

MODELS AND SOFTWARE FOR AGENT-BASED MODEL DEVELOPMENT BASED ON MODEL-DRIVEN APPROACH

Olga A. Nikolaychuk⁽¹⁾, Alexander I. Pavlov⁽¹⁾, Alexander B. Stolbov⁽¹⁾

⁽¹⁾ Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk,
Russia

The article discusses the application of the model-driven approach in the development of agent-based simulation models, presents an overview of agent architectures, considers the updated version of the hierarchical system of metamodels for formalizing the agent-based simulation models development methodology, as well as a description of the minimum set of software tools required to develop the software system for creation the applied agent-based simulation models..

Keywords: agent-oriented systems, agent architecture, model-driven approach, agent-based simulation models.

МОДЕЛИ И ПРОГРАММНЫЕ СРЕДСТВА РАЗРАБОТКИ АГЕНТНЫХ МОДЕЛЕЙ НА ОСНОВЕ МОДЕЛЬНО-УПРАВЛЯЕМОГО ПОДХОДА

Николайчук О.А.⁽¹⁾, Павлов А.И.⁽¹⁾, Столбов А.Б.⁽¹⁾

⁽¹⁾ Институт динамики систем и теории управления им. В.М. Матросова СО РАН, г.
Иркутск

В статье рассмотрены особенности использования модельно-управляемого подхода при разработке агентных имитационных моделей, приведен обзор агентных архитектур, представлена обновленная версия иерархической системы метамodelей для формализации методологии создания агентных имитационных моделей, а также описание минимально необходимого набора программных средств, необходимых для создания инструментального средства разработки прикладных агентных имитационных моделей.

Ключевые слова: агентно-ориентированные системы, архитектура агента, модельно-ориентированный подход, агентные имитационные модели.

Введение. Мультиагентный подход опирается на идею агентов – активных сущностей, координирующих свои действия друг с другом с целью поиска решения таких задач, которые не могут быть найдены ими индивидуально за счет собственных возможностей. В настоящее время применение мультиагентного подхода в качестве методологии решения различных сложных задач очень популярно и охватывает множество различных предметных областей (логистика, финансы, промышленность, биология, социальная сфера и др.). Такая популярность вызвана существенным развитием возможностей информационно-коммуникационных технологий, которое было достигнуто за последние 20 лет, а также постепенным накоплением опыта при применении

многоагентного подхода. Количество программных средств, предназначенных для разработки мультиагентных систем (МАС) и агентных имитационных моделей (АИМ), как частный случай МАС, измеряется десятками. Среди всего этого разнообразия можно выделить самые популярные: Anylogic, JADE, Repast, Netlogo, Swarm и др., но все же количество предлагаемых инструментов достаточно велико. В этой ситуации все более актуальной является проблема расширения области применения уже существующих программных средств. В современных исследованиях в этом направлении предлагаются более абстрактные механизмы построения МАС и АИМ, позволяющие повторно использовать и комбинировать различные технологии разработки путем формального описания основных элементов и преобразований, применяемых на этапах проектирования и реализации моделей и систем.

Модельно-управляемый подход при разработке МАС и АИМ. К настоящему времени разработано большое количество вариантов методологий агентно-ориентированной программной инженерии: AUML, MASE, MASSIVE, SODA, GAIA, MESSAGE, INGENIAS, BDIM, HLIM, MEI, PROMETHEUS, PASSI, ADELFE, COMOMAS, MAS-CommonKADS, CASSIOPEIA, TROPOS и др. Каждая методология предлагает свой набор базовых элементов и инструкций, а также в некоторых случаях и программное инструментальное средство, для создания МАС. Все это разнообразие подходов, в свою очередь, привело к необходимости объединения и обобщения методологий проектирования: метамодель FAML [1], абстрактная метамодель AMASON [2], методология MOBMAS [3], декларативные спецификации МАС [4] и др.

На практике конкретная реализация многоагентной системы может рассматриваться как некоторая программа, обладающая специфичным поведением и функционирующая в гетерогенной среде. Следовательно, к процессу разработки МАС и АИМ, как частного случая, можно применять широко известный подход модельно-управляемой разработки (MDD – Model Driven Development) [5]. Подход MDD предполагает обобщение специфичного поведения разрабатываемых программных систем в виде метамodelей и создание средств их обработки, на основе которых происходит построение конкретных моделей, описывающих проблемную ситуацию. При этом непосредственное создание программного обеспечения для решения рассматриваемой проблемы осуществляется автоматически без участия разработчика. В области применения MDD подхода к процессу разработки МАС получено достаточно много результатов, обзор которых приведен, в частности в работе [6].

Направление, связанное с созданием унифицированных методологий разработки АИМ, в настоящее время развивается менее динамично, чем МАС. Во-первых, это связано с тем, что кроме специфики агентного подхода, необходимо также учитывать особенности имитационного моделирования. Во-вторых, и что по мнению авторов наиболее существенно, основными пользователями АИМ являются специалисты-предметники, не обладающие навыками программирования и проектирования программных систем. В-третьих, АИМ – это как правило модель, требующая для её создания слаженной работы экспертов из разных областей, и, следовательно, в процессе разработки АИМ необходимо учитывать междисциплинарность исследования и решать проблемы согласования знаний экспертов. Тем не менее, в настоящее время разрабатываются несколько обобщенных методологий создания АИМ [7], в том числе применяющих MDD подход [8].

Эта работа также относится к данному направлению исследований и развивает идеи ранее предложенные в [9, 10, 11]. В основе авторской методологии к созданию АИМ находится четырёхуровневая система моделей и метамodelей, обеспечивающая представление информации о структуре и поведении элементов АИМ в форме онтологии (или концептуальной модели), включая как общее описание архитектуры АИМ, так и спецификацию конкретной АИМ. При этом для описания предметной специфики используется формализм продукционных правил и парадигма визуального программирования. В текущей работе представлено краткое описание обновленной версии иерархической системы метамodelей, включая обзор агентных архитектур, и приведено описание минимально необходимого набора программных средств, необходимых для создания инструментального средства разработки прикладных агентных имитационных моделей (ИСП АИМ).

Особенности разработки иерархической системы моделей и метамodelей. Основной задачей при разработке системы моделей является явное отделение информации, непосредственно связанной с предметной спецификой, от аспектов, относящихся к методологии создания АИМ. На основе анализа существующих подходов и методологий можно выделить следующие виды архитектур [12, 13]:

- архитектура, которая базируется на принципах и методах искусственного интеллекта, т.е. систем, основанных на знаниях (deliberative agent architecture”, “архитектура разумного агента”);
- архитектура, основанная на поведении (reactive architecture) или “реактивная архитектура”;
- гибридная архитектура, имеющая те или иные черты от архитектур вышеуказанных видов.

С точки зрения структурной организации выделяют: одноуровневую и многоуровневую архитектуры. Одноуровневая архитектура используется в простых приложениях, многоуровневая – отражает иерархию уровней, реализующих различные аспекты функциональности. Наиболее часто в архитектуре агента присутствуют уровни, ответственные за [13]:

- восприятие и исполнение действий,
- реактивное поведение,
- локальное планирование,
- кооперативное поведение,
- моделирование,
- формирование намерений, и обучение агента.

Многоуровневая архитектура имеет горизонтальную или вертикальную организацию, при этом в первом случае все уровни имеют доступ к уровню восприятия и исполнения, во-втором – только один из уровней.

Для того, чтобы обеспечить формализацию такого разнообразного набора методологий необходимо иметь возможность их описания на максимально более обобщенном уровне. В данной работе для это используется метаметамодель M^{Ont} в виде хорошо известной онтологической формы: концепт – атрибут – отношение и метамodelи средств реализации, описывающие, например, интерфейсы средства моделирования,

машины логического вывода, блока вызова внешних процедур и т.п. Использование стандартизированного набора интерфейсов позволяет описывать различные методологии проектирования АИМ на основе единого языка спецификаций, абстрагируясь от подробностей программной реализации конкретных моделей.

Предполагается, что описание структуры и поведения элементов предметной области должно быть максимально упрощено и понятно специалистам-предметникам. В данной работе это достигается за счет использования подходов, применяемых в системах, основанных на знаниях. Так, для описания структуры используются онтологии или концептуальные модели, а для поведения – продукционные базы знаний.

В результате, с учетом описанных выше соображений, будем иметь четырехуровневую архитектуру моделей и метамodelей следующего вид:

- M0 – имитационная модель во время выполнения;
- M1 – спецификация АИМ;
- M2 – спецификация элементов M1 согласно выбранной методологии разработки АИМ (архитектуры АИМ);
- M3 (M^{Ont}) – определяет наиболее абстрактные элементы для спецификации M2.

К уровню M2 относятся следующие модели: M^{KB} – метамodelь продукционной базы знаний в терминах M^{Ont} ; M^{Op} – метамodelь для описания спецификаций алгоритмов; M^{Op-I} – метамodelь для описания средств реализации агентных имитационных моделей (движок моделирования, машина логического вывода, подсистем визуализации и т.д.) на высоком уровне абстракции в терминах M^{Op} ; M^{St-A} – метамodelь для описания независимой от предметной области структуры элементов АИМ (агент, среда, событие и т.д.) в терминах M^{Ont} ; M^{Op-A} – метамodelь для описания независимого от предметной области поведения АИМ; M^A – предметно-независимая, агентно-ориентированная метамodelь для описания некоторой методологии разработки АИМ, включая как структурные, так и поведенческие аспекты.

На уровне M1 находятся следующие модели: M^{Ont-D} – онтология рассматриваемой предметной области; M^{KB-D} – база знаний, описывающая поведение объектов рассматриваемой предметной области; M^{St-A-D} – структура конкретной АИМ для рассматриваемой предметной области; M^{A-D} – спецификация конкретной АИМ для рассматриваемой предметной области; M^{A-D-I} – спецификация начальных условий конкретной АИМ;

К уровню M0 относится *codI* – результаты выполнения спецификации АИМ с учетом начальных условий M^{A-D-I} с помощью интерпретатора спецификаций АИМ.

Программные средства ИСР АИМ. Инструментальное средство проектирования прикладных агентных имитационных моделей развивает и обобщает идеи сформулированные при реализации прототипа системы поддержки проектирования агентов для имитационных моделей сложных систем [10] и программной платформы создания систем, основанных на знаниях [14]. Инструментальное средство включает в себя следующие компоненты.

1. Веб-редактор онтологий [15] позволяет создавать метамodelи и модели в терминах M^{Ont} .

2. Веб-редактор баз знаний [16] обеспечивает возможность визуального конструирования правил в терминах МКВ, формирования начальных условий, генерацию кода в формате JESS и DROOLS.
3. Компонент управления данными обеспечивает единообразный механизм для хранения, модификации и поиска информации, который используется другими компонентами инструментального средства. При программной реализации компонента использована СУБД PostgreSQL 9+.
4. Веб-редактор предметно-независимого поведения агентной модели в терминах M^{Op-I} . Для визуального представления используется семантическая сеть, где вершины – это методы и их параметры, а ребра показывают потоки данных и управления.
5. Веб-редактор агентной имитационной модели непосредственно реализует процесс разработки спецификаций и обеспечивает все необходимые при этом действия (хранение промежуточной информации, преобразование моделей, вызов других компонентов).
6. Веб-редактор управления вычислительным экспериментом позволяет формировать начальные условия для имитационной модели, а также управлять процессом симуляции.
7. Интерпретатор спецификаций АИМ обеспечивает непосредственное исполнение агентной имитационной модели. Для реализации функциональности, связанной с агентной спецификой, используется Madkit [17]; для логического вывода – Drools и Jess; для визуализации результатов – авторская разработка [18].

Заключение. Применение предлагаемых в статье моделей и средств позволяет решить проблему расширения практики применения уже существующих программных инструментов и унификации разнообразных методологий проектирования АИМ. Одной из отличительных особенностей предлагаемой иерархической системы моделей является возможность явного разделения спецификации на две части: агент-ориентированную и предметно-ориентированную. Эта позволяет явно описывать различные аспекты методологий, комбинировать их и повторно применять при создании конкретных предметных агентных моделей.

Основным направлением будущей работы является создание различных систем поддержки для разработки АИМ. Одним из самых простых вариантов может быть система для автоматизированного тестирования гипотез о строении и поведении элементов агентной модели. Например, для некоторой предметной области, для которой необходимо создать агентную модель, на основе многовариантных расчетов могут быть протестированы различные архитектуры агентов без изменения предметно-ориентированной части спецификации агентной модели.

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

Результаты получены при использовании сетевой инфраструктуры ЦКП «Интегрированная информационно-вычислительная сеть Иркутского научно-образовательного комплекса» (<http://net.icc.ru>).

- [1] 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. 35(6), pp. 841–863.
- [2] Klügl, F., Davidsson, P.: AMASON: Abstract Meta-model for Agent-Based Simulation. In: Klusch, M., Thimm, M., Paprzycki, M. (eds.) *MATES // LNCS*, 2013. vol. 8076, pp. 101–114.
- [3] Tran, Q., Low G.: MOBMAS: A methodology for ontology-based multi-agent systems development. // *Information and Software Technology* 2008. 50, pp. 697–722.
- [4] Challenger, M., Mernikb, M., Kardasa, G., Kosarb, T.: Declarative specifications for the development of multi-agent systems. // *Computer Standards & Interfaces* 2016. 43, pp. 91–115.
- [5] France R., Rumpe B. Model-Driven Development of Complex Software: A Research Roadmap // *Proc. Of the Intern. Conf. "Future of Software Engineering"*. Minneapolis, 2007. P. 37–54.
- [6] Kardas, G.: Model-driven development of multiagent systems: a survey and evaluation. // *The Knowledge Engineering Review* 2013. 28(4), pp. 479–503.
- [7] Siegfried, R.: *Modeling and Simulation of Complex Systems: A Framework for Efficient Agent-Based Modeling and Simulation*. Berlin: Springer, 2014.
- [8] Garro, A., Russo, W.: easyABMS: A domain-expert oriented methodology for agent-based modeling and simulation // *Simulation Modelling Practice and Theory* 2010. 18(10), pp. 1453–1467.
- [9] Павлов А.И., Столбов А.Б. Архитектура системы поддержки проектирования агентов для имитационных моделей сложных систем // *Программные продукты и системы*. 2015. №109. С. 12–16.
- [10] 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.
- [11] 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» (ИММОД-2017, 18-20 октября 2017 г., С-Петербург)*. 2017. С. 288-293.
- [12] Тарасов В.Б. Агенты, многоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте // *Новости искусственного интеллекта*. 1998, № 2. С.5-63.
- [13] Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы (обзор) // *Новости искусственного интеллекта*. 1998, №2. С. 64-116.
- [14] 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*, edited by Karolj Skala. Opatija, Croatia. 2018. pp. 1234–1239.
- [15] Редактор концептуальных моделей: а.с. 2016617626 Рос. Федерация.: А.И. Павлов, О.А. Николайчук, А.Б. Столбов; правообладатель ИДСТУ СО РАН. – Заявл. 18.05.2016; Опубл. 11.07.2016. 1 с.

- [16] Web-ориентированный редактор производственных правил: а.с. 2016663618 Рос. Федерация.: А.И. Павлов, О.А. Николайчук, А.Б. Столбов; правообладатель ИДСТУ СО РАН. – Заявл. 24.10.2016; Опубл. 13.12.2016.1 с.
- [17] Gutknecht, O. Ferber J. The madkit agent platform architecture // AGENTS 2000, LNCS / edited by Wagner, T., Rana, O.F. – Heidelberg: Springer, 2000. Vol. 1887. pp. 48-55.
- [18] Коршунов С.А., Николайчук О.А., Павлов А.И. Программное средство визуализации трехмерных сцен на основе онтологий // Научная визуализация. 2017. Т.9, №2. С. 102-119.