МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Томский государственный университет
Горно-Алтайский государственный университет
Институт оптики атмосферы им. В.Е. Зуева СО РАН

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ИССЛЕДОВАНИИ СЛОЖНЫХ СТРУКТУР

МАТЕРИАЛЫ ДЕСЯТОЙ РОССИЙСКОЙ КОНФЕРЕНЦИИ С МЕЖДУНАРОДНЫМ УЧАСТИЕМ

Томск
Издательский Дом Томского государственного университета
2014

можно не использовать вовсе. Привычные понятия ER-модели обеспечивают более понятные человеку формы восприятия данных.

6. Язык манипулирования данными может использоваться для определения бизнес-правил. С помощью операций над отображениями определяются так называемые получаемые (или производные) отображения, экстенсионалы которых в отличие от хранимых отображений вычисляются системой. Их характеристики являются дополнительными средствами выражения семантики ПрО.

7. Схема данных может содержать избыточные описания ПрО за счет:

– синонимии (использования в схеме разных форм представления для одних и тех же явлений ПрО и законов их взаимодействия);

– следствий и эквивалентностей отображений.

8. Имеется возможность определять в схеме хранимые и получаемые данные, последние непосредственно не хранятся на диске, а вычисляются с помощью тех или иных правил, определенных в схеме.

С использованием ERM-модели семантическая методика проектирования схем БД может быть доведения до идеала – исчерпывающая требования к БД формализация описаний ПрО в рамках семантической модели и трансляция этих описаний в полную и эффективную СУБД-ориентированную схему данных, не требующую дополнительных расширений.

Литература

1. *Chen P.P-S.* The Entity-Relationship Model – Toward a Unified View of Data // ACM Transactions on Database Systems. Vol. 1, No. 1. March 1976. pp. 9–36.
2. *Bennett S., McRobb S. and Farmer R.* Object-Oriented Systems Analysis and Design, Fourth Edition. McGraw-Hill Higher Education. 2010, 688 p.
3. *Halpin, T. and Morgan, T.* Information Modeling and Relational Databases, Second Edition. Morgan Kaufman, 2008, 943 p.
4. *Бабанов А.М.* Семантическая модель «Сущность – Связь – Отображение» // Вестник ТГУ. УВТиИ. 2007. № 1. С. 77–91.

К РАЗРАБОТКЕ ПРОГРАММЫ КОДОГЕНЕРАЦИИ «ИСКУССТВЕННЫЙ ПРОГРАММИСТ»

С.В. Батрацкий, С.А. Прокопенко

Национальный исследовательский Томский государственный университет, Томск, Россия
twilight7775@gmail.com, s.prokopenko@sibmail.com

В данном докладе предлагается подход к разработке программы автоматической кодогенерации по блок-схеме алгоритма на основе булевых матриц. На первом шаге мы рассматриваем только такие блок-схемы алгоритмов, которые удовлетворяют определенным условиям. Как обычно, овальные блоки «Начало» и «Конец» соответствуют началу и концу программы. Соответственно блок «Начало» не имеет входящих дуг и имеет одну исходящую дугу; блок «Конец» не имеет исходящих дуг. Блоки-прямоугольники соответствуют выполнению некоторой (одной) вычислительной операции, и так же, как и блок ввода-вывода данных (параллелограмм), имеют не менее одной входящей дуги и только одну исходящую дугу. Ромбовидный логический блок описывает условия, требуемые для выполнения последующих операций, и имеет не менее одной входящей дуги и только две исходящие дуги. Таким образом, в овальные блоки начала и конца можно занести только значения «начало» и «конец» соответственно, в блоке вычислений могут быть только действия, например, $i = i + j$; в ромбовидном блоке – условие, например, $i = j$; в блоке ввода-вывода данных – переменная, значение которой необходимо ввести с клавиатуры или вывести на экран компьютера.

Для разработки программы автоматической генерации программы по блок-схеме алгоритма последнюю можно рассматривать как ориентированный граф [1, 2], вершинам которого поставлены в соответствие блоки блок-схемы, а ребрам – дуги, ассоциированные с блоками блок-схемы. Соответственно компьютерное представление блок-схемы алгоритма содержит две матрицы: матрицу смежности вершин и матрицу инцидентности графа, и одномерный массив, в котором хранится соответствие между вершинами графа и блоками блок-схемы с выполняемыми в них действиями. Номер i элемента массива соответствует i-й вершине графа, то есть определённому блоку блок-схемы. В i-м элементе масси-

ва находится содержимое соответствующего блока блок-схемы. Матрица смежности вершин представляется двумерным массивом, строкам и столбцам которого соответствуют вершины графа, элемент [*i*, *j*] массива равен 1, если *i*-я и *j*-я вершины графа являются смежными, в противном случае элемент [*i*, *j*] равен нулю.

Код программы, построенный автоматически по блок-схеме, хорошо поддается верификационному тестированию [3 – 5]. По блок-схеме строятся тесты (в соответствии с требуемой полнотой), которые далее подаются на разработанную программу. Более того, в программу можно автоматически вносить ошибки (например, с использованием некоторого генератора для мутационного тестирования [6 – 8]). Если построенным тестом ошибка не обнаруживается, то тест для обнаружения такой ошибки можно достроить по соответствующей блок-схеме, что может оказаться более эффективным, чем построение соответствующей различающей последовательности по программной реализации. Синтезированные тестовые последовательности можно, в частности, использовать при переносе сгенерированной программы на другие платформы и языки программирования.

Литература

1. *Кристофидес Н.* Теория графов: алгоритмический подход / пер. с англ. Э.В. Вершкова, И.В. Коновальцева ; под ред. Г.П. Гаврилова. М. : Мир, 1978. 432 с.
2. *Оре О.* Теория графов / пер. с англ. И.Н. Врублевской ; под ред. Н.Н. Воробьева, 2-е изд., стереотип. М. : Наука, 1980. 336 с.
3. *Майерс Г., Баджетт Т., Сандлер К.* Искусство тестирования программ. 3-е изд. М. : Диалектика, 2012. 272 с.
4. *Калбертсон Р., Браун К., Кобб Г.* Быстрое тестирование. М. : Вильямс, 2002. 374 с.
5. *Синицын С.В., Налютин Н.Ю.* Верификация программного обеспечения. М. : БИНОМ, 2008. 368 с.
6. *Ma Y.-S., Offutt J., Kwon Y.R.* MuJava: An Automated Class Mutation System. URL: http://www.cs.gmu.edu/~offutt/rsrch/papers/mujava.pdf.
7. *Offutt A.J.* A Practical System for Mutation Testing: Help for the Common Programmer. URL: http://cs.gmu.edu/~offutt/rsrch/papers/practical.pdf.
8. *Информация* о продукте μjava // μJava Home Page. URL: http://cs.gmu.edu/~offutt/mujava/

ПРИМЕНЕНИЕ ПРИНЦИПА ДИХОТОМИИ ОБЪЕКТНОГО ПОДХОДА В ПРОЦЕССЕ РАЗРАБОТКИ ПО

О.А. Змеев, Л.С. Иванова

Национальный исследовательский Томский государственный университет, Томск, Россия
ozmeyev@gmail.com, lida@redlg.ru

Для множества понятийных сущностей объектно-ориентированного подхода может быть применена дихотомия, разделяющая всё множество на два подмножества: сущности-классы и сущности-объекты. В 1998 году исследователи из Берлинского технического университета в своей статье [1] обозначили первое подмножество интенсиональными сущностями, второе – экстенсиональными сущностями. Структура экстенсиональной сущности определяется соответствующей интенсиональной сущностью. В качестве примера можно привести пары класс-объект, ассоциация-ссылка, вариант использования–сценарий и т.п. Кроме того, данное свойство сущностей ООП рассмотрено авторами книги «Язык UML. Руководство пользователя»[2].

Перечисленные авторы в качестве сущностей для дихотомии рассматривали понятия, относящиеся к области разработки программного обеспечения. Такая важная область ООП, как управление проектами, совершенно не затронута. Определение интенсионального и экстенсионального подмножеств в управлении проектами важно как с точки зрения теории ООП, так и с точки зрения практики. Выявленные соответствия можно использовать для автоматической генерации динамических объектов жизненного цикла ПО на основе статических интенсиональных сущностей.

В качестве интенсиональных можно использовать сущности, представленные в стандарте Software & Systems Process Engineering Meta-Model (SPEM) [3]. Данный стандарт содержит метамодель для описания процессов разработки. На основе этой модели можно выделить следующий набор интенсиональных сущностей: роль, задача, артефакт, руководство, описание процесса и другие. Данные сущности содержат лишь описания и не являются объектами жизненного цикла ПО. Перечисленным сущ-