные проблемы, связанные с применением студентами языка моделирования UML для моделирования программных систем, а также способы их решения, предложенные в данном курсе.

1. Студенты легко понимают, что при изучении или моделировании любой системы нужно рассматривать структурные и динамические аспекты организации этой системы. Но это легко понять, но тяжело применить на практике. Очень часто структурные и динамические отношения между элементами системы смешиваются. В предложенном курсе на начальном этапе изложения делается упор на разделение структурной организации системы и её динамического поведения. Все понятия, встречающиеся при моделировании программной системы, рассматриваются именно с этих двух точек зрения. Это не совсем верно, так как между структурой системы и её поведением существует взаимосвязь, но это полезно с методической точки зрения. Здесь для программистов хорошо рассмотреть пример классов и подробно разобрать структур объектов класса и их поведение.

2. При моделировании функциональных требований к программной системе студентами трудно усваивается смысл вариантов использования и диаграмм вариантов использования. Очень тяжело добиться понимания того, что вариант использования – это функциональное требование к программной системе, а не блок какого-то алгоритма взаимодействия пользователя с системой. Это не

удивительно, так как в наших университетах студентам, специализирующимся в области информационных технологий, как правило, читаются курсы по программированию, теории алгоритмов и структур данных и т. п., в которых основное внимание уделяется алгоритмическим аспектам программирования. В этом то и состоит проблема. Студенты не понимают, что прежде чем писать программу, должен быть разработан точный интерфейс этой программы. А курсов по формальной спецификации интерфейсов и программ студенты не слушали. Эту проблему решить без специальных курсов по спецификации программных систем очень сложно. Здесь приходится только подробно объяснять материал и исправлять ошибки, возникающие при моделировании программной системы.

3. При моделировании поведения программной системы студенты не понимают, как правильно использовать диаграммы поведения. Возникают вопросы о том, поведение чего нужно моделировать и какую диаграмму поведения для этого выбрать. Это опять же следствие недостатка материалов по формальным моделям вычислений. Как может студент качественную разработать диаграмму состояний, если он даже не слышал, что такое цифровой автомат. Для решения этой проблемы на каждом структурном уровне моделирования программной системы подробно рассматриваются объекты, поведение или взаимодействие которых нужно моделировать, а также выбираются диаграммы поведения, которые наилучшим образом подходят для этого моделирования.

4 ЗАКЛЮЧЕНИЕ

В представленном докладе предложен подход изложению материала курса по унифицированному языку программирования UML, который использован в Учебно-методическом комплексе «Программирование», разработанном в Белорусском государственном университете. Рассмотрены основные проблемы, которые встречаются у студентов при изучении данного курса и предложены пути их решения.

ЛИТЕРАТУРА

- [1] Software Engineering in the Academy / В. Meyer // Computer - 2001. - May. - P. 28-35.
- [2] Объектно-ориентированный анализ и проектирование. 2-е изд.: Пер. с англ. / Буч Г. СПб.: Бинум, 2000. 558 с.
- [3] Приемы объектно-ориентированного проектирования. Паттерны проектирования. Пер. с англ. / Гамма Е., Хелм Р., Джонсон Р., Влиссидес Дж. СПб.: Питер 2004. 366 с.
- [4] Язык UML. Руководство пользователя. 2-е изд.: Пер с англ. / Буч Г., Рамбо Дж., Якобсон И. М.: ДМК 2007. 496 с.