

УДК 681.3.07

О.А. Змеев, С.П. Кондратьев

РАСШИРЕНИЕ МОДЕЛИ UML 2 ДЛЯ ПОДДЕРЖКИ ОПИСАНИЙ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

В работе предлагается расширение модели UML 2, позволяющее описывать сценарии вариантов использования. Анализируются функциональные возможности редактора вариантов использования, реализованного на базе предлагаемого расширения.

Варианты использования изначально были предложены в качестве практически универсального средства для определения требований к любым программным системам. Впервые концепция вариантов использования (прецедент, use case) была представлена в [1] и подробно рассмотрена в [2]. С появлением унифицированного процесса разработки программного обеспечения роль вариантов использования существенно выросла – фактически именно они направляют весь процесс разработки ПО, обеспечивая средства трассировки между элементами процесса.

Описание вариантов использования может иметь различный формат и различную длину. Однако в своей основе вариант использования определяет контракт внешнего исполнителя с программной системой и имеет текстовую форму, несмотря на то, что они могут быть написаны в графической нотации с помощью диаграмм последовательностей, состояний или взаимодействия. К сожалению, в рамках UML (сокр. от англ. Unified Modeling Language – унифицированный язык моделирования) выразительные средства, предназначенные для моделирования поведенческой составляющей вариантов использования, практически отсутствуют. В рамках настоящей работы предлагается небольшое расширение стандартной модели UML 2, предназначенное для решения этого недостатка. Построение расширения рассматривается на примере обсуждения функциональных требований к редактору вариантов использования. Редактор вариантов использования – это инструмент, позволяющий разрабатывать поведенческую составляющую модели вариантов использования.

1. Описание модели UML 2

Модель UML 2, последняя версия которой (2.1) датируется 2007 г., содержит следующие пакеты, предназначенные для моделирования поведения [3]:

1. **CommonBehaviours**. (Поведения. Пакет отвечает за базовые определения аспектов поведения).

2. **Actions** (Действия. Пакет отвечает за описание примитивных действий).

3. **Activities** (Деятельности. Пакет отвечает за описание последовательности примитивных действий).

4. **Interactions** (Взаимодействия. Пакет предлагает модели описания взаимодействий в рамках поведений).

5. **StateMachines** (Машины состояния. Пакет описывает возможности использования машин состояния для моделирования поведений).

6. **UseCases** (Варианты использования. Пакет содержит высокоуровневые классы, необходимые для описания вариантов использования, такие как собственно варианты использования, актеры в вариантах использования).

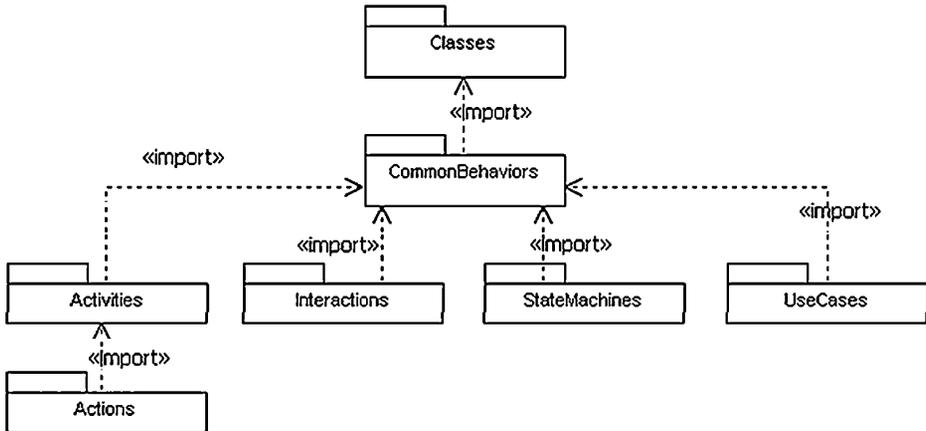


Рис.1. Диаграмма пакетов UML 2, содержащих элементы, предназначенные для моделирования поведения

Модель позволяет определять актеров и варианты использования. Она дает средства для описания включений и расширений вариантов использования. Однако она не представляет средств для моделирования сценариев вариантов использования.

Место сценариев в модели UML 2

В спецификации утверждается, что варианты использования обычно описываются в виде различных форматов, таких как естественные языки, таблицы, деревья и т.д. Поэтому не так просто точно или хотя бы в общем отобразить их структуру в виде формальной модели [3].

Тем не менее, в некоторых случаях было бы удобно иметь определенную модель для описания именно вариантов использования, причем не со структурной точки зрения, как это сделано в UML 2, а с точки зрения модели поведения, что в UML 2, к сожалению, не предлагается делать именно с вариантами использования.

Модель поведения для вариантов использования желательно иметь как на шаге анализа системы (в момент сбора требований к системе, в момент составления вариантов использования), так и на этапе программирования (для проверки того, что мы ничего не забыли), и даже на этапе тестирования (так как это позволит получать тесты в полуавтоматическом режиме).

Имеющиеся на сегодняшний день инструменты не позволяют так подходить к вариантам использования. Некоторые инструменты можно использовать для этих целей (например Rational Requisite Pro), однако это не будет удобно и не является стандартным способом его применения. В самой базовой модели не отражается факт необходимости формализации поведенческого аспекта вариантов использования и их сценарии.

2. Требования к модели сценариев

Перед созданием такого расширения необходимо уточнить, какие именно возможности должна предлагать новая модель. В настоящей работе это уточнение сделано исходя из требований к инструменту, реализующему эту модель (и позволяющему редактировать варианты использования).

Итак, основные требования к редактору вариантов использования.

Возможность создания, редактирования и удаления актеров. Редактор должен вести общий список всех актеров, позволяя тем самым в каждом новом варианте использования внутри одного проекта не создавать новых актеров. Обусловлено это, прежде всего, тем, что в любом проекте число актеров невелико. При анализе удобно просматривать список вариантов использования, которые связаны с тем или другим актером.

Поддержка нескольких проектов. Очень желательно, чтобы редактор позволял параллельно работать с несколькими проектами. Это позволяет упростить разработку родственных проектов. Это требование накладывается не столько на поведенческую модель вариантов использования, сколько на само инструментальное средство.

Поддержка экспорта описаний в текстовый файл. Поскольку основным методом описания вариантов использования в модели UML является текст, то необходимо, чтобы редактор не нарушал это требование. Все данные, введенные пользователем, могут быть выведены в виде сформатированного по правилам текста.

Поддержка сценариев и возможность их повторного использования внутри одного варианта использования. Очевидно, что редактор должен позволять тот или иной способ определения сценариев. Он должен позволять указывать для них пред- и постусловия, триггеры. Необходима возможность связывания сценариев с альтернативами.

Поддержка пошагового описания сценария. Теоретически возможно использование различных моделей для ввода описания сценариев. Это может быть просто текст (наиболее удобный человеку способ, однако наименее формализованный и, тем самым, наименее удобный для машины). Это может быть диаграмма (удобный для человека способ, удобный для машины, однако лишенный возможности неформального подхода). Поэтому предлагается модель ввода в качестве таблицы шагов. Т.е. за человеком оставляется возможность ввода текста, однако он визуально (для себя и для машины) разбивает его на блоки, указывая тем самым порядок и действующих лиц.

На средствах, предлагаемыми UML 2, это выглядит очень лаконично (рис. 2).

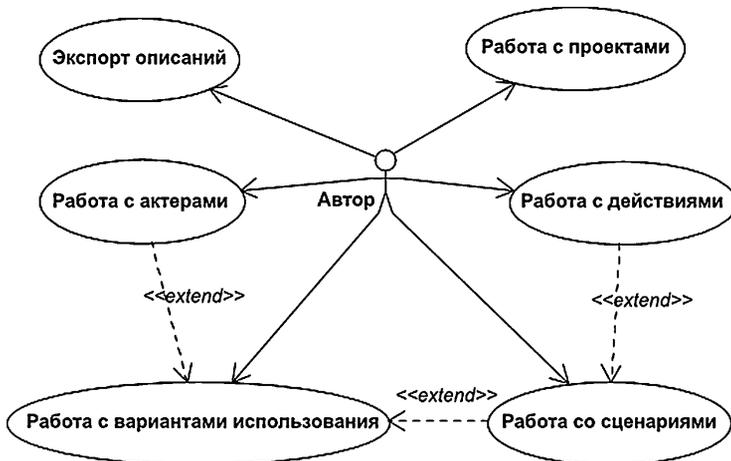


Рис. 2. Диаграмма вариантов использования редактора

Таким образом, предлагается расширить модель UML 2 так, чтобы новый пакет позволял удовлетворить эти требования.

3. Предлагаемое расширение модели

В основу расширения ложатся уже имеющиеся в UML 2 пакеты: базовый пакет **Kernel** («Ядро»), пакет **Actions** («Действия»), пакет **BasicBehaviour** («Поведения»), пакет **UseCases** («Варианты использования»).

Структура нового пакета-расширения следующая:

- Сценарий (класс унаследован от Поведения (пакет «Поведения» UML 2)).
- Альтернатива (класс унаследован от Сценария (новый пакет «Сценарии»)).
- Линия сценария (вспомогательный класс).
- Шаг сценария (унаследован от Действия, пакет «Действия» UML 2).

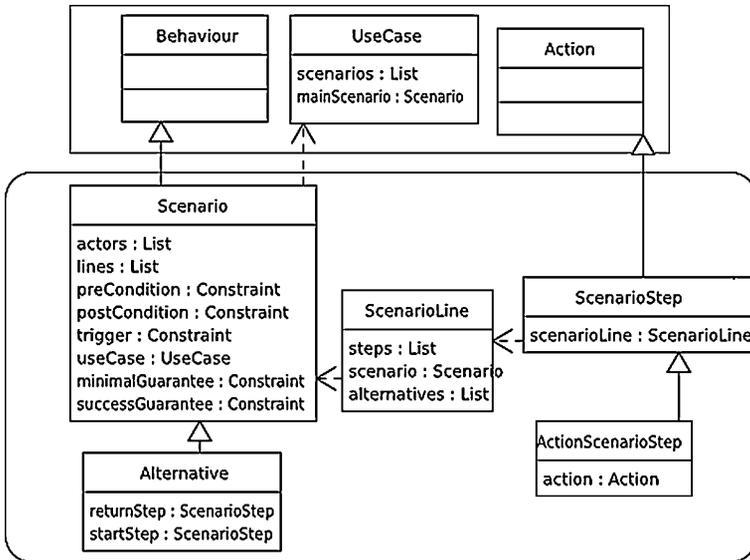


Рис. 3. Диаграмма классов предлагаемого расширения

Сценарий варианта использования хранит в себе информацию об участвующих в нем актерах. Также он хранит информацию о предусловиях, постусловиях, триггерах, минимальных гарантиях и гарантиях успеха.

Класс Вариант использования расширен несколькими свойствами (по отношению к оригинальной модели UML 2). В него добавлены свойства, на основе которых реализована связь оригинальной модели с расширением. Этот класс хранит в себе общий список сценариев (т.е. основной сценарий и все альтернативы).

Класс Линия сценария хранит в себе список возможных альтернатив и расширений на данном шаге. Конкретное описание шага хранится в следующем классе.

Внутренняя нумерация линий сценария является собственной для каждого сценария.

Класс Шаг сценария хранит в себе описание действий того актера, к которому он относится. Он содержится в Линии сценария.

В качестве проверки охвата предлагаемой модели был реализован редактор сценариев вариантов использования, в основе которого была положена эта модель.

Функциональные возможности редактора исходя из модели UML 2 без расширений

Согласно модели UML 2, вариант использования можно описать, указав его название и сделав к нему описание (т.е. привязав сущность Комментарий). Вы также можете проассоциировать актеров с данным вариантом использования и указать другие варианты использования, которые расширяют или включаются в данный вариант использования. На этом возможности стандартной модели UML 2 заканчиваются.

Функциональные возможности редактора исходя из модели UML 2 с предлагаемым расширением

Теперь для каждого варианта использования можно создавать сценарии (один, основной, будет создан самостоятельно). Сценарии можно описать по шагам. Можно создать новые альтернативные сценарии и связать их с основным. Можно задать предусловия и постусловия любого из сценария; как основного, тогда это будет автоматически считаться предусловием и постусловием для всего варианта использования, так и любой альтернативы. Такое пронизывание условиями позволит в дальнейшем реализовать систему генерации тестов на основе полностью описанного варианта использования.

Возможности экспорта из редактора

Результат разработки можно экспортировать в текстовом варианте в стандартной нотации для текстовых описаний вариантов использования [4]. В этом случае понятие Линии сценария объединяется с Шагом и будет представлять шаги в текстовом описании. Альтернативные сценарии превратятся в альтернативные потоки действий с иерархической нумерацией. Данное описание можно сохранить либо поставить как комментарий к варианту использования.

Это позволяет использовать результаты в точном соответствии с моделью, предлагаемой в UML 2: варианты использования должны иметь текстовое описание.

Таким образом, вы видим, что данное расширение модели позволяет достаточно эффективно описывать сценарии вариантов использования, что делает варианты использования более структурированными. Редактор вариантов использования, основанный на предлагаемой модели, демонстрирует её устойчивость и соответствие требованиям.

ЛИТЕРАТУРА

1. *Jacobson Ivar*. Object-oriented development in an industrial environment // Proceedings of OOPSLA'87, Special issue of SIGPLAN Notices 22 (12), December 1987. P. 183 – 191.
2. *Jacobson Ivar, Magnus Christerson, Patrik Jonsson, Gunnar Overgaard*. Object-Oriented Software Engineering; A Use-Case Driven Approach, Reading, MA: Addison-Wesley, 1992.
3. *UML 2.1.1 Superstructure Specification* [Электронный ресурс] / Object Management Group Режим доступа: <http://www.omg.org/cgi-bin/doc?formal/07-02-05>, свободный. Загл. с документа. Яз. англ.
4. *Alistair Cockburn*. Writing Effective Use Cases. Addison-Wesley Professional, 2000. 270 p.

Статья представлена кафедрой программной инженерии факультета информатики Томского государственного университета, поступила в научную редакцию 2 июля 2007.