

## **О ВОЗМОЖНОСТИ АВТОМАТИЧЕСКОЙ ТРАНСЛЯЦИИ UML-ДИАГРАММ ДЕЯТЕЛЬНОСТИ В СЕТИ ПЕТРИ\***

**И.В. ЗИМАЕВ**

Предложен ряд концептуальных соглашений в области построения UML-диаграмм деятельности, направленных на облегчение процесса автоматизации перевода диаграмм в сети Петри, существующий набор правил таких переходов расширен, приведена возможная последовательность трансляции.

**Ключевые слова:** UML-диаграммы, сети Петри, программная инженерия.

### **ВВЕДЕНИЕ**

Визуальное представление моделей сложных процессов и алгоритмов – основной способ проектирования современного программного обеспечения, позволяющий заранее проанализировать структуру и поведение рассматриваемой системы. Фактическим стандартом в области инструментария визуального моделирования являются UML-диаграммы, набор которых весьма разнообразен, но не снабжен четкими правилами по применению, а синтаксис самих диаграмм лишен какого бы то ни было формализма.

Для представления последовательностей действий (в том числе параллельных) в языке UML предусмотрены диаграммы деятельности (activity diagram), отчасти схожие с традиционными блок-схемами (ГОСТ 19.701-90). Диаграмма деятельности представляет собой граф, вершины которого соответствуют действиям, а дуги – переходам от одного действия к другому. Если при проектировании системы допущены ошибки, потенциально они могут быть выявлены при анализе самих диаграмм, эффективность которого напрямую зависит от опыта и внимательности проектировщика. Для более надежной автоматизированной верификации используется математический аппарат сетей Петри.

Сети Петри во многом похожи на диаграммы деятельности, но в отличие от них позволяют осуществлять имитационное моделирование, выявляющее различные ошибки модели [5]. В ряде работ [1–3, 7, 9] предложены некоторые правила преобразования UML-диаграмм в сети Петри, однако в силу отсутствия четких рекомендаций к построению диаграмм и некоторой неполноты

---

\* Статья получена 12 декабря 2009 г.

Работа выполнена при финансовой поддержке Федерального агентства по образованию, ГК № П694 от 12.08.2009, конкурс НК-81П

самих наборов правил процесс трансляции характеризуется определенными границами применимости и не автоматизирован. Классификация возможных элементов диаграмм деятельности (с описанием их смысловой нагрузки) и правил их сочетания может значительно увеличить шансы на полную автоматизацию трансляции.

## 1. КОНЦЕПТУАЛЬНЫЕ СОГЛАШЕНИЯ

Необходимо определить роль различных элементов диаграмм деятельности и концептуальные правила их применения. К основным элементам диаграмм деятельности относятся [4]: 1) действия (actions); 2) объекты (objects); 3) условные переходы (decision); 4) слияния (merge); 5) ветвления (fork); 6) объединения (join); 7) разделы (partitions).

**Правило 1.** Модель системы характеризуется выбранным уровнем абстракции, все элементы диаграммы соответствуют этому уровню [6], иные уровни абстракции модель не представляет, следовательно, они исключаются из верификации.

**Пример.** Элементами диаграммы деятельности являются компьютер, канал связи и запрос данных с терминала. Внутренняя структура и функционирование компьютера на этой диаграмме не раскрыты и при построении сети Петри верифицироваться не будут, поэтому условно полагается, что ошибки в работе этого блока отсутствуют.

**Правило 2.** Диаграмма отражает логику, т. е. работа системы на выбранном уровне абстракции (см. п. 1) отражена в виде графа действий над объектами с возможными ветвлениями, и вся необходимая информация о работе системы представлена самой структурой диаграммы.

**Пример.** Для срабатывания некоторого перехода необходимо соблюдения двух условий, например звонок телефона при условиях наличия вызова и незанятости линии. На диаграмме необходимо отдельно указать два объекта («звонок» и «линия свободна») с переходами к общему объединению (join), тем самым абстрагируясь от природы объектов в универсальной для данного случая схеме. Структура данной диаграммы однозначно описывает некое событие, наступление которого возможно при соблюдении двух условий, а если бы эти условия, к примеру, были указаны лишь в виде текстовых комментариев к одному элементу – логика условий была бы понятна только человеку, но утеряна при автоматическом переходе к сети Петри.

**Правило 3.** Каждое условие, необходимое для наступления некоторого события (т. е. срабатывания перехода join) является объектом.

**Пример.** В предыдущем случае событие «звонок телефона» есть операция над объектами «вызов» и «незанятость линии». Хотя это несколько против-

речит привычному пониманию, логично полагать, что у любого действия (события) есть объекты-причины (условия возникновения) и объекты-следствия (результат действия). В случае же условного перехода (decision) также происходит оперирование каким-либо объектом (точнее его свойством).

## 2. БАЗОВЫЕ ЭЛЕМЕНТЫ И ПРОСТЫЕ ТРАНСЛЯЦИИ

Трансляция переходов и действий осуществляется согласно следующему правилу [1], представленному на рис. 1.

Действию в UML-диаграмме соответствует переход сети Петри, и в подтверждение данной трактовки приводятся [2, с. 60] следующие аргументы.

1. Переходы в сетях Петри изначально предназначены для представления событий, а позиции отражают условия для возможности их срабатывания.

2. Маркёр сети Петри может быть остановлен только в позиции, но не на переходе. Если бы действия на диаграмме соответствовали бы позициям, такая ситуация могла бы быть истолкована как приостановка действия, которое на выбранном уровне абстракции рассматривается атомарным.

Потоки последовательности действий могут разделяться и соединяться [8], при этом каждый поток выполняется параллельно с другими (рис. 2). В

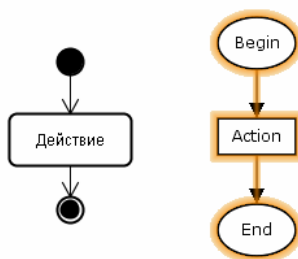


Рис. 1. Преобразование действия

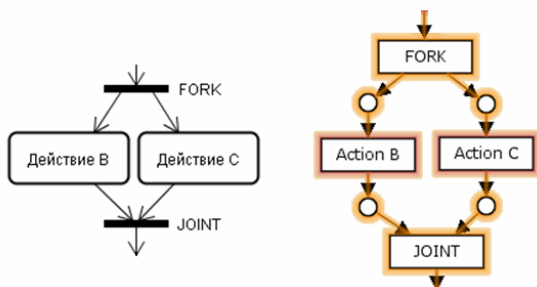


Рис. 2. Ветвления и объединения

сети Петри переход fork помещает маркёр во все свои выходные позиции, т. е. количество маркёровкратно увеличивается. Переход joint может сработать только при наличии маркёров во всех входных позициях, тем самым количество маркёровкратно уменьшается.

Отдельно необходимо отметить различие между объединением (join) и слиянием (merge). В отличие от объединения, слияние — это соединение  $N$  потоков действий, из которых активен только один, и таким

образом, для срабатывания перехода слияния не требуется наличия маркеров во всех входных позициях (рис. 3).

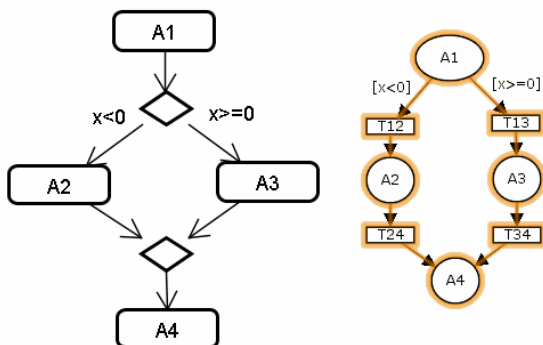


Рис. 3. Условие и слияние

В зависимости от условия, которое, по сути, является операцией над свойством объекта  $x$ , осуществляется переход либо к действию  $A_2$ , либо к действию  $A_3$ , срабатывание обоих переходов исключено по определению. После выполнения одного из двух действий необходимо перейти к действию  $A_4$  посредством перехода слияния (merge), что в

случае сети Петри реализуется двумя отдельными переходами, один из которых помещает свой маркер в общую выходную позицию.

### 3. ДИНАМИЧЕСКИЕ И СТАТИЧЕСКИЕ ОБЪЕКТЫ

Предположим, что для срабатывания некоего перехода  $T$  необходимо выполнения нескольких условий  $C_1 \dots C_N$ , что в синтаксисе сети Петри соответствует наличию маркеров во входных позициях перехода. Входные позиции для объединяющего перехода сопоставляются объектам диаграммы деятельности, и эти объекты подразделяются на статические и динамические.

Динамические объекты существуют или находятся в состоянии доступности не всегда и могут представлять условия, истинность которых вариативна во времени (например, объект «запрос» появляется в определенный момент времени).

Статические объекты существуют постоянно и непрерывно (в рамках времени работы системы). Это, например, запись в базе данных телефонных номеров, тогда как динамический объект – вызываемый в данный момент номер.

С точки зрения сети Петри динамический объект – это позиция, маркер в которой может быть в наличии, либо отсутствовать, статический объект – это позиция, в которой всегда есть маркер. Как правило, динамические объекты чаще участвуют в условиях, в которых важно само наличие маркера («запрос номера» и «канал связи свободен»), а для статических объектов условиями

учитываются его свойства («ввод пароля» и «правильный пароль» – должны совпадать).

В динамический объект маркёр попадает из какого-либо перехода и по другому переходу его покидает, таков критерий отличия этого типа объектов от статических, у которых после трансляции из UML-диаграммы есть только одна выходная дуга. Для обеспечения правильного функционирования таких переходов, необходимо возвращать маркёр после каждого срабатывания перехода (рис. 4).

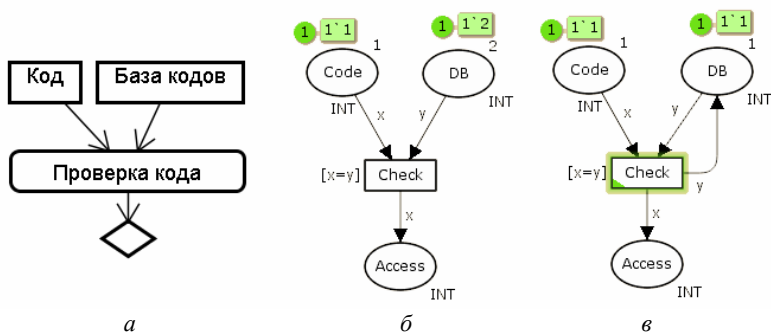


Рис. 4. Трансляция статических объектов

В данном примере представлен процесс сравнения некоего цифрового кода с эталонным, находящимся в базе данных (рис. 4, а). Приходящий код – это динамический объект, постоянно существующий эталонный код – это статический объект. Для постоянного сохранения маркёра в статическом объекте добавляется дополнительная дуга перехода (рис. 4, б и в).

#### 4. СТЕРЕОТИПЫ

Если диаграмма деятельности своей структурой достаточно полно описывает логику работы системы, то после трансляции в сеть Петри маркёр необходим лишь как активный указатель потока деятельности, при этом он не характеризуется какими-либо дополнительными свойствами. Это очень простой вариант сети Петри, ветвления которой не могут содержать условия (которые, впрочем, весьма трудно автоматически транслировать из диаграммы деятельности). Если же при построении сети Петри необходимо задавать тип объектов и соответствующих им маркёров (описывающих состояния объектов), можно воспользоваться таким понятием языка UML, как стереотипы (рис. 5).

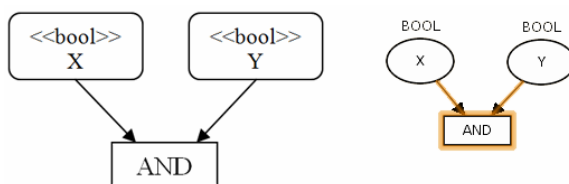


Рис. 5. Стереотипы и типы данных

В данном примере на основе диаграммы деятельности построена схема операций над логическими значениями (аналогично схемам цифровой электроники). Объекты *X* и *Y* снабжены пользовательским UML-стереотипом «bool», сообщающим о том, что эти объекты характеризуются значением логического типа. Блок AND представляет собой обычный join-переход, хотя вместо него может быть любой другой модуль (вложенная сеть, реализующая логическую операцию). При автоматической генерации сети Петри стереотипу «bool» ставится в соответствие тип данных BOOL (типы данных в раскрашенных сетях Петри принято называть цветами).

## 5. АВТОМАТИЗАЦИЯ ТРАНСЛЯЦИИ

В ряде источников описывается минимальный набор правил [3] по преобразованию UML-диаграмм деятельности в ординарную сеть Петри. В [1, с. 76] этот набор правил (*R1–R4*) был дополнен правилами *R5–R7* и итоговый ряд из семи правил имеет достаточно широкие границы применимости (в неавтоматическом режиме). Процесс автоматического преобразования UML-диаграммы деятельности в сеть Петри гипотетически представляется в виде следующей последовательности:

- 1) построение диаграммы с учетом рекомендаций и соглашений (правила 1–3);
- 2) преобразование основных элементов: действий и объектов (раздел «базовые элементы»), ветвлений и объединений (*R2*), условий (*R3*), критических секций и семафоров (*R5–R6*), разделов (*R7*);
- 3) преобразование merge-слияний (раздел «базовые элементы»);
- 4) преобразование статических объектов;
- 5) определение типов данных по стереотипам объектов;
- 6) добавление элементов генерации внешних сигналов;
- 7) добавление элементов работы со временем.

Необходимость экспериментальной проверки и дальнейшего развития методики автоматической трансляции требует разработки специализированного программного обеспечения.

## ЗАКЛЮЧЕНИЕ

Наличие формальных соглашений о трактовке назначения элементов диаграмм деятельности и правил их сочетания значительно облегчает возможность автоматической трансляции диаграмм в сеть Петри. Предложен подход полного отражения графа действий в структуру диаграммы (без использования типов данных и условий переходов), что позволяет сохранить информацию о поведенческой модели системы при автоматическом переходе к сети Петри. Дополнен существующий ряд правил перевода диаграмм в сети в совокупности с использованием UML-стереотипов как дескрипторов типов данных сети.

[1] *Коротиков С.В.* Применение сетей Петри в разработке программного обеспечения центров дистанционного контроля и управления: дис. ... канд. техн. наук. – Новосибирск: НГТУ, 2007.

[2] *Питерсон Дж.* Теория сетей Петри и моделирование систем / пер. с англ. – М.: Мир, 1984.

[3] *Eshuis R.* A comparison of activity diagram and Petri net variants / R. Eshuis, R. Wieringa // Proc. 2 Int. coll. on Petri net technolog. for model. com. based syst., Berlin (Germany), 2001. – P. 93–104.

[4] *Ларман К.* Применение UML 2.0 и шаблонов проектирования. – 3-е изд. / пер. с англ. – М.: ООО «И.Д. Вильямс», 2007.

[5] *Воевода А.А., Зимаев И.В.* Моделирование системы многокальной визуализации с использованием аппарата сетей Петри // Сб. науч. тр. НГТУ. – 2008. – № 3 (53). – С. 43–48.

[6] *Воевода А.А., Зимаев И.В.* Соотношения структурных и временных масштабов UML-диаграмм / Сб. науч. тр. НГТУ. – 2008. – № 4 (54). – С. 59–62.

[7] *Воевода А.А., Зимаев И.В.* Об особенностях преобразования UML-диаграмм деятельности в сети Петри // Сб. науч. тр. НГТУ. – 2009. – № 2(56). – С. 79–89.

[8] *Зимаев И.В., Воевода А.А.* Моделирование асинхронной сети автоматов обработки данных // Сб. науч. тр. НГТУ – 2009. – № 4(58). – С. 147–15.

[9] *Воевода А.А., Романников Д.О., Зимаев И.В.* Применение UML-диаграмм и сетей Петри при разработке встраиваемого программного обеспечения // Науч. вестн. НГТУ. – 2009. – № 3(36).

**Зимаев Игорь Валентинович** – аспирант кафедры автоматики Новосибирского государственного технического университета.

Тел. (383) 346-11-19

**I.V. Zimaev**

**An approach to the automatical translation between UML activity diagrams and Petri nets.**

Article considers some rules of transitions between UML diagrams and Petri nets, including a number of conceptual agreements about diagrams development, also shows a possible translation's sequence.

**Key words:** UML, Petri net, CPN, software engineering.