

РАЗРАБОТКА МЕТОДИКИ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ ДИНАМИКИ СЛОЖНЫХ ПРОГРАММНЫХ СИСТЕМ

С.А. Олизаренко
(представил д.т.н., проф. Ю.Г. Даник)

Статья посвящена автоматизации процесса разработки модели переходов состояний класса в нотации UML с использованием математического аппарата конечных автоматов Мура.

В настоящее время одним из центральных звеньев всей деятельности по созданию сложной программной системы (СПС) является проектирование [1 – 3]. При этом процесс проектирования СПС определяется, прежде всего, как процесс построения и последовательного развития комплекса согласованных моделей в рамках так называемой интегрированной модели сложной программной системы (ИМ СПС).

Модели создаются, преобразовываются и контролируются с помощью соответствующих Case-средств [3, 5]. При этом процесс создания и преобразования моделей, или, точнее, соответствующих им графических диаграмм, осуществляется с помощью специальных графических редакторов (моделеров), являющихся сервисными элементами Case-средств на стадиях анализа и проектирования СПС.

Современные моделеры обеспечивают [5]:

- создание иерархически связанных диаграмм, в которых комбинируются графические и текстовые объекты;
- создание и редактирование объектов в любом месте диаграммы;
- сохранение связей между объектами при их перемещении и изменении размеров;
- автоматический контроль ошибок и др.

При этом, как правило, осуществляют следующие типы контроля: контроль синтаксиса диаграмм и типов их элементов; контроль полноты и состоятельности диаграмм; сквозной контроль диаграмм одного или различных типов на предмет их состоятельности по уровням – вертикальное и горизонтальное балансирование диаграмм.

Таким образом, процесс построения моделей с использованием Case-средств фактически сводится к реализации возможностей, обеспечиваемых их графическим редактором. При этом переход от моделей концептуального уровня представления ИМ СПС к моделям логического, а затем и физического уровня осуществляется вручную.

С целью автоматизации создания моделей в процессе поуровневого перехода от одной модели к другой и тем самым обеспечения их горизонтальной сбалансированности предлагается следующая методика автоматизации проектирования моделей переходов состояний классов.

В основе данной методики используются: модель последовательности сообщений объектов (МПСО) и модель переходов состояний класса (МПСК) в нотации универсального языка моделирования UML [1 – 3]; усовершенствованный метод синтеза конечных автоматов Мура по реализуемым ими алфавитным отображениям [4]. МПСО используется для представления временных особенностей передачи и приема сообщений между объектами ИМ СПС. МПСК обеспечивает формализованное описание поведения класса и показывает событийную связь всех его возможных состояний.

В дальнейшем будем считать, что для каждого сценария $ssu \in SSU_{su}$ (set of scripts of use case), реализующего прецедент $su \in SU$ ИМ СПС (set of use case), разработана соответствующая модель последовательностей сообщений $mm \in MM_{su}$ (model of message), т.е. задано биективное отображение

$$F_{MM_{ssu}}: MM_{su} \rightarrow SSU_{su}.$$

Понятия прецедента, сценария, объекта, класса, события и сообщения введены в [2]. В UML некоторые элементы различных моделей могут существовать параллельно и при этом быть взаимосвязанными [2]. Так получение объектом некоторого сообщения можно считать экземпляром события [1]. Таким образом, в МПСО понятия сообщения и экземпляра события в дальнейшем будем рассматривать как эквивалентные.

В результате абстрагирования объектов, участвующих во всех МПСО в виде классов ИМ СПС, а также сообщений в виде соответствующих событий, для каждого полученного класса можно получить совокупность всех последовательностей событий, у которых предназначением является только данный класс. При этом будем говорить, что данная совокупность образует множество так называемых трассировок последовательностей событий класса (ТПСК) $sc \in SC$ (set of classes) в алфавите его событий SE_{sc} (set of events), которое обозначим через STE_{sc} (set of tracing of events).

Результат выполнения классом некоторой последовательности действий [2] $sch \in SCH_{sc}$ (SCH_{sc} (set of combination of handles) совокупность всех возможных последовательностей действий в алфавите SH_{sc} (set of handles), выполняемых в ответ на поступившие ТПСК из множества STE_{sc}) можно формализовать как инъективное отображение $F_{SteSch}: STE_{sc} \rightarrow SCH_{sc}$, т.е. как некоторое отображение множества слов в конечном алфавите событий класса во множество слов в конечном алфавите действий класса. При этом собственно пару (ste, sch) будем называть каналом управления класса.

В языке UML МПСК представлена в нотации Харела [1, 2], которая объединяет в себе некоторые аспекты как автоматов Мура, так и автоматов Милли, а также концепции, относящиеся к вложенным и параллельным состояни-

ям. Однако, при использовании метода синтеза конечных автоматов на начальном этапе построения МПСК возникает необходимость приведения модели состояний к одному из видов конечных автоматов: автомату первого рода (Мили) или автомату второго рода (Мура) [4, 6].

В нашем случае предлагается использовать инициальный конечный автомат Мура, так как в данном автомате выходные сигналы (в нашей интерпретации деятельности ИМ СПС) связаны с состояниями, что соответствует представлению МПСК в UML. При этом операция преобразования МПСК в изоморфный ей конечный автомат Мура A_{sc} состоит собственно в переобозначении элементов входного алфавита, выходного алфавита и множества состояний конечного автомата в терминах МПСК.

Для полученной конечно-автоматной модели поведения класса A_{sc} характеристическая функция переходов может быть определена не только на конечном входном алфавите SE_{sc} , состоящем из всех событий класса, но и на множестве всех возможных ТПСК в алфавите SE_{sc} .

В общем случае произвольное множество слов в любом алфавите принято называть событием в этом алфавите, а сам алфавит – входным алфавитом этого события. Этот термин стал традиционным и в теории автоматов [4, 6]. Так, например, множество всех ТПСК класса STE_{sc} также можно рассматривать как некоторое событие во входном алфавите SE_{sc} (множестве событий этого класса) конечного автомата A_{sc} , где каждая трассировка последовательности $ste \in STE_{sc}$ представляет собой входное слово в алфавите SE_{sc} . В дальнейшем, для того, чтобы не путать событие как множество слов во входном алфавите и событие как абстракцию его экземпляров ИМ СПС, первое будем называть событием в произвольном входном алфавите, а второе – событием класса ИМ СПС.

В самом общем случае функционирование произвольного конечного автомата заключается в реализации некоторого отображения множества слов входного алфавита (события в данном алфавите) во множество слов выходного алфавита [4, 6].

В нашем случае, при формализации МПСК конечным автоматом Мура A_{sc} , подобное отображение реализуется следующим образом: некоторая ТПСК ste в алфавите событий класса SE_{sc} событие за событием поступает на вход конечного автомата A_{sc} , установленного предварительно в начальное состояние. Возникающая таким образом конечная последовательность событий класса на основании закона функционирования автомата однозначно вызывает появление определенной конечной последовательности деятельностей класса sch . Определяя таким образом каждой ТПСК ste соответствующую ей последовательность деятельностей sch , получаем отображение множества ТПСК в конечном алфавите событий класса SE_{sc} во множество последовательностей деятельностей класса в конечном алфавите SH_{sc} .

Построенное указанным выше способом отображение (обозначим его

через F_{SteSch}^A) представляет собой так называемое автоматное отображение, индуцированное конечным автоматом Мура A_{sc} .

Если произвольное алфавитное отображение удовлетворяет условиям автоматности отображения, то можно построить конечный автомат Мура, индуцирующий это отображение [4].

Приведем точную формулировку задачи синтеза модели переходов состояний класса ИМ СПС согласно усовершенствованному методу синтеза конечных автоматов Мура по реализуемым ими алфавитным отображениям: для любого класса $sc \in SC$ по инъективному отображению F_{SteSch} (т.е. множеству каналов управления класса) требуется построить конечно-автоматную модель поведения A_{sc} , заданную соответствующей отмеченной таблицей переходов.

Таким образом, с точки зрения синтеза конечно-автоматной модели поведения класса ИМ СПС по реализуемому ею множеству каналов управления и согласно усовершенствованному методу синтеза конечных автоматов Мура метод автоматизации проектирования МПСК ИМ предполагает реализацию следующих этапов:

1. Построение для каждого класса $sc \in SC$ ИМ СПС алфавитного отображения $F_{\text{SteSch}}: \text{STE}_{sc} \rightarrow \text{SCH}_{sc}$, т.е. определение множества каналов управления класса.

2. Запись алфавитного отображения F_{SteSch} в виде таблицы соответствия, где каждая ее строка имеет вид: $\text{ste} \rightarrow F_{\text{SteSch}}(\text{ste})$ или $\text{ste} \rightarrow \text{sch}$, $\text{ste} \in \text{STE}_{sc}$, $\text{sch} \in \text{SCH}_{sc}$. Приведение полученного отображения к отображению F_{SteSch}^A , удовлетворяющему условиям автоматности [4].

3. Нахождение для исходного автоматного отображения F_{SteSch}^A канонического множества событий $\text{SFE}^A = \{\text{SFE}_{sh}\}$ во входном алфавите SE_{sc} данного отображения. $\text{SFE}_{sh} \in \text{SFE}^A$ (formal set of events) представляет собой автоматное событие, которое состоит из всех тех и только тех последовательностей событий класса, которые вызывают появление последовательностей деятельности, оканчивающихся одной и той же деятельностью $sh \in \text{SH}_{sc}$.

4. Построение множества регулярных выражений $\text{SRE}^A = \{\text{SRE}_{sh}\}$ для событий канонического множества. $\text{SRE}_{sh} \in \text{SRE}^A$ (set of regular expressions) представляет собой регулярное выражение для автоматного события SFE_{sh} , которое до выполнения тождественных преобразований задается в виде дизъюнкции начальных отрезков последовательностей событий класса, т.е. в виде дизъюнкции одноэлементных неэлементарных событий в алфавите SE_{sc} .

5. Выполнение алгоритма синтеза конечно-автоматной модели поведения класса ИМ СПС с естественной областью запрета по каноническому множеству событий SFE^A , заданному регулярными выражениями из множества SRE^A посредством своих выходных сигналов (в данной

интерпретации деятельности из множества \mathbf{SH}_{sc}):

- представление множества простых регулярных выражений правильно записанными регулярными выражениями;
- разметка мест правильно записанных регулярных выражений;
- приписывание индексов всем местам правильно записанных регулярных выражений;
- выполнение операций отождествления соответственных мест и отождествления подобных мест;
- распространение основных индексов всех основных мест, используемых в качестве неосновных индексов, на соответствующие подчиненные места;
- построение таблицы переходов конечно-автоматной модели поведения класса ИМ СПС;
- построение отмеченной таблицы переходов конечно-автоматной модели поведения класса ИМ СПС;
- нахождение неопределенных состояний конечно-автоматной модели поведения класса ИМ СПС;
- переобозначение выходных сигналов и состояний синтезированной конечно-автоматной модели поведения класса ИМ СПС.

6. Построение по полученной отмеченной таблице переходов конечно-автоматной модели поведения класса ИМ СПС соответствующей диаграммы в виде графа переходов состояний инициального частного конечного автомата Мура.

7. Приведение полученной диаграммы МПСК в нотации Мура к нотации Харела с целью использования возможностей автоматов Мили, а также концепций, относящихся к вложенным и параллельным состояниям.

Таким образом, предложенная методика позволяет автоматизировать процесс создания моделей при поуровневом переходе от моделей последовательностей сообщений объектов к моделям состояний классов и тем самым обеспечить их горизонтальную сбалансированность.

ЛИТЕРАТУРА

1. Буч Г., Рамбо Дж., Джекобсон А. *UML. Руководство пользователя*: Пер. с англ. – М.: ДМК, 2001. – 432 с.
2. Буч Г., Рамбо Дж., Джекобсон А. *UML. Специальный справочник*: Пер. с англ. – СПб.: Питер, 2002. – 656 с.
3. Вендров А.М. *Проектирование программного обеспечения экономических информационных систем*. – М.: Финансы и статистика, 2000. – 347 с.
4. Глушков В.М. *Синтез цифровых автоматов*. – М.: Физматгиз, 1962. – 476 с.
5. Калянов Г.Н. *CASE–технологии: консалтинг в автоматизации бизнес-процессов*. – М.: Горячая линия - Телеком, 2000. – 320 с.
6. Кузнецов О.П., Адельсон-Вельский Г.М. *Дискретная математика для инженера*. – М.: Энергоатомиздат, 1988. – 480 с.

Поступила 14.05.2002

ОЛИЗАРЕНКО Сергей Анатольевич, ст. научный сотрудник научного центра при ХВУ.
Окончил ХВВКИУРВ в 1993 г. Область научных интересов – системы обработки информации.
