
УДК 681.3.06

И. В. Карпусь

Волынский государственный университет им. Леси Украинки
(Украина, 43000, Луцк, пр-т Воли, 13,
тел.: (03322) 51720; E-mail: igor_karpus@ukr.net)

Особенности построения модели предметной области проблемно-ориентированных программных систем

(Статью представил д-р техн. наук В.Д. Самойлов)

Рассмотрены основные способы практического моделирования предметной области при построении прикладных программных систем. Для повышения качества моделирования предложен подход, состоящий в связывании характеристик объектов заданной предметной области с программными компонентами и модулями программной системы.

Розглянуто основні способи практичного моделювання проблемної галузі при побудові прикладних програмних систем. Для поліпшення якості моделювання запропоновано підхід, що полягає в зв'язуванні характеристик об'єктів заданої предметної області з програмними компонентами та модулями програмної системи.

К л ю ч е в ы е с л о в а: проблемная область, программный компонент, характеристика, модель характеристик, вариантная технология.

Проблемно-ориентированные программы представляют собой программные средства многократного применения, которые отличаются от конкретных прикладных программ большей изменчивостью [1]. Основной методикой выявления и фиксации изменчивости является моделирование характеристик. Характеристика — это логическая единица поведения, определяемая множеством функциональных и нефункциональных требований [2]. Одна характеристика соответствует нескольким требованиям, покрывающим определенный сегмент предметной области. Характеристики могут определять одно требование или логическую комбинацию требований. Как правило, характеристики взаимосвязаны, например, наличие одной из них может обуславливать наличие или отсутствие другой [1]. Выявление релевантных характеристик — одна из наиболее важных задач в процессе анализа предметной области, для решения которой используется моделирование.

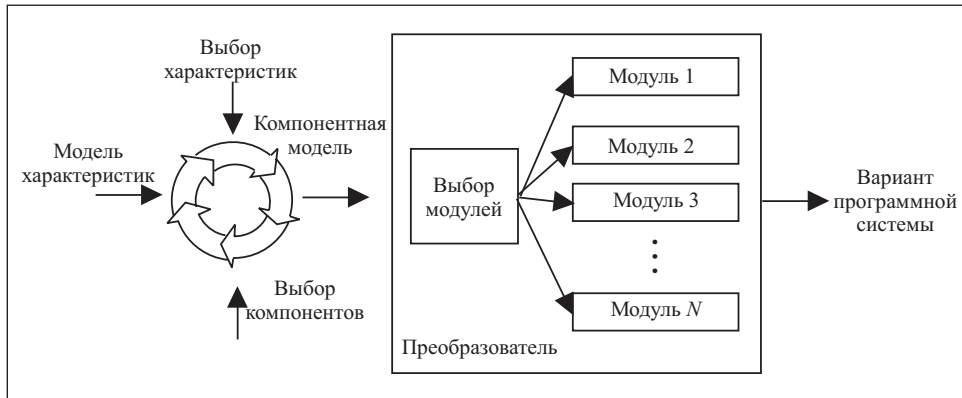


Рис. 1. Вариантная технология синтеза прикладных программных систем

Моделирование характеристик представляет собой моделирование общих и переменных свойств понятий и их взаимозависимостей, а также их упорядочивание в связную характеристическую модель, которая определяет общие и изменяемые характеристики понятий и выражает зависимости между изменяемыми характеристиками. Основой характеристической модели является диаграмма характеристик [3]. Характеристические модели, создаваемые в процессе моделирования характеристик, обеспечивают абстрактное, компактное и точное описание изменений требований и состава программных средств.

Постановка задачи. Предлагаемый подход основан на моделировании характеристик объектов проблемной области с последующей их привязкой к объектам пространства решений — программным компонентам (модулям). Такая комбинированная технология позволяет разрабатывать проблемно-ориентированные системы на основе трех моделей:

- 1) модель характеристик, описывающая данную проблемную область;
- 2) модель компонентов, описывающая пространство решений поставленной задачи в этой области;
- 3) модель вариантов, выделяющая конкретное решение из множества возможных (рис. 1).

Синтез программы осуществляется путем выбора программных модулей в соответствии с заданными функциональными требованиями (характеристиками).

Модель характеристик программной системы описывает функциональные требования, а также их возможные сочетания: обязательные, альтернативные и выборочные характеристики.

Модель компонентов программной системы описывает способы сочетания программных модулей в этой системе. Компоненты могут иметь вид классов, объектов, функций, переменных.

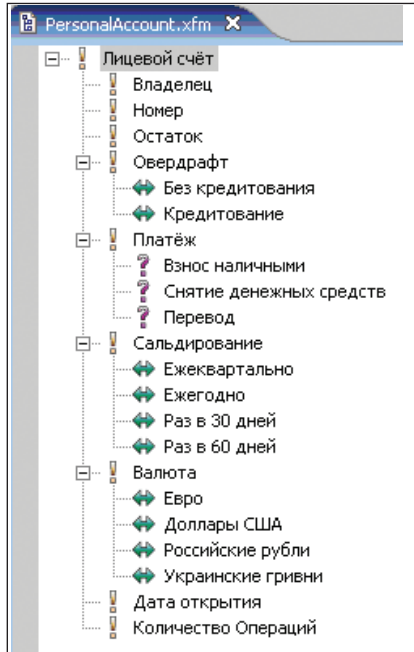


Рис. 2. Модель характеристик лицевого счета

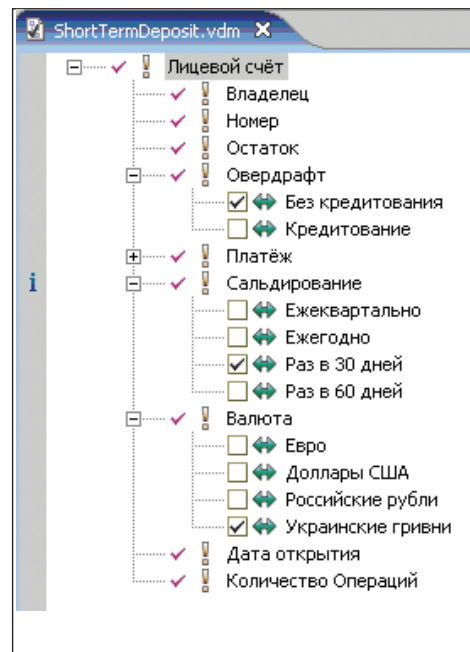


Рис. 3. Депозит до востребования

Модель представления вариантов описывает конкретный вариант программной системы. В ее основе лежит модель характеристик, из которой отбираются необходимые характеристики, соответствующие заданным требованиям.

Рассмотрим моделирование характеристик предметной области на примере лицевого банковского счета. Для упрощения примера опустим некоторые важные изменяемые параметры: расчет начисленных процентов, срок действия контракта, условия прекращения его действия, несгораемый остаток и др. Кроме того, реальная предметная область банковского счета предполагает включение ряда вспомогательных предметных областей, в числе которых валюта (символы, формат, конверсия и др.), методы платежей, аудит и другие. Модель характеристик упрощенно представлена на рис. 2, где обозначено: ! — обязательные характеристики; ? — выборочные характеристики; ⇔ — альтернативные характеристики.

Различные типы банковских счетов предполагают различные сочетания таких характеристик. Например, депозит до востребования предполагает такую комбинацию характеристик (рис. 3): ежемесячное сальдирование (баланс каждые 30 дней) и отсутствие кредитования в случае овердрафта.

Открытый счет предполагает возможность кредитования в случаях овердрафта (рис. 4), а также ежеквартальное сальдирование.

Сальдирование текущих счетов осуществляется ежеквартально, а сберегательного счета — один раз в год. Представление всех возможных моделей вариантов в табличном виде позволяет обнаружить их общность и различия (см. таблицу, где выбранные характеристики обозначены ✓, ☑).

Модель компонентов включает два основных компонента: файлы для построения данной прикладной программной системы и файлы, содержащие исходный код различных частей этой системы (рис. 5). Как видно

Варианты лицевых банковских счетов

Элемент модели	Обозначение	Уровень	Текущий счет	Срочный депозит	Открытый счет	Сберегательный счет	Депозит до востребования
Лицевой счет	!		✓	✓	✓	✓	✓
Владелец	!	1	✓	✓	✓	✓	✓
Номер	!	2	✓	✓	✓	✓	✓
Остаток	!	3	✓	✓	✓	✓	✓
Овердрафт	!	4	✓	✓	✓	✓	✓
Без кредитования	⇔	4.1	☑	☑	☐	☑	☑
Кредитование	⇔	4.2	☐	☐	☑	☐	☐
Платеж	!	5	✓	✓	✓	✓	✓
Взнос наличными	?	5.1	☐	☐	☐	☐	☐
Снятие денежных средств	?	5.2	☐	☐	☐	☐	☐
Перевод	?	5.3	☐	☐	☐	☐	☐
Сальдирование	!	6	✓	✓	✓	✓	✓
Ежеквартально	⇔	6.1	☑	☐	☑	☐	☐
Ежегодно	⇔	6.2	☐	☐	☐	☑	☐
Раз в 30 дней	⇔	6.3	☐	☑	☐	☐	☑
Раз в 60 дней	⇔	6.4	☐	☐	☐	☐	☐
Валюта	!	7	✓	✓	✓	✓	✓
Евро	⇔	7.1	☐	☐	☐	☐	☐
Доллары США	⇔	7.2	☐	☐	☐	☐	☐
Российские рубли	⇔	7.3	☐	☐	☐	☐	☐
Украинские гривни	⇔	7.4	☑	☑	☑	☑	☑
Дата открытия	!	8	✓	✓	✓	✓	✓
Количество операций	!	9	✓	✓	✓	✓	✓

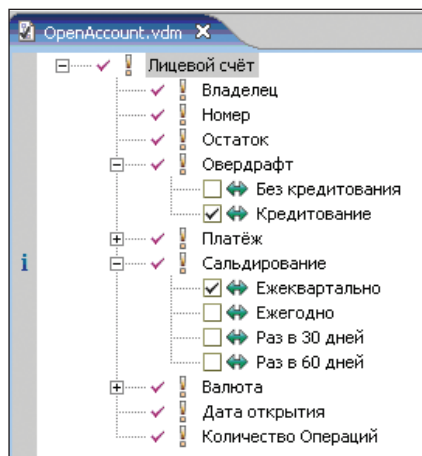


Рис. 4. Вариант открытого счета

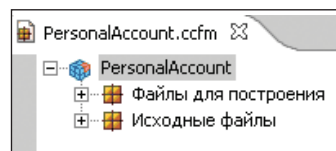
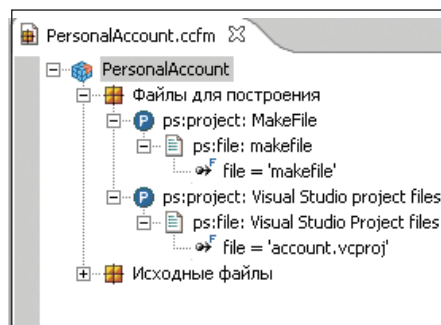


Рис. 5. Модель компонентов лицевого счета

Рис. 6. Компонент построения



из рис. 5, область решений представляет собой модульную структуру, состоящую из готовых файлов с кодом составных частей, которые используются для построения программной системы.

Рассмотрим подробнее построение модели компонентов. Первый компонент этой модели — «Файлы для построения» — содержит два файла: один из них предназначен для пакетного построения (makefile), а другой — для построения средствами Microsoft® Visual Studio® (рис. 6).

Второй компонент модели — «Исходные файлы» — представляет собой совокупность файлов реализации: функциональная часть лицевого счета собирается из готовых файлов, реализующих отдельные характеристики (рис. 7). Как видно из рис. 7, каждая характеристика реализована в отдельном файле (например, «UHR.h», «Credit.h»), который включается в проект только в случае выбора данной характеристики («hasFeature('UHR')», «hasFeature('Credit')»). После проверки допустимости моделей вариантов лицевых счетов выполняется генерирование лицевого счета с заданными характеристиками.

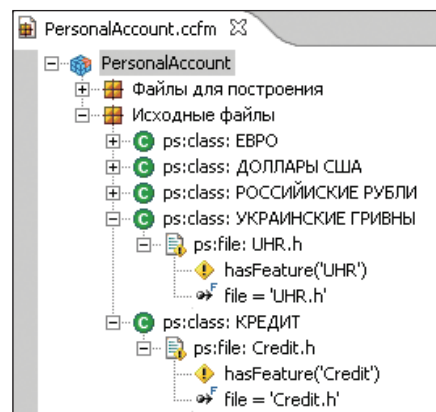


Рис. 7. Связывание программных компонентов с характеристиками предметной области

В отличие от других методов моделирования проблемных областей, предложенная технология обеспечивает разделение модели проблемной области на модель характеристик программной системы и модель программных компонентов (модулей). Непосредственная привязка характеристик объектов предметной области к программным компонентам позволяет синтезировать наиболее эффективный программный код, что дает возможность сократить время разработки проблемно-ориентированной программной системы.

Main methods are considered for the practical simulation of subject region by constructing applied program systems..For improving the simulation, an approach is proposed when the characteristics of the objects of the given subject field are communicated with program components and modules of the program system implementation field .

1. *Clarke S.* Composition of Object-Oriented Software Design Models// PhD. Thesis , School of Computer Applications. — Dublin: Dublin City University, 2001. — 273 p.
2. *Greenfield J., Short K.* Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools. — Indiana: Wiley Publishing, Inc., 2004. — 666 p.
3. *Чарнецки К., Айзенекер У.* Порождающее программирование: методы, инструменты, применение. Для профессионалов. — СПб. : Питер, 2005. — 731 с.

Поступила 04.07.06
после доработки 12.02.07

КАРЛУСЬ Игорь Васильевич, ст. преподаватель Волынского госуниверситета им. Л. Украинки, аспирант кафедры специализированных компьютерных систем Национального технического университета Украины «КПИ». В 1986 г. окончил Киевский политехнический ин-т. Область научных исследований — информационные технологии.