

В. В. МИХАЙЛОВ, И. С. СЕЛЯКОВ, Т. Н. СОЛОВЬЕВА
**ПРИМЕНЕНИЕ МНОГОАГЕНТНОГО ПОДХОДА
ДЛЯ МОДЕЛИРОВАНИЯ БИОРЕСУРСОВ**

Рассмотрены особенности биоресурсов как объектов агентного моделирования. Представлен обзор основных подходов к построению многоагентных систем. Приведены примеры разработки моделей на базе стандартного пакета моделирования и специально разработанного симулятора.

Ключевые слова: биоресурсы, многоагентная система, архитектура интеллектуального агента, пакеты моделирования, симуляторы.

Введение. Многоагентный подход является относительно новым и чрезвычайно быстро развивающимся направлением имитационного моделирования. Подход состоит в представлении компонентов моделируемой системы в виде отдельных, относительно независимых объектов — интеллектуальных агентов, каждый из которых имеет свои цели и задачи. Агенты имеют возможность взаимодействовать друг с другом и окружающей средой, обмениваться

информацией для достижения общих целей. Каждый агент обладает собственными вычислительными ресурсами и действует параллельно во времени с другими агентами.

В методологии системного моделирования локальные компоненты, как правило, не имеют собственных целей функционирования вне глобальной цели развития системы в целом. Задача моделирования, таким образом, заключается в определении пути развития системы, когда внутрисистемные связи и параметры известны. При агентном подходе локальные компоненты являются относительно независимыми сущностями, а глобальные закономерности развития системы в целом определяются свойствами компонентов и их взаимодействием на локальном уровне.

Агентный подход с соответствующими средствами программной поддержки может применяться при произвольном количестве подсистем — интеллектуальных агентов. Однако в наибольшей мере его преимущества проявляются в случае, когда моделируемая система состоит из достаточно большого количества однородных по свойствам, относительно независимых компонентов. Именно в этом случае можно ожидать появления новых свойств и новых знаний о системе в целом.

Биоресурсы как объект моделирования. В литературе по природопользованию биоресурсы определяются как источники и предпосылки получения людьми необходимых материальных и духовных благ, заключенных в объектах живой природы. В широком плане к биоресурсам относят все средообразующие объекты биосферы — продуценты, консументы и редуценты с заложенной в них генетической информацией. В настоящей статье, для конкретности, ограничимся рассмотрением массовых промысловых видов крупных наземных млекопитающих (например, оленей, лосей, сайгаков и т.п.). Известно, что биологические виды существуют в форме популяций. Численность, половозрастная и пространственная структура популяции складываются под влиянием внутривидовых и внешних факторов. К последним относятся погодноклиматические условия, кормовая база, пищевые конкуренты и хищники, антропогенные воздействия. Как природный ресурс популяция может быть объектом промысла или спортивной охоты (лова). Структурными компонентами популяций являются семьи, стаи, стада, группировки животных. Эти компоненты характеризуются высокой сезонной изменчивостью. Тем не менее именно эти компоненты с учетом половозрастного состава, размещения в ареале, сроков и путей миграций определяют пространственную структуру популяции. Разработка имитационных моделей пространственного размещения и миграций животных чрезвычайно важна с ресурсных позиций. Однако общие закономерности динамики пространственной структуры популяций в биологии недостаточно исследованы и слабо формализованы. Именно здесь в наибольшей степени может быть полезен агентный подход, использующий локальные свойства объектов для выявления общих закономерностей. В качестве агентов могут служить отдельные особи или более крупные относительно стабильные образования — стада животных как компоненты группировок, группировки как компоненты популяции.

Целями многоагентного моделирования применительно к вопросам природопользования могут быть: проверка гипотез о закономерностях функционирования биосистем; исследование возможности устойчивого использования биоресурсов с учетом пространственного аспекта и адаптивных свойств особей; оптимизация промысловой системы с учетом пространственного размещения и миграций группировок животных и др.

В связи с многообразием пакетов моделирования и вариантов архитектур агентов построение агентных систем представляется достаточно сложной задачей, требующей проведения предварительного анализа возможных подходов и имеющихся программных средств.

Архитектуры интеллектуальных агентов. В современной теории мультиагентного моделирования выделяют четыре основные архитектуры интеллектуальных агентов [1, 2]:

— *логическую архитектуру*, где принятие решений реализовано на основе логических правил;

— *реактивную архитектуру*, где принятие решений реализовано за счет установления прямых связей между состояниями и действиями;

— *архитектуру BDI (belief—desire—intention)*, где принятие решений основано на манипуляции структурами данных, включающих представление агента (belief), его цели (intentions) и пути достижения целей (desires);

— *гибридную архитектуру*, где принятие решений реализовано через несколько программных слоев, каждый из которых представляет отдельный уровень абстракции.

Агентов, которые используют только точное представление картины мира в символьной форме, а решения (например, о действиях) принимают на основе формальных рассуждений и использования методов сравнения по образцу, принято определять как *делиберативные* (от англ. deliberation — обсуждение, рассуждение). Применительно к таким агентам используется термин „*логическая архитектура*“ [3]. Блок принятия агентом решений проверяет выполнимость логических утверждений, используя текущие данные базы знаний об окружающей среде. На основе результатов проверки выбирается нужное решение. Логическая архитектура проста и понятна, но может оказаться неэффективной ввиду сложности формирования модели среды в символьном виде.

Принципы *реактивной архитектуры* были выдвинуты Бруксом [4] как альтернативный подход к логической архитектуре. Архитектура Брукса основывается на двух принципах. Первый принцип заключается в том, что каждому состоянию окружающей среды, воспринимаемому блоком наблюдения, соответствует определенное действие. В отличие от делиберативного агента, реактивный агент не сохраняет данные о среде в базе знаний. Решение принимается мгновенно, исходя из входных данных. Второй принцип заключается в представлении поведения агента в виде отдельных уровней. Каждый уровень имеет определенный приоритет. При получении агентом входного сигнала сначала решаются наиболее важные и приоритетные задачи. К недостаткам реактивных агентов можно отнести их узкую специализацию для систем реального времени и невозможность применения в случае, когда от агента требуется сложный анализ целей, выбор поведения и прогнозирование ситуации.

Архитектура BDI [5, 6]. Поведение BDI-агентов напоминает итеративный процесс мышления человека при выборе нужного действия для достижения поставленных целей. Агенты с такой архитектурой способны не только оперативно реагировать на происходящее вокруг, но и анализировать свои цели и выбирать оптимальное поведение для их достижения. В отличие от логической и реактивной, BDI-архитектура позволяет агенту действовать „более интеллектуально“. К недостаткам можно отнести сложность реализации архитектуры и высокие требования к вычислительной способности отдельных агентов.

Попытки соединить классический логический и реактивный подходы привели к появлению разнообразных *гибридных архитектур* [7, 8]. Для реализации нескольких видов поведения агент представляется несколькими подсистемами, каждая из которых отвечает за определенный вид поведения. Основная проблема реализации гибридной архитектуры — создание адекватного поставленной агенту задаче и эффективного блока управления отдельными подсистемами для координации их действий и выбора общего решения.

Пакеты моделирования многоагентных систем. Среди многообразия современных пакетов агентного моделирования сложно выделить какой-то один универсальный и эталонный. Выбор пакета должен быть обусловлен типом решаемой задачи и квалификацией разработчика системы.

Пакет SWARM [9] позволяет моделировать сложные многоагентные системы, которые включают большое количество агентов. Для описания агентов используется внутренний язык наподобие языка С. Агенты имеют логическую архитектуру, что обеспечивает прозрачность

алгоритмов их поведения. Основной недостаток этого пакета — необходимость освоения нового языка программирования разработчиком системы.

Система моделирования и анализа мультиагентных систем Taems [10] основана на структуре задач и отношений между ними. Ее главное преимущество — возможность построения сложных интеллектуальных агентов, способных принимать решения для достижения поставленных целей, и наличие встроенных графических редакторов для представления внутренних блоков агента и модели его поведения. Однако система плохо документирована и имеет много ошибок в реализации, что усложняет действия разработчика.

Система AnyLogic [11] является многофункциональным графическим пакетом имитационного моделирования. Главное преимущество AnyLogic — широкие возможности по графическому представлению модели, которая создается путем графического соединения различных блоков из множества внутренних библиотек и дополнительных вставок на языке Java. Простота разработки моделей сопровождается, однако, недостаточной универсальностью пакета.

Инструментальная среда MASDK [12], разработанная в Санкт-Петербургском институте информатики и автоматизации РАН, предназначена для построения прикладных многоагентных систем. Главной особенностью данной среды является поддержка полного цикла разработки агентных систем, начиная с этапа описания предметной области и заканчивая моделированием с участием агентов с заданной структурой и поведением. Описание этой среды осуществляется с помощью специального языка AFW, созданного на основе XML. Для удобства разработчика в среде имеется набор графических редакторов, упрощающих процесс описания системы и ее структуры. В системе имеется так называемый „генератор агентов“, который на основе описания системы на языке AFW выполняет генерацию программного кода на языке C++.

Пакет JADE (Java Agent Development Framework) [13, 14] полностью поддерживает стандарт для мультиагентных систем FIPA [15], который является передовым в области многоагентного моделирования. Пакет написан на языке Java. Он позволяет создавать агентов, поддерживать их жизненный цикл, назначать агентам задачи, определять их поведение и т.п. Таким образом, пакет JADE реализует парадигму так называемого агентно-ориентированного программирования. В отличие от среды MASDK пакет JADE не дает возможности описать предметную область, а позволяет лишь создать реализацию агентной системы.

Создание моделей на языке Java требует от разработчика хороших навыков программирования, но при этом он не ограничен в своих возможностях и может создать гибкую эффективную систему с собственным графическим интерфейсом. Важным достоинством языка Java является многоплатформенность. Благодаря поддержке технологии RMI имеется возможность проводить распределенное моделирование многоагентной системы в компьютерной сети. При этом никакой дополнительной работы от программиста не требуется.

Формирование агентной модели и интерфейс пользователя. Стандартные пакеты моделирования. Наиболее простой вариант построения агентной модели состоит в непосредственном использовании возможностей стандартных пакетов моделирования. В зависимости от квалификации пользователь самостоятельно или с участием системного аналитика-программиста выполняет разработку модели и проводит имитационные эксперименты, используя средства программной поддержки и интерфейс выбранного пакета моделирования.

В качестве примера рассмотрим построение агентной модели поведения стадных животных с использованием пакета AnyLogic. Для моделирования многоагентных систем в пакете предусмотрено создание большого количества экземпляров классов, которые способны функционировать одновременно. Для удобства задания поведения объектов используются карты состояний и таймеры. Систему можно представить на экране в виде анимации. Разработчик модели может использовать все возможности объектно-ориентированного программирования модифицированного языка Java. Однако для придания модели более прозрачной

структуры код программы разбивается на блоки, а назначение переменных, задание связей и другие стандартные действия осуществляются графическими средствами. Рассмотрим построение агентной модели поведения стадных животных в пакете AnyLogic.

Объектом моделирования является стадо животных [16]. В зимний период стадо перемещается в пределах ограниченной области и состоит из 20—40 животных, находящихся в пределах видимости (менее 200 м). Бюджет времени оленей содержит 3 фазы: кормление, отдых и перемещение.

Фаза кормления продолжается 0,5—1,5 ч. При кормлении происходит циклическая смена двух процессов.

1) Фиксация животного в точке для раскапывания лунки в снегу и кормления (от 2 до 10 мин). При этом в одной точке может находиться только одно животное.

2) Микроперемещение на 10—15 м к новой фиксированной точке (около 5 мин). Направление перемещения — случайное. При перемещении животные оценивают расстояние до соседей в стаде. Если животное отходит на расстояние более 100 м, то вероятность движения по направлению от стада уменьшается, а вероятность движения по направлению к стаду увеличивается. Животное, отошедшее далее чем на 200 м, теряет связь со стадом. При этом оно может войти в зону притяжения другого стада и соединиться с ним.

Этап отдыха после кормления длится 1—2 ч. Постепенно животные прекращают кормление и ложатся.

Третий этап — перемещения. Он начинается после отдыха, но может быть и при кормлении. Здесь проявляется эффект лидерства. Одна из особей встает и начинает движение в некотором направлении. Затем начинают постепенно вставать другие животные и двигаться за лидером. Движение в результате происходит по общей тропе. В следующем цикле лидером может быть другая особь. После перемещения вновь начинается фаза кормления.

Для моделирования описанной системы в пакете AnyLogic было создано два класса объектов: Herd (стадо) и Deer (олень). Каждый объект класса Herd включает в себя некоторое число объектов класса Deer. Это число выбирается между 20 и 40 случайным образом каждый раз при запуске модели. Для задания поведения стада и особи для каждого класса были сформированы карты состояний (стейтчарты) и ряд функций для определения координат животного и его ближайших соседей.

Стейтчарт стада содержит два состояния: stop (стоп) и go (перемещение). Изначально стадо находится в состоянии stop. По истечении тайм-аута (2,5 ч) объект переходит в состояние go, и происходит случайный выбор лидера. В состоянии go случайно выбирается направление перемещения стада и лидеру посылается сигнал „вперед“. Через 90 мин (время, задаваемое согласно тайм-ауту) происходит возвращение стада в состояние stop и лидеру посылается сигнал „стоп“.

Поведение отдельного оленя описывается более сложной картой состояний. Она содержит гиперсостояние eat (фаза кормления) и два состояния — rest (отдых) и go. Гиперсостояние eat включает в себя состояния stop и walk (микроперемещение).

Для описания действий животных служат встраиваемые в графическую структуру модели вставки на языке Java.

Для обеспечения связи между соседними особями создана функция Nbh(m) (от англ. neighbourhood — соседство), используемая для поиска ближайшего соседа.

Переход между состояниями stop и walk осуществляется в соответствии со случайными тайм-аутами. Переход из состояния eat в состояние rest также осуществляется согласно тайм-ауту — случайному числу между 60-й и 90-й минутами. Переход из состояния rest в состояние go и из состояния go в состояние eat осуществляется по сигналам „вперед“ и „стоп“ соответственно. Кроме того, животное в состоянии go каждые 5 минут посылает ближайшему отдыхающему соседу сообщение „вперед“.

Таким образом, путем организации связей между особями с помощью сигналов реализуется стадность. В состояниях, соответствующих движению (walk и go), происходит пересчет текущих координат особи по специально разработанным алгоритмам.

Симуляторы. Если возможности стандартных пакетов недостаточны для эффективного решения поставленной задачи, то возникает задача разработки симулятора. Симуляторы служат для проведения имитационных экспериментов специалистами конкретной предметной области и разрабатываются на базе тех или иных пакетов моделирования, а также дополнительных программных средств для задания моделей окружающей среды и отображения размещения агентов. Для эффективного решения прикладных задач симулятор должен иметь удобный пользовательский интерфейс.

В качестве примера рассмотрим симулятор, разработанный авторами для моделирования природно-хозяйственной системы [17]. Ресурсной базой является популяция животных, мигрирующих в пределах ареала. Пути миграций зависят от кормовой базы и погодноклиматических условий. Использование ресурса заключается в изъятии части животных. Для отстрела животных организуются стационарные и мобильные промысловые пункты. Цель моделирования состоит в выборе мест размещения промысловых пунктов в зависимости от путей миграций группировок животных. В качестве агентов были выбраны отдельные стада как компоненты группировок животных. Второй тип агентов — стационарные и мобильные промысловые пункты. Была выбрана логическая архитектура многоагентной системы, подразумевающая наличие базы данных о состоянии агентов и окружающей среды.

Симулятор разработан с использованием средств пакета JADE ввиду отмеченных ранее достоинств пакета (гибкости, поддержки стандарта FIPA, многоплатформенности). Интерфейс симулятора содержит главное окно, которое включает панель управления моделированием, меню для задания параметров моделирования и рабочую область. Рабочая область служит для отображения перемещения агентов в пространстве в режиме настройки параметров агентной среды.

В режиме моделирования симулятор работает с геоинформационной системой ArcGIS [18]. ArcGIS служит для задания и визуализации модели окружающей среды, а также визуализации расположения агентов на конкретной территории с точной географической привязкой, обеспечивая графический интерфейс симулятора. Симулятор производит вычисления и обновляет данные о размещении агентов с использованием модели среды, полученной геоинформационной системой. Обмен данными в интегрированной системе осуществляется в реальном времени через Shape-файл.

Заключение. Агентный подход позволяет создать распределенную динамическую систему, основанную на взаимодействии локальных агентов с учетом их территориального размещения, половозрастных различий и, возможно, их генетической неоднородности. Учет этих свойств позволяет увеличить степень адекватности предлагаемой модели реальной природно-хозяйственной системе. В агентном подходе постулируется, что глобальные закономерности функционирования системы выводятся из свойств ее локальных компонентов. Однако вопрос о том, какое количество агентов должна содержать система и какими свойствами должны обладать агенты, не является до конца решенным ни в теории агентного моделирования, ни в теории систем.

Исследования, выполненные по данной тематике, проводились при финансовой поддержке Отделения нанотехнологий и информационных технологий РАН (проект № О–2.3/03).

СПИСОК ЛИТЕРАТУРЫ

1. Nwana H. S. Software agents: An overview // Knowledge Engineering Review. 1996. Vol. 11, N 3. P. 1—40.

2. Weiss G. Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA, 2001. P. 36—61
3. Genesereth M. R., Nilsson N. Logical Foundations of Artificial Intelligence. San Mateo, CA, 1987. 416 p.
4. Brooks R. A. A robust layered control system for a mobile robot // IEEE Journal of Robotics and Automation. 1986. P. 14—23.
5. Konolige K. A Deduction Model of Belief. San Mateo, CA, 1986. 230 p.
6. Rao A. S., Georgeff M. P. Modeling rational agents within a BDI-architecture // Proc. of Knowledge Representation and Reasoning. San Mateo, CA, 1991. P. 473—484.
7. Ferguson I. A. TouringMachines: An Architecture for Dynamics, Rational, Mobile Agents. Cambridge, MA, 1992. 205 p.
8. Muller J. A Cooperation Model for Autonomous Agents. Berlin, 1997. P. 245—260.
9. SWARM Specification [Электронный ресурс]: <<http://www.swarm.org>> (по состоянию на 01.11.2007).
10. Horling B., Lesser V. The Taems White Paper / Univ. of Massachusetts. 2004. 50 p.
11. Карпов Ю. Г. Введение в моделирование с использованием среды AnyLogic [Электронный ресурс]: <<http://www.xjtek.com>> (по состоянию на 01.12.2006).
12. Городецкий В. И., Карасев О. В. Технология разработки прикладных многоагентных систем в инструментальной среде MASDK // Тр. СПИИРАН. 2006. Т. 1, вып. 3. С. 11—32.
13. JADE Programmer's Guide [Электронный ресурс]: <<http://jade.tilab.com>> (по состоянию на 20.02.2007).
14. JADE Administrator's Guide [Электронный ресурс]: <<http://jade.tilab.com>> (по состоянию на 20.02.2007).
15. FIPA Specification [Электронный ресурс]: <<http://drogo.csel.stet.it/fipa/>> (по состоянию на 01.02.2007).
16. Михайлов В. В., Колпацников Л. А. Особенности пространственно-временной структуры таймырской популяции диких северных оленей // Актуальные проблемы природопользования на Крайнем Севере: Сб. Новосибирск: ИПЦ „Юпитер“, 2004. С. 18—34.
17. Михайлов В. В., Селяков И. С. Использование мультиагентного симулятора при моделировании распределенных систем // Изв. вузов. Приборостроение. 2008. Т. 51, № 11. С. 69—73.
18. ArcGIS Desktop [Электронный ресурс]: <http://esri.com/software/arcgis/about/desktop_gis.html> (по состоянию на 10.04.2008).

Сведения об авторах

- Владимир Валентинович Михайлов** — д-р техн. наук, доцент; СПИИРАН, кафедра вычислительных систем и сетей; E-mail: mwwsari@mail.ru
- Игорь Сергеевич Селяков** — аспирант; СПИИРАН, кафедра вычислительных систем и сетей; E-mail: seliakov@mail.ru
- Татьяна Николаевна Соловьева** — аспирант; Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра вычислительных систем и сетей; E-mail: famsol@yandex.ru

Рекомендована СПИИРАН

Поступила в редакцию
09.07.10 г.