УДК 004.942: 519.876.5

М.А. Волк, Т.В. Филимончук, Р.Н. Гридель, Ал Шиблак Муаз

Харьковский национальный университет радиоэлектроники, Харьков

ФОРМАЛИЗАЦИЯ ПРОЦЕССОВ РАСПРЕДЕЛЕННОЙ ИМИТАЦИИ ИНФОРМАЦИОННЫХ СИСТЕМ

Предлагается формальное описание процессов в распределенных системах имитационного моделирования. В качестве элементов описания выступают активности, отображающие функциональность модулей моделирующей среды, и данные, реализующие ресурсы моделей. Рассматривается процесс управления локальными и глобальными данными моделей. Вводится понятие менеджера памяти моделей как инструмента унификации управления распределенными моделями с целью поддержки стандарта распределенного имитационного моделирования HLA. Предложенное формальное описание может быть использовано при создании современных систем имитационного моделирования информационных систем.

Ключевые слова: *распределенная имитационная модель, информационные системы, процесс, активность, данные.*

Введение

Использование распределенного имитационного моделирования в науке и технике позволило значительно расширить класс реализуемых имитационных моделей за счет использования большого количества мощных вычислительных ресурсов, к которым можно отнести локальные и глобальные компьютерные сети, суперкомпьютеры, кластера. Одной из актуальных задач в этой области является формализация процессов, протекающих в распределенных имитационных средах. Актуальность задачи усиливается значительной стоимость эксплуатации такого вида ресурсов.

Существует ряд работ в этой области, среди которых можно выделить два основных направления. Первое из них используется в тех случаях, когда в проблемной области, для которой выполняется построение модели, существуют традиционные средства моделирования. Например, в теории массового обслуживания широкое распространение получили программные системы моделирования на базе языка GPSS [1]. Для исследования дискретных динамических систем более широкого класса – аппарат сетей Петри [2].

В этих случаях разработано множество методических рекомендаций, научных аналитических публикаций, которые помогут разработчику выбрать нужные средства моделирования [3]. Второе направление ориентировано на природу имитационной модели как представителя того или иного методологического подхода или использует идею о том, что любая имитационная модель представляет собой программный продукт. К таким работам следует отнести [4 – 6].

В данной статье предлагается формализация, выполненная в рамках второго направления.

Цель работы и основные понятия

Целью статьи является формальное представление распределенной имитационной модели информационных систем. За основу примем представление имитационной модели, приведенное в [4]. Предлагается следующий алфавит активностей модели:

$$\mathrm{Act} = \Phi \cup \Lambda \cup \Delta, \{\varphi\}_{f=1}^{F} \subseteq \Phi, \{\lambda\}_{l=1}^{L} \subseteq \Lambda, \{\delta\}_{d=1}^{D} \subseteq \Delta, \quad (1)$$

где Act – алфавит активностей; Φ – множество внутренних активностей модели; Λ – множество активностей управления данными модели; Δ – множество активностей взаимодействия моделей.

Отметим, что такое представление расширяет модель, множеством активностей управления данными модели, поэтому для сохранения эквивалентности моделей укажем, что множество Λ было выделено из множества Φ по функциональному признаку обработки данных модели, то есть при переходе к модели [4] выполняется $\Lambda \subset \Phi$. Модель в [7] более подробна (мощность алфавита больше), поэтому для установления эквивалентности моделей введем отношения:

$$\{\text{АП}, \text{М}_\text{Т}\} \subset \Phi; \{\text{ДМ}\} \subseteq \Lambda, \{\text{Н}\} \subseteq \Delta \ . \quad (2)$$

Таким образом, положения, изложенные в [4, 7] будут распространяться и на модель, обладающую алфавитом активностей (1).

В имитационной модели выделяют событие (event) *e*, которое приводит либо к инициализации одной из активностей модели, либо к изменению состояния модели. Также существует понятие процесса *P*, который является ответом на это событие *e*. В обозначениях процессной алгебры последовательность *событие→процесс* обозначается как *e→P* (за событием *e* следует выполнение процесса *P*). В нашем случае непосредственно выполнение процесса реализуется активностями, поэтому будем счи-

© М.А. Волк, Т.В. Филимончук, Р.Н. Гридель, Ал Шиблак Муаз

тать записи $e \to P$ и $e \to A$ эквивалентными.

Большинство расширений процессной алгебры включает понятие состояния s объекта (модели, программы, процесса) и понятие перехода объекта из одного состояния (*i*) в другое (*j*): $s_i \xrightarrow{A} s_j$. Осуществление перехода происходит под воздействием какой-либо активности А.

Состояние модели *s,* события *e*, а также другие элементы модели, которые в программной реализации представлены данными (сегментом, данных, фрагментом памяти и т.п.), будем непосредственно связывать с данными модели *dm*, то есть $s,e \subset dm$ Действительно, любая программа (в том числе и программная модель) определяется как совокупность данных и кода, который эти данные изменяет в процессе исполнения программы (моделирования). В этом случае понятия *изменения состояния модели* и *изменения данных модели* будем считать эквивалентными. Или, другими словами, данные модели являются физической реализацией абстрактного понятия состояния модели. К подобным элементам также относятся транзакты, сообщения, локальные и глобальные переменные, входные и выходные параметры модели т.д. Изменения в модели, приводят к изменению данных под влиянием одной из активностей $dm \xrightarrow{A_i} dm'$.

Таким образом, элементами имитационной модели являются активности A_i и данные (dm_i). Активности A_i осуществляют функционирование на основе данных модели. Рассмотрим более подробно классификацию данных модели.

Выделим среди всей совокупности данных dm_i те, которые доступны только из частной модели (рис. 1). Назовем их локальными данными частной модели dm_i^l. Управляющая программа моделирования (УПМ) выделяет требуемую область памяти данных для активности A_i и передает права на использование этой памяти активностям *i*-й частной модели. Все операции с dm_i^l осуществляются исключительно кодом активности из множества A_i. К этой же группе данных можно отнести те данные, которые может использовать УПМ или мониторинговые программы только для функций чтения, так как они не изменяют состояния частной модели. Обычные dm_i^l размещаются совместно с кодом активности, реализуя, таким образом, свойство инкапсуляции объектно-ориентированного программирования [8].

Глобальные данные модели используются для реализации двух основных функций:

1. Хранение параметров внешней (по отношению к частной модели) среды (dm^{gs}).

2. Передача данных, отвечающих за взаимодействие частных моделей между собой (dm^{gm}).

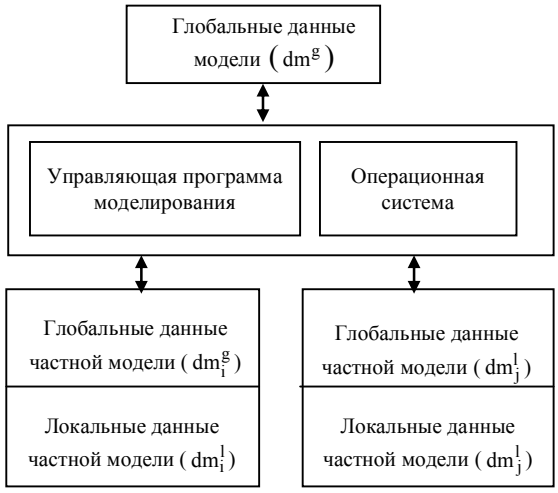

Рис. 1. Глобальные и локальные данные имитационной модели

Управляющая программа моделирования отвечает за реализацию следующих отношений:

$$dm^g = \bigcup_i^n dm_i^g; dm^g = dm^{gm} \cup dm^{gs}; \quad dm^{gs} \cap dm^{gm} = \varnothing. \tag{3}$$

Нарушение отношений (3) считается ошибкой моделирования. Условие не пересечения множеств dm^{gs} и dm^{gm} дает возможность разнести данные, определяющие состояние частной модели и данные, передаваемые в виде сообщений между частными моделями.

Это дает возможность размещения этих данных на разных ресурсах и под управлением разного программного обеспечения (осуществить виртуализацию хранения данных двух типов).

Определим локальные данные частных моделей как данные, существующие только во время выполнения активности. Локальные данные частной модели создаются перед выполнением активности и уничтожаются при ее завершении. Такое определение локальных данных, с одной стороны, соответствует локальным переменным большинства современных систем программирования (локальные данные создаются в стеке подпрограммы на время ее вызова, при условии отсутствия статических локальных переменных), с другой стороны, позволяет модифицировать определение транзакционной модели, введенной в [9], и считать транзакционной моделью ту, которая выполняет все операции с глобальными данными модели через интерфейс менеджера памяти.

В результате такого определения исключим из дальнейшего рассмотрения локальные данные частных моделей и подсистем моделирования, считая их префиксными и суфиксными активностями для любой активности частной модели:

$$\begin{cases} dm_j^g \xrightarrow{A(dm_j^l).A_i.A(dm_j'^l)} dm_j'^g; \\ dm_j^g \cap dm_j^l = \varnothing; \\ dm_j'^l \cap dm_j'^g = \varnothing, \end{cases} \quad (4)$$

где A_i – любая активность частной модели; $A(dm_j^l)$ – активность создания локальных данных для активности частной модели; $A(dm_j'^l)$ – активность уничтожения локальных данных после выполнения активности частной модели. Как правило, реализация последних двух активностей осуществляется средствами операционной системы.

Активности управления данными модели

Большинство современных имитационных моделей реализуют свои оригинальные средства управления данными. В работе [4] высказывается предложение создания библиотеки алгоритмов управления состоянием распределенных имитационных моделей. Такой подход сталкивается с трудностями, вызванными разнородностью формального описания различных подходов к моделированию. Один из немногих стандартов в области распределенного имитационного моделирования – HLA, – накладывает обязательства на программные модели предоставлять системе моделирования федераты (federates), отвечающие за возврат модели во времени, то есть стандартизованный интерфейс. При этом реализация федерата может быть произвольной (рис. 2). Ф1,…,ФN представляют собой интерфейсные функции, реализуемые федератами. Следует отметить, что HLA стандартизует только интерфейсы федератов с RTI, а не сами реализации. Поэтому корректность исполнения распределенной федерации, т.е. гарантия, что консервативному федерату не будет послано сообщение с временной меткой, меньшей, чем его локальное модельное время, обеспечивается только при условии корректной реализации самого федерата, ответственность за которую несет разработчик федерата.

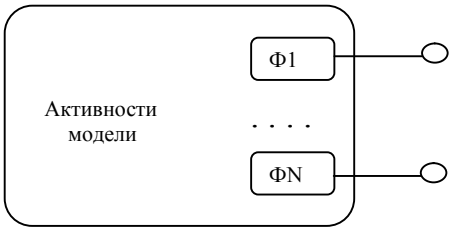

Рис. 2. Реализация федератов в имитационной модели

На основе представленной выше модели, предлагается новый подход – ввести стандарт на операции с данными модели и реализовать модуль менеджера управления данными модели (рис. 3). Реализация этого действия открывает принципиальную возможность реализовать любой из федератов, управляющих состоянием модели во времени и освободить разработчика модели от необходимости реализации этого интерфейса вручную.

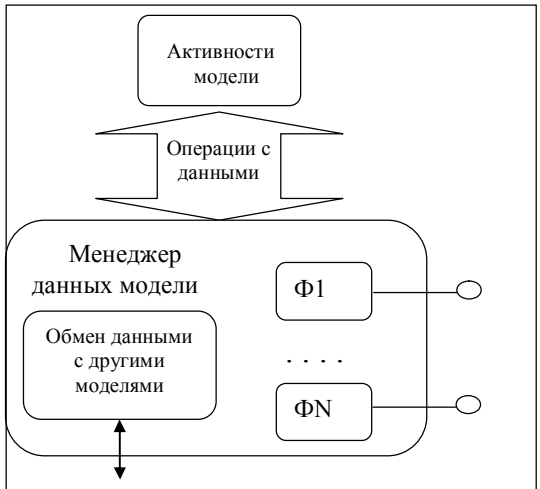

Рис. 3. Реализация федератов и активностей взаимодействия моделей в менеджере данных

Из множества активностей A^{data}, которые может выполнять менеджер данных модели, выберем активность, реализующую сохранение данных модели (дамп памяти) $A^{save}(T_k)$ и активность, позволяющую вернуть состояние модели в один из моментов модельного времени в прошлом $A^{load}(T_k)$:

$$A^{data} = \left\{A^{save}(T_k), A^{load}(T_k)\right\}. \quad (5)$$

Доказано, что используя активность сохранения данных модели до или после выполнения поведенческих активностей в фиксированный момент модельного времени можно достигнуть однозначности в определении состояния частных моделей в этот момент модельного времени. Рассмотрим применение менеджера памяти для консервативных и оптимистических алгоритмов синхронизации модельного времени.

Активности менеджера памяти, реализующие интерфейсы HLA, обеспечивающие алгоритмы синхронизации модельного времени.

Множество активностей менеджера памяти A^{data} получается путем объединения подмножеств активностей обработки данных модели и соответствующих активностям, реализующим консервативные и оптимистические федераты:

$$A^{data} = A^{data}_{опт} \cup A^{data}_{конс} =$$
$$= \left\{A^{save}(T_k), A'^{load}(T_k), A^{rav}, A^{tag}(T_e), A^{TSO}, A^{fqg}\right\}.$$

Отметим, что пересечение указанных двух множеств не есть пустое множество:

$$A^{data}_{опт} \cap A^{data}_{конс} = \left\{ A^{save}(T_k), A'^{load}(T_k), A^{rav} \right\}, \quad (6)$$

то есть, ряд активностей используется как в оптимистических, так и в консервативных алгоритмах синхронизации модельного времени.

Активности обмена данными частных моделей

Распределенное имитационное моделирование, в общем случае, подразумевает параллельное исполнение частных моделей, которые в процессе функционирования обмениваются друг с другом некоторой информацией. Это могут быть сообщения (которые поддерживаются наиболее известными интерфейсами параллельного программирования, например, MPI – Massage Passing Interface), взаимодействие по входными или выходным данным модели, объекты синхронизации при использовании разделяемых ресурсов, объекты сетевого взаимодействия сеансового уровня (сокеты) и т.д.

Так как любая подобная информация программно реализуется посредством данных модели, то введем в множество активностей менеджера памяти дополнительные активности для реализации функций обмена данными между частными моделями. Стандарт HLA не описывает данные элементы имитационных моделей, то есть подразумевается, что эта функциональность реализуется самостоятельно создателями моделей в произвольной форме. Формализация же этой функциональности открывает дополнительные возможности для реализации методов анализа эффективности применения алгоритмов синхронизации распределенных имитационных моделей.

Процесс передачи информации взаимодействия частных моделей содержит два этапа: подготовка данных в области локальных или глобальных данных модели (создание и заполнения буфера данных) и выполнение интерфейсной функции, непосредственно передающей данные с указанием в параметрах функции адреса буфера и объема передаваемых данных.

Введем следующее ограничение: буфер данных buf для передачи информации между частными моделями и адресная информация addr должны быть создан в глобальных данных модели:

$$\{buf, addr\} \subset dm^g .$$

Введем множество активностей A^s менеджера памяти для реализации пересылки данных между частными моделями:

$$A^s = \{ A^{send}(buf, addr),$$

$$A^{asend}(buf, addr), A^{hsend}(buf, addr) \} \subset A^{data}, \quad (7)$$

где $A^{send}(buf, addr)$ – синхронная передача данных с ожиданием подтверждения завершения процесса передачи информации; $A^{asend}(buf, addr)$ – асинхронная передача данных без ожидания подтверждения завершения процесса передачи информации; $A^{hsend}(buf, addr)$ – полусинхронная передача, которая заключается в одновременной передаче всех сообщений поведенческих активностей, накопленных в фиксированный момент модельного времени.

Перечисленные активности должны иметь соответствующие им активности менеджера памяти на принимающей стороне:

$$A'^s \{ A'^{send}(buf, addr),$$

$$A'^{asend}(buf, addr), A'^{hsend}(buf, addr) \} \subset A^{data}. \quad (8)$$

Активности частной модели могут использовать как все активности передачи данных, так и любое их подмножество.

Важно отметить, что операция передачи по сети данных модели включает в себя несколько уровней представления данных (рис. 4).

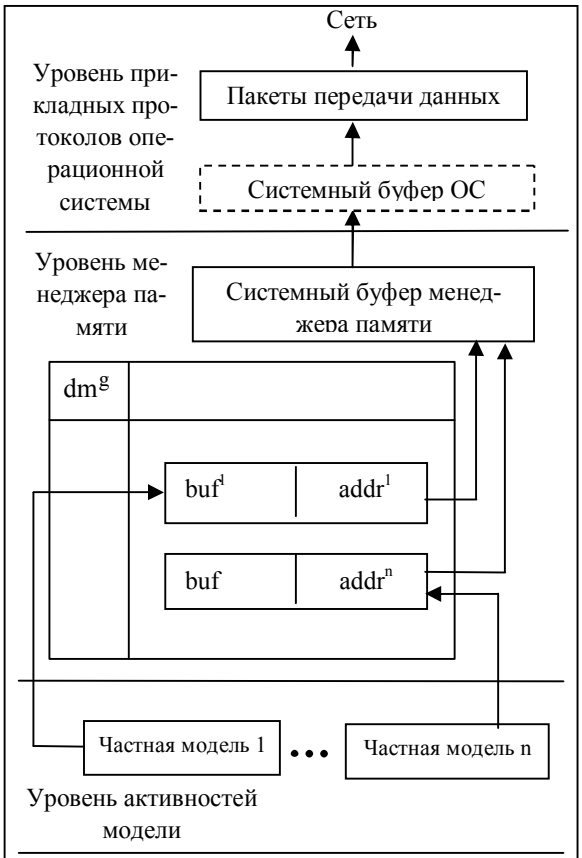

Рис. 4. Передача информационных данных частных моделей в сети

Частные модели осуществляют подготовку данных в буфере данных, расположенном в адресном пространстве глобальных данных модели. Затем вызываются активности из множества (8), осуществляющие копирование данных в системный буфер интерфейса

обмена данными по сети (в нашем случае – это менеджер памяти). После чего происходит обращение к функциям (или объектам) операционной системы, осуществляющим формирование пакетов (прикладного, представительного или сеансового уровней модели сети).

Перспективным является использование на смежных уровнях одного и того же участка памяти для реализации буфера данных, что даст возможность сократить временные затраты на копирование данных.

Выводы

В работе предложено новое формальное представление распределенной имитационной модели на основе выделения совокупности активностей модели и данных модели, отражающих изменение состояний модели во времени и информационные сообщения между активностями частных моделей. На ее основе показана возможность управления распределенными имитационными моделями.

Введено понятие менеджера памяти модели. Показана возможность построения на основе введенных активностей оптимистических алгоритмов синхронизации распределенных имитационных моделей на основе стандарта HLA. Показано, что при использовании менеджера памяти для управления состоянием в пространстве модельного времени, возможна автоматическая реализация федератов стандарта HLA и федератов распределенной системы моделирования RTI, поддерживающих консервативные и оптимистические алгоритмы синхронизации средствами менеджера памяти. Рассмотрен процесс передачи сообщений между частными моделями с точки зрения управления потоками данных, осуществляемого менеджером памяти при помощи одного из протоколов операционной системы.

Полученные результаты открывают возможности унификации программных имитационных моделей, ведущие к упрощению и ускорению процесса их построения на основе простого интерфейса, основанного на управлении данными модели, а также создает предпосылки к созданию эффективных методов.

Список литературы

1. *Томашевский В. Имитационное моделирование в среде GPSS / В. Томашевский, Е. Жданова. – М.: Бестселлер, 2003. – 416 с.*
2. *Юдицкий С.А. Метод анализа конфигураций организационных систем на сетях Петри / С.А. Юдицкий, И.А. Мурадян // УБС. – 2007. – №16. – С. 163-170.*
3. *Крэйн М. Введение в регенеративный метод анализа моделей / М. Крэйн, О. Лемуан. – М.: Наука, 1982. –104 с.*
4. *Окольнишников В.В. Разработка средств распределенного имитационного моделирования для многопроцессорных вычислительных систем / Диссертация. д-ра техн. наук: 05.13.18 / Окольнишников В.В. – Новосибирск, 2006. – 227 с. РГБ ОД71:07-5/433*
5. *Mitra D. Analysis and optimum performance of two message-passing parallel processors synchronized by rollback / D. Mitra , I. Mitrani // Performance'84. – 1984.– P. 35-50.*
6. *Миков А.И. Программные средства оптимизации распределенного имитационного эксперимента / А.И. Миков, Е.Б. Замятина, А.А. Козлов // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской суперкомпьютерной конференции (21-26 сентября 2009 г., г.Новороссийск). – М.: Изд-во МГУ, 2009. – 524 с.*
7. *Волк М.А. Методы и средства распределенного имитационного моделирования электронных систем. Диссертация канд. техн. наук / Волк М.А. – Х.: ХНУРЭ, 1999. – 132 с.*
8. *Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб.: Питер, 2008. – С. 37.*
9. *Волк М.А. Процессное представление состояний распределенных имитационных моделей с учетом специфики их программной реализации / М.А. Волк // Вісник НТУ "Харківський політехнічний інститут". Збірник наукових праць. Тематичний випуск: Інформатика і моделювання. – Х.: НТУ „ХПІ", 2009. – №13. – С. 23-33.*

Поступила в редколлегию 18.01.2013

Рецензент: д-р техн. наук проф. С.С. Танянский, Харьковский национальный университет радиоэлектроники, Харьков.

ФОРМАЛІЗАЦІЯ ПРОЦЕСІВ РОЗПОДІЛЕНОГО ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

М.О. Волк, Т.В. Филимончук, Р.М. Грідель, Ал Шиблак Муаз

Пропонується формальний опис процесів в розподілених системах імітаційного моделювання. В якості елементів опису виступають активності, що відображають функціональність модулів моделюючого середовища, і дані, що реалізують ресурси моделей. Розглядається процес управління локальними і глобальними даними моделей. Вводиться поняття менеджера пам'яті моделей як інструменту уніфікації управління розподіленими моделями з метою підтримки стандарту розподіленого імітаційного моделювання HLA. Запропонований формальний опис може бути використано при створенні сучасних систем імітаційного моделювання інформаційних систем.

Ключові слова: *розподілена імітаційна модель, інформаційні системи, процес, активність, дані.*

FORMALIZE THE PROCESS OF DISTRIBUTED SIMULATION OF INFORMATION SYSTEMS

M.O. Volk, M. Al Shiblak, R.M. Gridel

The formal description of the processes in distributed simulation. As elements of the description present activities, that show functional modules simulation environment, and the data that implement resource models. The process of management of local and global data models. Introduce the concept of the memory manager models as tools unify the management of distributed models to support the distributed simulation standard HLA. The proposed formal specification can be used to create advanced systems simulation of information systems.

Keywords: *distributed simulation model, information systems, process, activity, data.*