УДК 004.89

МОДЕЛИ БАЗОВЫХ КОМПОНЕНТОВ СИСТЕМЫ ПРОЕКТИРОВАНИЯ АГЕНТНЫХ ИМИТАЦИОННЫХ МОДЕЛЕЙ И ИХ РЕАЛИЗАЦИЯ В ПРОГРАММНОМ КОМПЛЕКСЕ ADSKIT

Павлов Александр Иннокентьевич

К.т.н., с.н.с., e-mail: asd@icc.ru

Столбов Александр Борисович

К.т.н., м.н.с, e-mail: stolboff@icc.ru

Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской Академии наук, 664033, Иркутск, ул. Лермонтова, 134

Аннотация. В статье обсуждаются вопросы разработки моделей базовых компонентов, обеспечивающих функционирование системы проектирования агентных имитационных моделей Adskit (agent development support kit). Согласно используемому в статье подходу агентная имитационная модель рассматривается как программная система, обладающая специфической функциональностью, которую можно свести к множеству операций, которые связанны с жизненным циклом имитационной модели, коммуникацией между элементами модели; логическим выводом на основе баз знаний; организацией вызова внешних подпрограмм и т.п.. В статье приведены примеры программной реализации моделей в программном комплексе Adskit с использованием Java, Jess, Drools, Madkit.

Ключевые слова: многоагентные системы, имитационное моделирование.

Цитирование: Павлов А.И., Столбов А.Б. Модели базовых компонентов системы проектирования агентных имитационных моделей и их реализация в программном комплексе Adskit // Информационные и математические технологии в науке и управлении. 2018. №4 (12). С. 155–162. DOI: 10.25729/2413-0133-2018-4-16

Введение. Агентная имитационная модель (АИМ) относится к классу вычислительных моделей, основной отличительной особенностью которых является идея множественности активных и автономных элементов – агентов, находящихся и действующих в общей среде. Как правило, АИМ содержит разные типы агентов, методы принятия решений которых могут варьироваться от простых реактивных правил «ЕСЛИ ТО» до более сложных когнитивных поведенческих шаблонов, учитывающих целеполагание, коллективное планирование и самообучение. Наиболее эффективно применять АИМ для анализа систем с гетерогенно распределенными, автономными и проактивными элементами. В настоящее время наиболее популярно использование АИМ в таких областях как логистика, финансы, промышленность, биология, социальная сфера.

Согласно одному из последних обзоров [5], существует более 80 программных средств, которые могут быть использованы для разработки АИМ. В связи с этим одной из основных тенденций научного поиска в данном направлении является исследования по анализу и обобщению накопленного опыта при применении АИМ и соответствующего программного обеспечения. Результатами этих исследований являются метамодели и

методологии [6, 9, 12, 17 и др.], классифицирующие и унифицирующие существующие подходы. Таким образом, в настоящее время актуальной является тема создания программных комплексов, поддерживающих процесс разработки АИМ на более абстрактном уровне, что, в свою очередь, позволяет варьировать разные подходы агентного моделирования и повторно использовать существующие программные средства. Данная статья вносит вклад в теорию и практику создания таких комплексов, развивает и дополняет подход к созданию инструментального средства разработки агентных имитационных моделей (ИСР АИМ), который представлен в серии публикаций [2-4].

Согласно авторскому подходу вся информация, необходимая для разработки АИМ, формализуется в форме иерархического множества моделей. Более подробно множество моделей и последовательность трансформаций этих моделей с целью формирования спецификации конкретной прикладной АИМ описаны в статье [2]. В соответствии с предложенным подходом процесс исполнения АИМ осуществляется путем интерпретации её спецификации в некотором специализированном программном средстве. В качестве такого средства в настоящее время используется научный прототип ИСР АИМ – программный комплекс Adskit (agent development support kit). Основная цель данной статьи заключается в более подробном описании тех метамоделей, которые связаны с обеспечением непосредственной реализацией процесса моделирования на основе спецификации АИМ.

Особенности подхода Adskit . На практике конкретная реализация агентной имитационной модели может рассматриваться как некоторая программа, обладающая специфичным поведением и функционирующая в гетерогенной среде. Поэтому основная идея подхода Adskit – применение к процессу создания АИМ принципов модельно-управляемый разработки MDD (model-driven development) [8], который за последние годы активно применяется не только для решения проблем, связанных с непосредственной разработкой программного обеспечения, но также используется и в других смежных областях, например, в имитационном моделировании [7, 9]. Подход MDD предполагает обобщение специфичного поведения разрабатываемых программных систем в виде метамоделей и создание средств их обработки, на основе которых происходит построение конкретных моделей, описывающих проблемную ситуацию. Придерживаясь принципов MDD, в Adskit подходе осуществляется явное разделение информации на предметно-зависимую и предметно-независимую, которая, в свою очередь, может быть далее декомпозирован на проблемно-ориентированную, описывающую способы решения задач, и агентно-ориентированную, содержащую информацию о применяемой методологии агентного моделирования. В результате такого подхода к формализации информации, необходимой в процессе создания АИМ, формируется иерархическое множество моделей и метамоделей, создание и преобразование которых осуществляется в соответствии со следующим обобщенным алгоритмом:

1. Создание множества метамоделей М2, описывающих методологию разработки АИМ, включая состав элементов АИМ (среда, агент, сенсор, сообщение и т.п.) и их поведение (создать агента, отправить сообщение и т.п.).

2. Создание концептуальной модели предметной области ($М^{Ont\text{-}D}$) и баз знаний ($М^{KB\text{-}D}$), описывающих поведение объектов с содержательной точки зрения.

3. Формирование спецификации конкретной АИМ для рассматриваемой предметной области ($М^{A\text{-}D}$) за счет определения связи между метамоделями М2 и $М^{Ont\text{-}D}$, $М^{KB\text{-}D}$.

4. Установка начальных условий для конкретной АИМ на основе спецификации $M^{A\text{-}D}$.

5. Выполнение симуляции АИМ за счет интерпретации её спецификации с начальными условиями в программном комплексе Adskit.

С каждым пунктом обобщенного алгоритма связано несколько моделей или метамоделей, а также соответствующие преобразования [2]. Все модели описываются в единых терминах метамодели верхнего уровня M^{Ont}, представляющей собой онтологию вида «понятие-атрибут-отношение». Для целей текущего исследования интерес представляет первый пункт алгоритма, связанный с описанием поведения элементов АИМ и более подробно описываемый в следующем разделе.

Модели базовых компонент системы проектирования агентных имитационных моделей. В подходе Adskit нет жестко заданных примитивов, задающих какую-либо архитектуру многоагентной системы, например, как в работах [6, 9]. Вместо этого пользователям предоставляется возможность создавать свою собственную концептуальную модель методологии имитационного моделирования в терминах метамодели M^{Ont} и проектировать поведение её элементов в форме потока работ (workflow) [14, 15]. При этом описание такого предметно-независимого поведения можно в конченом счете свести к множеству операций, исполняемых в процессе выполнения АИМ и называемых далее *базовыми операциями*. Например, это могут быть операции, связанные с жизненным циклом имитационной модели, коммуникацией между элементами модели; логическим выводом на основе баз знаний; организацией вызова внешних подпрограмм; визуальным представлением процесса моделирования; обеспечением доступа к данным во время выполнения программы и т.п.

Для представления базовых операций и способов их комбинации на самом высоком уровне абстракции используется метамодель M^{Op}.

$$M^{Op}=<Op^{B}, Op^{C}>$$

<Op^{B}>=<Имя метода>{<Параметр>} <Реализация>

<Реализация>=<Адрес><Протокол доступа>

<Протокол доступа> = HTTP | SOAP | Websocket | Локальный вызов

<Параметр>=<Имя><Вход |Выход><Тип>

<Тип>= <Литерал>|<Понятие>

<Op^{C}>=<Имя> {<Параметр>} {<Op^{B}>|<Op^{C}>|<Оператор>{<Op^{B}>|<Op^{C}>}}

<Оператор>= O^{if} | O^{L}| O^{B}

Здесь <Op^{B}> – описание базовой операции, выполняемой в процессе имитационного моделирования; <Op^{C}> – описание композитной операции; <Адрес> – информация о физическом расположении программной реализации метода (например, URL сервера и ID клиента, путь к исполняемому файлу и т.п.); O^{if} – оператор ЕСЛИ; O^{L} – оператор цикла; O^{B} – оператор группировки операций. Метамодель M^{Op} является обновленной версией M^{Com}, описанной в статье [2].

В Adskit подходе реализация базовых операций в процессе имитационного моделирования должна быть осуществлена некоторыми программными модулями, которые по аналогии называются *базовыми компонентами*, а под моделями таких базовых компонентов понимаются программные интерфейсы. На данный момент предложено 3 модели (интерфейса) базовых компонент:

$$M^{Op\text{-}I} = <I^{sim}, I^{inf}, I^{run}>.$$

Интерфейс движка моделирования (I^{sim}) содержит набор операций, выполняемых имитационной программной средой: запуск/остановка процесса симуляции, продвижение модельного времени;. создание/запуск/удаление агентов; посылку широковещательных и адресных уведомлений; обработку входящих уведомлений и т.п.. На данный момент в I^{sim} не включаются операции, непосредственно обеспечивающие поведения сенсоров и эффекторов АИМ, так как их реализации в конкретных движках моделирования могут сильно отличаться дуг от друга, или вообще не поддерживаться. Вместо этого поведение сенсоров и эффекторов моделируется в форме композитных операций Op^C с использованием механизмов уведомлений, описанных в интерфейсе I^{sim}.

Интерфейс машины логического вывода (I^{inf}) предоставляет операции для интерпретации и вывода на основе базы знаний (БЗ), описанной в форме M^{KB-D}: загрузить базу знаний (в формате M^{KB}); загрузить начальное состояние БЗ (в форме экземпляра понятий M^{Ont}); запустить/приостановить/завершить логический вывод; получить список сработавших правил; получить текущее состояние рабочей памяти (в форме экземпляра понятий M^{Ont})

Интерфейс контроллера времени выполнения (I^{run}) предназначен для поддержки выполнения рутинных операций с элементами АИМ в процессе программной имитации модели. Все операции интерфейса I^{run} можно разделить на два подмножества: получение и изменение данных и работа с операциями, описанными в форме Op^B или Op^C (например, выполнить операцию, получить спецификацию операции и т.п.). Фактически благодаря интерфейсу I^{run} можно организовать вызовов любых внешних процедур, которые поддерживают спецификацию Op^B и обеспечивают возможность преобразования своих входов и выходов в терминах метамодели M^{Ont}.

На практике использование универсального механизма вызова интерфейса I^{run} наиболее востребовано при отправке текущих результатов моделирования внешним системам обработки информации (например, для визуализации данных). Для более сложных случаев подходящим представляется явное расширение состава базовых компонентов и соответствующие изменения концептуальных моделей из множества M2. Одним из вариантов развития модели M^{Op-I} является её дополнение проблемно-ориентированными методами, например, для решения задач перемещения в 2D и 3D среде, построение маршрутов, выбора альтернатив и т.п. Таким образом, в рамках предлагаемого подхода имеется потенциал поэтапной специализации программного комплекса Adskit как на различные предметные, так и проблемные области.

Примеры реализации базовых компонент. Согласно метамодели M^{Op} для реализации базовых операций могут быть использованы любые средства, поддерживающие необходимые протоколы. При создании научного прототипа Adskit выбор был сделан в пользу программного обеспечения, написанного на языке Java. Экосистема Java содержит развитую кроссплатформенную инфраструктуру, включая средства автоматизации для реализации протоколов доступа метамодели M^{Op}; большое количество библиотек и каркасов, поддерживающих различные этапы процесса разработки сложных программных комплексов, а также имеет большое сообщество профессиональных программистов. С точки зрения проблемной ориентированности на создание АИМ, Java платформа также обладает преимуществом, так как к настоящему времени разработано большое количество свободно

распространяемого программного обеспечения, которое можно использовать для реализации моделей базовых компонент.

Интерфейс движка моделирования (I^{sim}) реализован с использованием библиотеки среды мультагентного моделирования Madkit [10]. Ключевым элементом этой реализации является специально разработанный класс «Типовой агент», который является потомком встроенного Madkit класса «Абстрактный агент», предназначенного для выполнения синхронного типа агентного взаимодействия. При этом в типовом агенте создаются специальные методы, ответственные за реализацию на каждом шаге моделирования соответствующих операций из множества I^{sim}. Также с использованием библиотеки Madkit разработан специальный планировщик, в качестве которого используется агент асинхронного типа, который в рамках своего жизненного цикла обеспечивает управление процессом моделирования на макро уровне.

В настоящее время существуют как коммерческие, так и свободно распространяемые оболочки продукционных экспертных систем (JESS, G2 GENSYM, CLIPS, OPS5, DROOLS и др.), с помощью которых можно реализовать интерфейс машины логического вывода (I^{inf}). В Adskit были использованы системы JESS [11] и Drools [16], так как оба эти средства написаны на языке Java, что упрощает процесс их интеграция в программный комплекс. Кроме того важным фактором в выборе именно этих средств стала возможность их бесплатного использования в академических целях. Трансформация БЗ из обобщенного вида $M^{KB\text{-}D}$ и M^{Ont} в форматы Jess и Drools (диалект mvel) основана на модуле преобразования правил и фактов, описанном в статье [1].

Для реализации интерфейса контроллера времени выполнения (I^{run}) необходимо решить задачу унификации обмена данными и обеспечить возможность доступа к произвольным внешним процедурам в соответствии с протоколами доступа, указанным в метамодели M^{Op}. Для решения этой задача был использован опыт разработки инструментального средства создания систем, основанных на знаниях. В работе [13] была предложена концепция унифицированного интерфейса компонента – IComponent, обеспечивающего возможность формировать прикладные системы на основе интеграции проблемно-ориентированных средств. Интерфейс IComponent позволяет получать метаописания компонента в терминах M^{Ont} за счет чего достигается возможность управления его состоянием и поведением в единообразном виде. Применение подхода на основе интерфейса IComponent позволяет организовать процесс добавления внешнего программного модуля в комплекс Adskit на основе следующего обобщенного алгоритма:

- Выбор проблемно-ориентированной программной системы, обладающей нужной функциональностью.
- Реализация интерфейса IComponent путем разработки программной оболочки для выбранной системы с использованием шаблона проектирования «фасад».
- Интерпретация описания свойств и методов компонента в терминах M^{Ont}.

Для интерпретации композитных операций Op^{C} разработан специальный модуль Adskit, который реализует логику поддерживаемых операторов метамодели M^{Op}, а для выполнения операций Op^{B} использует описанные выше базовые компоненты. В настоящее время модуль реализован в виде Java программы.

Заключение. В статье рассмотрены модели предметно-независимых операций, используемых для проектирования и реализации АИМ в рамках авторского подхода к

созданию инструментального средства разработки АИМ. Описана метамодель операций базовых компонент, приведены примеры этих операций, рассмотрены особенности программной реализация компонент. Явное разделение спецификации АИМ на предметно-независимую и предметно-зависимую части позволяет формализовать методологию разработки АИМ с использованием предложенных в статье моделей базовых компонент. В соответствии с Adskit подходом полученное таким образом описание методологии может быть повторно использовано для агентного имитационного моделирования различных предметных областей.

Работа выполнена при частичной финансовой поддержке РФФИ (грант № 18-07-01164, 18-08-00560).

СПИСОК ЛИТЕРАТУРЫ

1. Николайчук О.А., Павлов А.И., Коршунов С.А. Web-ориентированный компонент продукционной экспертной системы// Программные продукты и системы. 2015. №2. С. 20–25. DOI:10.15827/0236-235X.110.020-025.
2. Николайчук О.А., Павлов А.И., Столбов А.Б. Особенности разработки агентных имитационных моделей на основе модельно-управляемого подхода // Тр. восьмой всерос. науч.-практической конф. по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» (Санкт-Петербург, 18–20 октября 2017 г.). 2017. С. 288–293.
3. Павлов А.И., Столбов А.Б. Архитектура системы поддержки проектирования агентов для имитационных моделей сложных систем // Программные продукты и системы. 2015. №1. С. 12–16. DOI:10.15827/0236-235X.109.012-016.
4. Павлов А.И., Столбов А.Б. Прототип системы поддержки проектирования агентов для имитационных моделей сложных систем // Программные продукты и системы. 2016. №3. С. 79–84. DOI: 10.15827/0236-235X.115.079-084.
5. Abara S., Theodoropoulos G.K., Lemarinier P., O'Hared G.M.P. Agent Based Modelling and Simulation tools: A review of the state-of-art software // Computer Science Review. 2017. Vol. 24. Pp. 13–33.
6. Beydoun G., Low G., Henderson-Sellers B., Mouratidis H., Sanz J.G., Pavon J., Gonzalez-Perez C. FAML: A generic metamodel for MAS development // IEEE Transactions on Software Engineering. 2009. Vol. 35, №6. Pp. 841–863.
7. Cetinkaya D., Verbraeck A., Seck M.D. Model continuity in discrete event simulation: A framework for model-driven development of simulation models // ACM Trans. Model. Comput. Simul. 2015. Vol. 25. №3. Article 17.
8. France R., Rumpe B. Model-Driven Development of Complex Software: A Research Roadmap // Proc. Of the Intern. Conf. «Future of Software Engineering» (USA, Minneapolis, 23–25 May 2007). 2007. Pp. 37–54.
9. Garro A., Russo W. EasyABMS: A domain-expert oriented methodology for agent-based modeling and simulation // Simulation Modelling Practice and Theory. 2010. Vol. 18. №10. Pp. 1453–1467.
10. Gutknecht O., Ferber J. The madkit agent platform architecture // Lecture Notes in Computer Scienc. 2000. Vol. 1887. Pp. 48–55.

11. Jess: the Rule Engine for the Java Platform. Available at: http://www.jessrules.com/jess/docs/71/ (accessed: 1.07.2018).
12. Klügl F., Davidsson P. AMASON: Abstract Meta-model for Agent-Based SimulatiON // Lecture Notes in Computer Science. 2013. Vol. 8076. Pp. 101–114.
13. Nikolaychuk O.A., Pavlov A.I., Stolbov A.B. The software platform architecture for the component-oriented development of knowledge-based systems // Proceedings of MIPRO 2018 (Opatija, Croatia, 21 – 25 May 2018), edited by Karolj Skala. 2018. Pp. 1234–1239.
14. Pereira J.L., Silva D. Business Process Modeling Languages: A Comparative Framework // New Advances in Information Systems and Technologies. Advances in Intelligent Systems and Computing. 2016. Vol. 444. Pp. 619–628.
15. Russell N., Hofstede A.H.M., Edmond D., Aalst W.M.P. Workflow Data Patterns. Queensland University of Technology. Brisbane. 2004. 75 p.
16. Salatino M., De Maio M., Aliverti E. Mastering JBoss Drools 6. Birmingham: Packt Publishing Ltd. 2016. 330 p.
17. Siegfried R. Modeling and Simulation of Complex Systems: A Framework for Efficient Agent-Based Modeling and Simulation. Berlin: Springer. 2014. 233 p.

UDK 004.891.2

MODELS OF THE BASIC COMPONENTS FOR THE AGENT-BASED SIMULATION MODELS DEVELOPMENT SYSTEM AND THEIR IMPLEMENTATION IN THE ADSKIT SOFTWARE PACKAGE

Alexander I. Pavlov
PhD, senior researcher, e-mail: asd@icc.ru
Alexander B. Stolbov
PhD, junior researcher, e-mail: stolboff@icc.ru
Matrosov Institute for System Dynamics and Control Theory Siberian Branch
of the Russian Academy of Sciences 134, Lermontov Str., 664033, Irkutsk, Russia

Abstract. The article discusses the models of the basic components that provide the software system operating for agent-based simulation models development – Adskit (agent development support kit). According to the approach utilized in the article, the agent-based simulation model is considered as a software with specific functionality that can be reduced to a variety of operations related to the life cycle of the simulation model, communication between the elements of the model; inference based on knowledge bases; calls to external subroutines, etc.. The article provides examples of the software implementation of discussed models in the Adskit system with the help of Java, Jess, Drools, Madkit.

Keywords: agent-based simulation modeling, model driven approach.

References

1. Nikolaychuk O.A., Pavlov A.I., Korshunov S.A. Web-orientirovannyj komponent produkcionnoj jekspertnoj sistemy [Web-oriented component of an expert system] // Programmnye produkty i sistemy = Software & Systems. 2015. №2. Pp. 20–25. DOI:10.15827/0236-235X.110.020-025. (in Russian).

2. Nikolaychuk O.A., Pavlov A.I., Stolbov A.B. Osobennosti razrabotki agentnyh imitacionnyh modelej na osnove model'no-upravlyaemogo podhoda [Towards agent-based simulation models development utilizing model-driven approach] // Proceedings of the 8th Russian scientific conference on simulation and its application in science and industry "Simulation. Theory and Practice» (Saint-Petersburg, 18-20 October 2017). 2017. Pp. 288–293. (in Russian).
3. Pavlov A.I., Stolbov A.B. Arhitektura sistemy podderzhki proektirovaniya agentov dlya imitacionnyh modelej slozhnyh sistem [Architecture of agents design support system for complex systems simulation models] // Programmnye produkty i sistemy = Software & Systems. 2015. №1. Pp. 12–16. DOI:10.15827/0236-235X.109.012-016. (in Russian).
4. Pavlov A.I., Stolbov A.B. Prototip sistemy podderzhki proektirovaniya agentov dlya imitacionnyh modelej slozhnyh sistem [A prototype of an agents design support system for complex system simulation models] // Programmnye produkty i sistemy = Software & Systems. 2016. No. 3. Pp. 79–84. DOI: 10.15827/0236-235X.115.079-084. (in Russian).
5. Abara S., Theodoropoulos G.K., Lemarinier P., O'Hared G.M.P. Agent Based Modelling and Simulation tools: A review of the state-of-art software // Computer Science Review. 2017. Vol. 24. Pp. 13–33.
6. Beydoun G., Low G., Henderson-Sellers B., Mouratidis H., Sanz J.G., Pavon J., Gonzalez-Perez C. FAML: A generic metamodel for MAS development // IEEE Transactions on Software Engineering. 2009. Vol. 35. №6. Pp. 841–863.
7. Cetinkaya D., Verbraeck A., Seck M.D. Model continuity in discrete event simulation: A framework for model-driven development of simulation models // ACM Trans. Model. Comput. Simul. 2015. Vol. 25. №3. Article 17.
8. France R., Rumpe B. Model-Driven Development of Complex Software: A Research Roadmap // Proc. Of the Intern. Conf. «Future of Software Engineering» (USA, Minneapolis, 23–25 May 2007). 2007. Pp. 37–54.
9. Garro A., Russo W. EasyABMS: A domain-expert oriented methodology for agent-based modeling and simulation // Simulation Modelling Practice and Theory. 2010. Vol. 18, №10. Pp. 1453–1467.
10. Gutknecht O., Ferber J. The madkit agent platform architecture // Lecture Notes in Computer Scienc. 2000. Vol. 1887. Pp. 48–55.
11. Jess: the Rule Engine for the Java Platform. Available at: http://www.jessrules.com/jess/docs/71/ (дата обращения: 1.07.2018).
12. Klügl F., Davidsson P. AMASON: Abstract Meta-model for Agent-Based SimulatiON // Lecture Notes in Computer Science. 2013. Vol. 8076, Pp. 101–114.
13. Nikolaychuk O.A., Pavlov A.I., Stolbov A.B. The software platform architecture for the component-oriented development of knowledge-based systems // Proceedings of MIPRO 2018 (Opatija, Croatia, 21 – 25 May 2018), edited by Karolj Skala. 2018. Pp. 1234–1239.
14. Pereira J.L., Silva D. Business Process Modeling Languages: A Comparative Framework // New Advances in Information Systems and Technologies. Advances in Intelligent Systems and Computing. 2016. Vol. 444. Pp. 619–628.
15. Russell N., Hofstede A.H.M., Edmond D., Aalst W.M.P. Workflow Data Patterns. Queensland University of Technology, Brisbane. 2004. 75 p.
16. Salatino M., De Maio M., Aliverti E. Mastering JBoss Drools 6. Birmingham: Packt Publishing Ltd. 2016. 330 p.
17. Siegfried R. Modeling and Simulation of Complex Systems: A Framework for Efficient Agent-Based Modeling and Simulation. Berlin: Springer. 2014. 233 p.