

УДК 681.3

Разработка и использование программных средств для построения и исследования агентных имитационных моделей

Е. Б. Замятина, Г. В. Чудинов

Пермский государственный университет, Россия, 614990, Пермь, ул. Букирева, 15
zamyatina@mail.ru; (342) 239-65-94

Рассматриваются вопросы разработки системы агентного имитационного моделирования. Именно агентное имитационное моделирование является удобным средством для решения некоторых классов задач. Подобные задачи часто возникают в сферах экономики и социологии. Кроме того, в статье затрагиваются вопросы описания и задания поведения отдельных агентов с помощью как DSL, так и средств инженерии знаний на примере экспертных систем.

Ключевые слова: имитационное моделирование; агентное моделирование; когнитивные агент; обучение.

Введение

В настоящее время агентное имитационное моделирование (АМ) и его методы находят все большее распространение. Известно, что многие практические задачи могут быть решены только с применением этих методов. Более того, имитация зачастую является единственным приемлемым способом исследования процессов, ситуаций или сложных динамических систем.

Однако традиционное дискретно-событийное моделирование (равно как и другие, часто применяемые методы: системная динамика, динамические системы [1]) обладает рядом недостатков. К ним следует отнести сложность добавления деталей к уже построенным моделям систем, их модификация и расширение в связи с изменением внешней среды. Обычно в этом случае требуется вносить корректировки в программный код и проводить повторную трансляцию.

Агентное имитационное моделирование предполагает, что агенты, моделирующие

исследуемые объекты, могут обучаться, приспосабливаться к изменению внешней среды и добиваться поставленной цели. В связи с этим АМ очень часто применяют для исследования поведения людей в аэропорту, на железнодорожном вокзале, для решения задачи цепочек поставок и т.д. Получается, что гибкость и адекватность АМ-системы повышается за счет введения когнитивных агентов, способных к адаптации.

Однако успешному применению АМ на практике мешают различные проблемы, связанные в первую очередь с инструментарием для разработки таких моделей. Существующие системы, как правило, достаточно трудны для овладения ими (Repast, Netlogo [2]). Кроме того, существуют сложности, возникающие с полным представлением моделей [3].

В этой статье предпринята попытка включить в уже разработанное авторами [4, 5] программное обеспечение компоненты для реализации когнитивных и адаптирующихся агентов и решить проблему описания модели и задания поведения индивидов. В AnyLogic [1], например, эта проблема решается так: описание структуры модели выполняется с помощью графического редактора, описание поведения объектов – с помощью UML-диаграмм.

Работа выполнена при финансовой поддержке грантов 08-07-90005 Бел_а, 08-07-90006 Бел_а, 10-01-00794-а

© Е. Б. Замятина, Г. В. Чудинов, 2010

Цели и предлагаемый подход

Итак, на кафедре математического обеспечения ВС ПГУ разрабатываются инструментальные средства для реализации АМ-системы «Агент-07». Предлагается использовать расширяемую библиотеку компонентов, построенную на основе .NET, и ядро для взаимодействия агентов и продвижения времени в модели. Данный подход открывает возможности для реализации сколь угодно сложных моделей. Расширяемость системы позволит обеспечить удобный интерфейс и базовые примитивы для построения моделей и проведения имитационных экспериментов.

Использование платформы .NET и языка С# в качестве основы для построения системы было обусловлено следующим:

- богатый выбор стандартных компонентов, а также возможность его наращивания с использованием любого удобного языка программирования, поддерживающего .NET;
- наличие удобных средств для удаленного взаимодействия (.NET Remoting), что позволяет добавить функциональность системы для распределенной работы в сети.
- безопасность, которую нам гарантирует использование средств .NET;
- многоплатформенность (реализация платформы .NET для операционных систем, отличных от Windows).

Описание компонентов АМ-системы «Агент-07»

Обычно системы АМ состоят из трёх частей:

- один или несколько агентов;
- среда для взаимодействия;
- правила, управляющие агентами.

Это деление было взято за основу при разработке инструментального средства для создания АМ. Структура разрабатываемой системы (ядро) и предполагаемое использование отображены на рис. 1.

Основой всей модели пользователя является *класс модели* (Model). В нем описана основная семантика поведения всей моделируемой системы в целом. Методы данного класса позволяют обеспечить вывод текущих результатов моделирования, а также запускать, останавливать, изменять основные параметры модели.

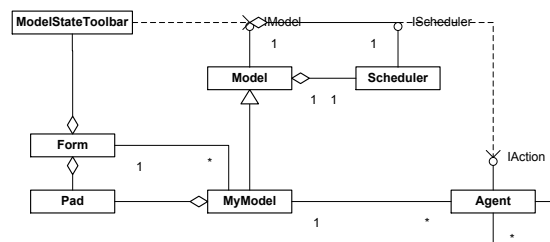


Рис. 1. Компоненты системы для построения АМ-моделей

Планировщик (Scheduler) отвечает за продвижение времени в системе (реализуя соответствующий интерфейс). Его главной задачей является определение тех агентов, которые получают право хода, т.е. фактически возможность выполнять активные действия в модели, что, собственно, составляет суть всего процесса моделирования. Подробнее о действиях сказано будет далее. Сейчас остановимся на базовом алгоритме работы планировщика. В качестве его используется традиционный алгоритм из дискретно-событийного моделирования с единственным центральным календарем событий. Планируется реализация распределенного моделирования и алгоритмов синхронизации.

Графический интерфейс представлен несколькими реализованными компонентами из разрабатываемой системы. Прежде всего, это планшет (Pad) для отображения различных графических объектов и привязки к ним агентов и сами графические объекты (рис. 2). Кроме того, к графическому интерфейсу относится *инспектор свойств* для графических объектов и самой модели и панель инструментов для управления состоянием выполнения модели. Все они рассчитаны на выполнения в многозадачной среде (то есть, безопасны при многопоточном выполнении). Рассмотрим подробнее эти компоненты.

Планшет используется для отображения графических объектов с учетом их порядка по оси OZ. Поддерживаются следующие функции: масштабирование, перемещение видимой области, перемещение графических объектов, их множественное и одиночное выделение, сжатие рабочей области планшета до реально занимаемой графическими объектами. В случае выхода графических объектов за границы рабочей области в любом из направлений планшет автоматически увеличивается в размерах. Также отметим, что данный элемент управления работает в многопоточном режиме.

Инспектор свойств используется для отображения свойств/параметров графических объектов (представляемых ими агентов) или модели. Им является компонент элемента управления PropertyGrid, который поддерживается обновление информации о выделенных объектах с продвижением имитационного времени модели и изменением состояния наблюдаемых агентов.

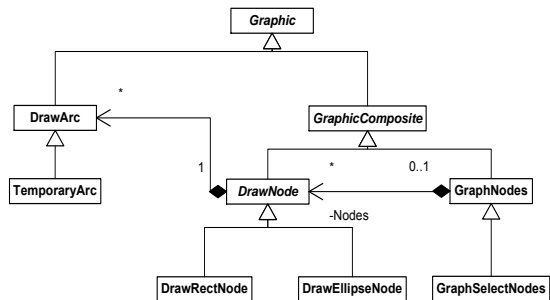


Рис. 2. Диаграмма классов для графических объектов

Извлечение информации из модели для долгосрочного хранения и анализа результатов выполнено с использованием инфраструктуры .NET. В будущем возможно создание более удобных узкоспециализированных классов для запоминания последовательных состояний модели, в том числе сети агентов и внутренних параметров агентов.

Тестовые задачи

Система «Агент-07» была апробирована на нескольких тестовых задачах. В частности, были рассмотрены задачи свободного передвижения и принятия решений в незнакомой обстановке, а также задача из области цепочек поставок.

Задача свободного передвижения и принятия решений в незнакомой обстановке (на рис. 3 – это вариация на тему поиска достопримечательностей в парке) заключается в моделировании человеческих действий, таких как восприятие мира, память, перемещение. Она может быть использована, например, для моделирования передвижения покупателей, впервые пришедших в магазин. Тогда можно проследить их маршруты движения и определить места для размещения ориентировочных указателей, а также товаров для продажи. Кроме того, отслеживаются временные показатели поиска и блуждания клиентов. Эти параметры могут быть оптимизированы в ходе имитационного эксперимента.

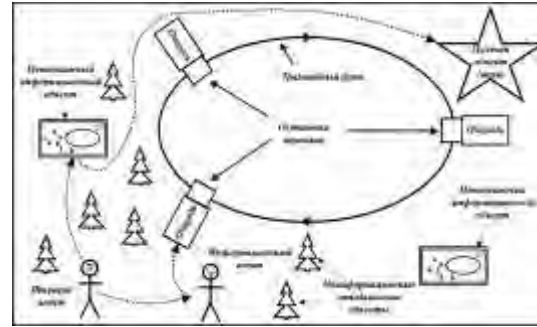


Рис. 3. Задача свободного передвижения и принятия решений (общий вид)

Решение приведенной выше задачи соответствует агентному подходу, при котором выделяются соответствующие классы агентов и объектов (возможно, пассивных агентов). Агентам соответствует покупатель, а также агентами могут считаться консультанты и другие покупатели, а объектам – стеллажи и полки в магазине, указатели и надписи отделов. Отметим, что здесь в первую очередь затрагивается пространственный аспект взаимодействия и обнаружения новых предметов, связанный со зрительным восприятием, а также передвижением и обходом препятствий.



Рис. 4. Действия, осуществляемые агентом цепочки поставок

Задача из области цепочек поставок (рис.4) тоже предполагает агентный подход к ее решению, однако здесь основной акцент сделан именно на связях между различными уровнями поставщиков и потребителей. А сама модель ориентирована на "социальное взаимодействие", в котором пространственные характеристики второстепенны по отношению к связям между участниками взаимодействия.

Решение подобных задач востребовано, например, в экономике, так как цепочки поставок возникают всякий раз, когда мы имеем дело с потребителями и производителями чего-либо (обычно товаров или иных материальных благ). Оптимизация подобной цепочки, определение ее параметров и узких мест и их устранение приносят ощутимый экономи-

ческий эффект, и, значит, обеим сторонам нужно взаимодействие. На рис. 4 изображено типичное поведение участника цепочки.

Когнитивный подход

В рассмотренных выше примерах использовались по большей части реактивные агенты, однако на практике их применение накладывает ряд существенных ограничений. Прежде всего, это отсутствие адаптируемости и обучаемости. В то же время все живые существа обладают этими свойствами, что заметным образом сказывается на их поведении.

Целью моделирования является получение адекватных действительности результатов (для чего необходимо как можно точнее отражать предметную область), поэтому разумным выбором в данной ситуации является использование когнитивных агентов. Они принимают решения с применением своего рода экспертной системы (ЭС). В простейшем случае ЭС и поведение агента управляются набором продукционных правил. В более сложных ситуациях возможно использование метаправил, а также методов символического обучения.

При разработке системы для реализации когнитивных агентов было решено выбрать продукционную парадигму. В итоге в систему был добавлен модуль, позволяющий описывать правила и метаправила поведения агентов, а также учитывать их при прогоне модели.

В правилах используются общедоступные свойства и методы программной реализации агентов. Метаправила позволяют управлять логическим выводом. Приведем пример правил и метаправил.

```

Rule «Критические запасы»
Group «Расчет заказа»
If склад.Запасы < критический
уровень >
Then заказ. Увеличить на
(недостающая величина)
Author «эксперт»
Rule «Метаправило»
Level MetaRule
If БЗ.Правила.Группа («Расчет
заказа»).Рассмотрены
Then
БЗ.ПерейтиКправилам().Группа («Оформ
ление заказа»)
Author «эксперт»
    
```

Здесь мы видим правило уровня предметной области, задающее поведение агента для заказа товара, и стратегическое метаправило, оптимизирующее процесс принятия решения.

Для описания подобных правил целесообразно идти по пути специализации системы, а по существу – к использованию языков, ориентированных на предметную область (DSL). В самом деле, DSL использует термины и выражения, понятные эксперту, что и позволяет эффективнее использовать (извлекать) его знания. Таким образом, внедрение DSL ведет к созданию более ясных программ и снижает их сложность.

Заключение

Итак, в статье представлена архитектура системы агентного моделирования. Использование когнитивных агентов упрощает построение адекватных расширяемых моделей любой степени сложности, что способствует распространению как самого АМ, так и его принципов.

Список литературы

1. *Борщёв А.* От системной динамики и традиционного ИМ к практическим агентным моделям: причины, технология, инструменты // www.gpss.ru
2. *Charles M. Macal C.M., North M.J.* Tutorial On Agent-Based Modeling And Simulation // Proceedings of the 2005 Winter Simulation Conference, Orlando, Florida, 2005. P.2–15.
3. *Castle C.J.E., Crooks A.T.* Principles and Concepts of Agent-Based Modelling for Developing Geospatial Simulations. Working papers series. Paper 110, 2006. P.60
4. *Замятина Е.Б., Чудинов Г.В.* Опыт разработки программных компонентов для реализации агентного моделирования // Современные проблемы механики и прикладной математики / Воронеж. ун-т. Воронеж: Научная книга, 2007. С.128–133.
5. *Чудинов Г.В.* Разработка компонентов системы для создания агентной имитационной модели // Новые информационные технологии. М.: МИЭМ, 2007. С.202.

Software for investigation of agent-based model: design and implementation

E. B. Zamyatina, G. V. Chudinov

Perm State University, Russia, 614990, Perm, Bukireva st., 15
zamyatina@mail.ru; (342) 239-65-94

This paper considers the problems of agent-based simulation system (ABMS). Namely agent-based simulation is a suitable and effective method to solve some classes of problems. These problems usually appear in economic and sociological domains. Besides, issues of declaration and definition single agent behavior, such as using DSL (Domain Specific Language) and knowledge engineering facilities by way of expert systems example, are touched upon in this article.

Key words: *agent-based simulation system; learning.*