

Зинкина Н.С. Логико-алгебраические спецификации имитационных моделей. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей IX Междунар. научно-техн. конф. – Пенза: ПДЗ, 2009. – С. 60-65.

ЛОГИКО-АЛГЕБРАИЧЕСКИЕ СПЕЦИФИКАЦИИ ИМИТАЦИОННЫХ МОДЕЛЕЙ АГЕНТНО-ОРИЕНТИРОВАННЫХ СИСТЕМ

Н.С. Зинкина

Пензенский государственный университет,
г. Пенза, Россия

Предлагается система моделирования, главной особенностью которой является то, что в ее основу положены механизмы реализации имитационных моделей на базе исполняемых логико-алгебраических спецификаций, т.е. на базе сценарных представлений и сетей абстрактных машин.

Zinkina N.S. Logical-algebraic specifications of simulation models in agent-oriented systems.

The article describes a modeling system, the main peculiarity of which is the mechanisms of realization the simulation models on the basis of logical-algebraic specifications, that is on the basis of scenario performances and networks of abstract machines.

С помощью систем моделирования на ЭВМ реализуется метод имитационного моделирования, а в самих имитационных моделях отображаются структура и динамика проектируемой или исследуемой системы. Адекватность имитационной модели определяется степенью представления разработчика о функционировании системы, точностью определения или предсказания ее параметров. Системы моделирования дают возможность быстрой перестройки имитационных моделей при возникновении новых ситуаций, например, при изменении технического задания на разработку вычислительной системы или ее операционной системы, при изменении режима ее использования.

Сложность современных вычислительных систем и устройств, отсутствие во многих случаях близких по характеристикам и структуре прототипов приводят разработчиков к необходимости использования имитационных моделей различных уровней. Имитационные модели традиционно используются для предсказания характеристик производительности вычислительных систем. Значительно меньше внимания уделялось использованию имитационных систем как средств проектирования и дальнейшей проверки правильности функционирования вычислительных систем и устройств. Это можно объяснить тем, что системы моделирования с подходящим для целей инженерной практики входным языком появились сравнительно недавно.

Общим недостатком многих известных систем является то, что процесс составления модели близок к процессу обычного программирования и кодирования задач, что затрудняет составление и верификацию самих моделей. Отличительными признаками предлагаемой системы моделирования являются способы реализации языка моделирования и его синтаксиса, средств редактирования и интерпретации функций и предикатов, модулей, атрибутов активностей. Главной особенностью системы моделирования является то, что в ее

основу положены механизмы реализации имитационных моделей на базе исполняемых логико-алгебраических спецификаций, т.е. на базе сценарных представлений и сетей абстрактных машин. Проиллюстрируем основные особенности использования подобного подхода, отмечая при этом, что его сфера применения шире, чем создание имитационных систем.

Рассмотрим некоторые подсистемы систем массового обслуживания, содержащие обслуживаемые устройства (центры обслуживания или обработки запросов), входные и выходные очереди. Функционирование данных подсистем опишем логико-алгебраическими выражениями СеАМ. Пусть, например, простая подсистема массового обслуживания включает в свой состав две входные очереди f_1 и f_2 , центр обслуживания c_1 и одну выходную очередь f_3 . Рассмотрим три варианта, или дисциплины обработки запросов в данной подсистеме и соответственно три логико-алгебраических выражения, описывающих данные варианты.

1. Дисциплина 1: в случае, если во входной очереди f_1 содержится не менее одного запроса, во входной очереди f_2 содержится два запроса, а выходная очередь f_3 пуста, то центр обслуживания c_1 выбирает на обработку один запрос из очереди f_1 и после обработки помещает ответ в очередь f_3 . Число запросов в очереди f_2 сохраняется неизменным. В дальнейшем будем использовать один термин «запрос» для обозначения запросов, заявок, ответов или других видов активностей. Работа центра обслуживания описывается следующей формулой:

$$c_1 = [f_{ge}(f_1(a_1), 1) \& f_{eq}(f_2(a_2), 2) \& f_{eq}(f_3(a_3), 0)] (\{f_1(a_1) \leftarrow f_{minus}(f_1(a_1), 1), f_3(a_3) \leftarrow 1\} \vee R_E),$$

где значениям унарных функций f_1 , f_2 и f_3 соответствуют длины одноименных очередей; f_{ge} и f_{eq} – бинарные предикаты сравнения на «больше или равно» и «равно»; f_{minus} – функция вычитания; R^E – тождественная модификация интерпретации текущей сигнатуры эволюционирующей алгебраической системы.

2. Дисциплина 2: если во входной очереди f_1 имеется в наличии не менее одного запроса, во входной очереди f_2 – не менее двух запросов, а выходная очередь f_3 пуста, то из очереди f_1 выбирается на обработку один запрос, из очереди f_2 удаляются все запросы, а в выходную очередь f_3 добавляется один запрос. Описанные действия реализуются следующим модулем:

$$c_2 = [f_{ge}(f_1(a_1), 1) \& f_{ge}(f_2(a_2), 2) \& f_{eq}(f_3(a_3), 0)] (\{f_3(a_3) \leftarrow f_{plus}(f_3(a_3), 1), f_1(a_1) \leftarrow f_{minus}(f_1(a_1), 1), f_2(a_2) \leftarrow 0\} \vee R_E),$$

где f_{plus} – функция сложения.

3. Дисциплина 3: если во входной очереди f_1 не менее двух запросов, в очереди f_2 – не менее четырех запросов, а в выходной очереди f_3 не более пяти запросов, то из очереди f_1 выбирается на обработку один запрос, из очереди f_2 выбираются на обработку все запросы, а к содержимому выходной очереди добавляются все обработанные запросы. Логико-алгебраическое выражение имеет следующий вид:

$$c_3 = [f_{ge}(f_1(a_1), 2) \& f_{ge}(f_2(a_2), 4) \& f_{le}(f_3(a_3), 5)] (\{f_3(a_3) \leftarrow f_{plus}(f_3(a_3), f_{plus}(f_1(a_1), f_2(a_2)))\}; (f_1(a_1) \leftarrow f_{minus}(f_1(a_1), 1), f_2(a_2) \leftarrow 0\} \vee R_E).$$

Во всех трех приведенных выше примерах в случае, если хотя бы одно условие не выполняется, центр обслуживания простаивает и ожидает выполнения всех условий. Напомним, что в третьем примере символ точки с запятой, разделяющий правила обновления, заключенные в фигурные скобки, является упрощенным символом темпоральной операции непосредственного следования, а символ простой запятой, разделяющей правила обновления во всех трех примерах,

является упрощенным символом темпоральной операции возможной одновременности. В выражениях для модулей s_1 , s_2 и s_3 символы a_1 , a_2 и a_3 являются именами абстрактных агентов, интерпретирующих логико-алгебраические выражения. Агенты проверяют при этом значения предикатов и выполняют согласованные правила обновления интерпретации текущей сигнатуры. В каждом из выражений s_1 , s_2 и s_3 достаточно было бы использовать по одному агенту, но даже в таких несложных выражениях использование нескольких агентов может дать выигрыш во времени исполнения.

Рассмотрим далее логико-алгебраическую модель простой очереди «с проталкиванием», или с известной в теории массового обслуживания дисциплиной FIFO (First In – First Out). Данная дисциплина описывается и затем реализуется следующими выражениями:

$$\begin{aligned} m_1 &= [p_1(a_1) \& \neg p_2(a_2)] (\{p_1(a_1) \leftarrow \text{false}, p_2(a_2) \leftarrow \text{true}\} \vee R^E), \\ m_2 &= [p_2(a_2) \& \neg p_3(a_3)] (\{p_2(a_2) \leftarrow \text{false}, p_3(a_3) \leftarrow \text{true}\} \vee R^E), \\ &\dots \\ m_i &= [p_i(a_i) \& \neg p_{i+1}(a_{i+1})] (\{p_i(a_i) \leftarrow \text{false}, p_{i+1}(a_{i+1}) \leftarrow \text{true}\} \vee R^E), \\ &\dots \\ m_{n-1} &= [p_{n-1}(a_{n-1}) \& \neg p_n(a_n)] (\{p_{n-1}(a_{n-1}) \leftarrow \text{false}, p_n(a_n) \leftarrow \text{true}\} \vee R^E), \end{aligned}$$

где унарные предикаты p_1, p_2, \dots, p_n используются для представления n мест для ожидания в очереди (символ n соответствует началу, или «голове» очереди), причем логика работы модулей СеАМ построена таким образом, что каждый новый запрос стремится занять место, непосредственно примыкающее к месту, где находится предыдущий запрос. Унарным предикатам p_1, p_2, \dots, p_n и абстрактным агентам с именами a_1, a_2, \dots, a_n соответствуют информационные объекты и программно или аппаратно реализованные «конкретные» агенты. Последние, изменяя состояние информационных объектов, осуществляют перемещение запросов. В приведенном примере каждое место для ожидания представлено отдельным предикатом, причем область определения каждого предиката p_i содержит единственный элемент a_i . Рассмотрим случай, когда вся очередь представлена единственным унарным предикатом p , областью определения которого является множество имен абстрактных агентов $\{a_1, a_2, \dots, a_n\}$. Система выражений, описывающая подобную реализацию очереди FIFO, имеет следующий вид:

$$\begin{aligned} m'_1 &= [p(a_1) \& \neg p(a_2)] (\{p(a_1) \leftarrow \text{false}, p(a_2) \leftarrow \text{true}\} \vee R^E), \\ m'_2 &= [p(a_2) \& \neg p(a_3)] (\{p(a_2) \leftarrow \text{false}, p(a_3) \leftarrow \text{true}\} \vee R^E), \\ &\dots \\ m'_i &= [p(a_i) \& \neg p(a_{i+1})] (\{p(a_i) \leftarrow \text{false}, p(a_{i+1}) \leftarrow \text{true}\} \vee R^E), \\ &\dots \\ m'_{n-1} &= [p(a_{n-1}) \& \neg p(a_n)] (\{p(a_{n-1}) \leftarrow \text{false}, p(a_n) \leftarrow \text{true}\} \vee R^E). \end{aligned}$$

Выбор конкретной системы СеАМ-выражений определяется требуемой производительностью системы массового обслуживания и затратами памяти на размещение объектов.

Для реализации дисциплин с прерыванием начавшегося обслуживания воспользуемся известным в вычислительной технике отношением прерывания $INT \subset Z \times Z$, где Z – множество запросов на обслуживание. Это иррефлексивное, транзитивное и антисимметричное отношение. Реализуя дисциплины обслуживания с прерываниями, в выражениях необходимо использовать правила

обновлений одноименного предиката вида $INT(z_i, z_j) \leftarrow true$ (запрос z_i может прервать начавшееся обслуживание запроса z_j), а при запрете прерывания – правило $INT(z_i, z_j) \leftarrow false$. Отношение INT используется для указания на возможность прерывания. Для фиксации факта совершившегося прерывания можно использовать другое отношение $ISINT \subset Z \times Z$, модифицируя его с помощью правил вида $ISINT(z_i, z_j) \leftarrow true$ и $ISINT(z_i, z_j) \leftarrow false$. Здесь, как и ранее, предполагается, что отношение $ISINT$ представлено областью истинности одноименного предиката. Истинность высказывания $ISINT(z_i, z_j)$, например, означает, что начавшееся исполнение, или обслуживание запроса z_j прервано запросом z_i .

Процессы занятия и освобождения ресурсов в системах массового обслуживания могут быть так же описаны логико-алгебраическими выражениями $SeAM$ и $PSeAM$. Пусть, как обычно, P – множество процессов, R – множество ресурсов, RR и RU – бинарные предикаты вида

$$RR: P \times R \rightarrow \{true, false\}, RU: R \times P \rightarrow \{true, false\},$$

где истинность высказывания вида $RR(p_i, r_j)$ означает, что процесс p_i затребовал ресурс r_j , а истинность высказывания вида $RU(r_j, p_i)$ означает, что ресурс r_j предоставлен в распоряжение процесса p_i .

Введем далее унарные предикаты, характеризующие активность процессов и занятость ресурсов соответственно:

$$PA: P \rightarrow \{true, false\}, RB: R \rightarrow \{true, false\}.$$

Истинность высказывания $PA(p_i)$ означает активность процесса p_i , а истинность высказывания $RB(r_i)$ означает, что ресурс r_i занят каким-либо процессом.

Рассмотрим в качестве примера задачу, в которой процессу p_1 надо предоставить в распоряжение ресурсы r_1 и r_2 . Следующее логико-алгебраическое выражение для модуля M_r $SeAM$ описывает процесс предоставления ресурсов одному процессу:

$$M_r = [PA(p_1) \& \neg RB(r_1) \& \neg RB(r_2) \& RR(p_1, r_1) \& RR(p_1, r_2)] \\ (\{RB(r_1) \leftarrow true, RB(r_2) \leftarrow true, RU(r_1, p_1) \leftarrow true, RU(r_2, p_1) \leftarrow true, \\ RR(p_1, r_1) \leftarrow false, RR(p_1, r_2) \leftarrow false\} \vee R^E).$$

Выражения, которыми описывается функционирование модулей абстрактных машин, удобны для преобразований и непосредственной интерпретации и могут быть использованы в качестве основы для реализации системы моделирования на основе концепции непосредственно интерпретируемых формальных спецификаций. Использование многосортных логик и логик высших порядков позволяет плодотворно сочетать как декларативные, так и процедурные подходы к построению имитационных моделей распределенных систем хранения и обработки данных. Построение имитационных, или поведенческих, моделей систем массового обслуживания базируется на согласованных взаимодействиях объектов через общее пространство – коммуникационную среду или общее пространство информационных объектов.

Библиографический список

1. Зинкин С.А. Согласование и координация объектов и процессов в агентно-ориентированных системах и сетях хранения данных // Вопросы радиоэлектроники. – 2009. – № 4. – С. 83 – 96.
2. Зинкин С.А. Реализация барьерной синхронизации и управление процессами в виртуальном сетевом дисковом массиве // Информационные технологии. – 2008. – № 12. – С. 22 – 29.