УДК 519.876.2; 004.89

Глибовець А. М.

АГЕНТНО-БАЗОВАНЕ МОДЕЛЮВАННЯ

У статті розглянуто ключові методології агентно-базованого моделювання та основні застосунки агентно-базованого моделювання.

Ключові слова: агентне моделювання, агенти, агентно-базована модель, засоби розробки агентних моделей.

Вступ

Агентне моделювання добре застосовувати у випадку, коли занадто складно або неможливо формалізувати поведінку системи на глобальному рівні та не існує адекватної математичної моделі предметної галузі. Агентне моделювання знаходить все більше і більше застосування. Це пояснюється тим, що ми живемо у світі, який постійно ускладнюється. Наприклад, моделювання економічних ринків, у припущенні про ідеальні ринки, однорідних посередників і довготривалій рівновазі, дозволяло вирішити багато задач аналітично. Але ці припущення досить сильні і суттєво обмежують широкий клас задач дослідження. У нагоді тут стає агентне моделювання (АМ), яке сягає своїм історичним корінням складних адаптивних систем (САС) і принципу побудови систем знизу вгору.

Томасу Шеллінгу (Thomas Schelling) приписується розробка першої соціальної моделі, заснованої на агентах, в якій агенти представляли людей, а взаємодії агентів – соціальний процес (агенти адекватно відображали цей процес) [10]. Шеллінг застосував поняття клітинного автомата, щоб вивчити закономірності житлової сегрегації (поділу). Він сформулював питання про те, чи можливі закономірності в поділі поселень, якщо більшість індивідуумів насправді не мають расових забобонів. Модель Шеллінга продемонструвала, що гетто можуть розвиватися спонтанно. Крім того, Шеллінг показав, що закономірності, які можуть з'являтися, не обов'язково слідують або навіть не узгоджуються з цілями (прагненнями) окремих індивідуумів. Це важливе спостереження викликало інтерес і дало поштовх застосуванню АМ у цій області. (Зазначимо, що початкові моделі Шеллінга навіть не були комп'ютерними: агенти представляли собою монети, що пересувалися шахівницею).

Герберт Саймон (Herbert Simon), Нобелівський лауреат (піонер в галузі штучного інтелекту), розробив теорію «задоволення», щоб описати як спостерігається поведінка людей і організацій в реальному світі [7, с. 3]. Поведінкова економіка – це відносно нова область, яка об'єднує експериментальні результати психології та когнітивних аспектів прийняття рішень агентами. Саймон виявив «позитивні цикли зворотного зв'язку» і «збільшуючі чинники» як основу динамічних процесів швидкого експоненціального зростання в економіці. Позитивні цикли зворотного зв'язку можуть створювати процеси, що самопідтримуються і швидко переводять системи з початкової точки (стартової позиції) в інший стан.

Агентне моделювання також використовується в соціології. У роботі [9] наведено детальний огляд агентного підходу як основи моделювання соціального життя. Соціальне життя розглядається як взаємодії між адаптивними агентами, що впливають один на одного у відповідь на вплив, який був вчинений на них. Агентне моделювання застосовується для моделювання поведінки бактерій, їх взаємодії та самоорганізації колоній бактерій [1].

Отже дослідження методологій АМ є актуальною задачею.

Структура агентно-базованої моделі

Система моделюється як набір автономних об'єктів, що приймають рішення, які називають агентами. Кожен агент індивідуально оцінює конкретну ситуацію і приймає рішення на основі встановлених правил. Повторювані взаємодії між агентами є характерною особливістю АМ. Агентна модель складається з системи агентів і відносин між ними. Складніша агентна модель може містити різні навчальні техніки, дозволяючи проводити більш реалістичне вивчення та адаптацію [8].

Розрізняють два основні класи архітектур агента [6]:

- архітектура, яка базується на принципах і методах штучного інтелекту, тобто систем, заснованих на знаннях (deliberative agent architecture, «архітектура розумного агента»);
- архітектура, заснована на поведінці (reactive architecture) або «реактивна архітектура» (заснована на реакції системи на події зовнішнього світу).

Наразі серед розроблених архітектур не існує такої, про яку можна було б точно сказати, що вона є виключно поведінковою або заснована тільки на знаннях. Будь-яка з розроблених архітектур є, по суті, гібридною, і має ті чи інші риси від архітектур обох типів.

З іншого боку, незалежно від того, що лежить в основі формалізації парадигми, архітектури агентів класифікуються відносно типу структури, накладеної на функціональні компоненти агента і прийнятих методів організації взаємодії його компонент в процесі роботи. Як правило, архітектура агента організується у вигляді декількох рівнів. Відповідно до роботи [5], серед багаторівневих архітектур розрізняють горизонтальну організацію взаємодії рівнів і вертикальну.

З класичного погляду архітектура на основі знань – це така архітектура, яка містить символьну модель світу, представлену в явній формі, до того ж в якій прийняття рішень про дії, які повинні бути зроблені агентом, здійснюється на основі міркувань логічного чи псевдологічного типів. Такий агент може розглядатися як спеціальний випадок системи, заснованої на знаннях.

Ідея архітектури агента на основі знань у наш час вже вийшла за межі логічної парадигми подання та обробки знань. Є архітектури, які сповідують лінгвістичний підхід (на основі формальних граматик), а також такі, що намагаються використовувати наближені знання і правдоподібні міркування [2].

Архітектура на основі планування («плануючий агент») розглядається як альтернатива підходу архітектури, заснованої на знаннях. У цьому підході планування розглядалося як конструювання послідовності дії, яка, будучи виконаною, призводила б, у результаті, до досягнення бажаної мети. Простим прикладом архітектури такого роду є архітектура, в якій реакція агента на зовнішні події генерується скінченим автоматом.

Тільки найпростіші програми агентів можуть бути реалізовані за однорівневою схемою. Як правило, функціональні модулі агента структуруються в кілька рівнів, однак за різними принципами. Зазвичай, рівні представляють різні функціональності: сприйняття зовнішніх подій і прості реакції на них; поведінка, керована цілями; координація поведінки з іншими агентами; оновлення внутрішнього стану агента, тобто переконань про зовнішній світ; прогнозування станів зовнішнього світу; визначення своїх дій на черговому кроці та інше. Найчастіше в архітектурі агента присутні рівні, відповідальні за: сприйняття і виконання дій, реактивну поведінку, локальне планування, кооперативну поведінку, моделювання, формування намірів, навчання агента.

У горизонтально організованій архітектурі всі рівні агента мають доступ до рівня сприйняття і дій (у загальному випадку – всі рівні можуть спілкуватися між собою в стилі «бродкастінг», а у вертикально організованій архітектурі тільки один з рівнів має доступ до рівня сприйняття і дій, а кожний з інших рівнів спілкується тільки з парою безпосередньо суміжних з ним рівнів).

Основні проблеми реалізації горизонтально організованої архітектури обумовлені складністю організації узгодженої роботи всіх рівнів. У вертикально організованій архітектурі проблема управління взаємодією рівнів не є настільки складною, оскільки вихідна інформація кожного з рівнів завжди має адресата.

Побудова агентно-базованої моделі

У роботі [4] визначено основні характерні особливості побудови агентних моделей. Як додаток до стандартних завдань побудови моделі, практичне АМ вимагає: визначити агентів і теоретичні основи поведінки агентів, визначити взаємовідносини між агентами і теоретичні основи таких відносин, знайти платформу для АМ та розробити стратегію АМ моделі, отримати необхідні дані для агентів, перевірити моделі поведінки агентів (на додаток до всієї моделі в цілому), запустити модель і проаналізувати вихідні дані з точки зору зв'язку між поведінкою агентів на мікрорівні і поведінкою всієї системи в цілому.

Визначення агентів з точним завданням їх поведінки і взаємодії з іншими агентами – це основа для розробки достовірних агентних моделей. Агенти – це ті, хто, зазвичай, приймають рішення в системі (особа, яка приймає рішення – ОПР). Вони традиційно представляють менеджерів, складні комп'ютерні системи зі складною поведінкою, групи (якщо це продиктовано цілями агентного моделювання).

Щойно агенти визначені, наступним завданням моделювання стає визначення їх поведінки. Тут рекомендовано знайти теоретичні основи поведінки агентів. Можна почати з нормативної моделі і використовувати цю модель в якості відправної точки для розробки простої і наочної евристичної моделі поведінки. Якщо є відповідна поведінкова теорія і результати її застосування виглядають адекватними, варто почати з поведінкової моделі. Наприклад, існує велика кількість теорій для моделювання поведінки покупця, заснованих на емпіричних знаннях.

Засоби розробки агентних моделей

Великомасштабне АМ розширює можливості простих агентних моделей і дозволяє більшій кількості агентів (від тисяч до мільйонів) брати участь у складних взаємодіях. Воно, зазвичай, виконується з використанням спеціальних середовищ моделювання. Традиційно ці середовища включають: планувальника, механізми комунікації, гнучкі топології взаємодій агентів, широкий вибір пристроїв для зберігання і відображення стану агентів. Найвідоміші з них це Repast [11], Swarm [12], NetLogo [13] і MASON [14].

Охарактеризуємо деякі з них.

The Recursive Porous Agent Simulation Toolkit (Repast) – це провідне відкрите і вільне джерело бібліотек для великомасштабного агентного моделювання. Repast підтримує розробку надзвичайно гнучких моделей агентів і використовується в моделюванні соціальних процесів. Користувач будує свою модель, включаючи у власні програми компоненти з бібліотеки Repast або використовуючи візуальний Repast для середовища Python Scripting.

Існує три версії Repast: Repast for Python (Repast Py), Repast for Java (Repast J) і Repast for the Microsoft.NET framework (Repast. NET).

Repast Py – це крос-платформна візуальна система для моделювання, яка дозволяє користувачам будувати моделі, застосовуючи графічний інтерфейс, і описувати поведінку агентів (скрипти Python). Всі можливості системи Repast доступні і в Repast Py, але Repast Py спроектований для швидкої розробки прототипу агентних моделей. Моделі Repast Py можуть бути автоматично експортовані у великомасштабне середовище розробки для Repast Py.

Repast J – це середовище моделювання, написане виключно на Java. Воно призначене для розробки великомасштабних агентних моделей і включає:

- паралельний дискретний планувальник за часом (календар подій);
- середовище для візуалізації моделі, засоби інтеграції з географічними інформаційними системами з метою моделювання агентів на реальних картах;
- адаптивні засоби поведінки: нейронні мережі та генетичні алгоритми.

Repast. NET – це середовище моделювання, написане виключно на C #, і переносить усі можливості Repast J на платформу Microsoft.NET. Repast. NET моделі можуть бути написані будь-якою мовою, підтримуваною платформою Microsoft.NET, як, наприклад Managed C + +, C #, Visual Basic або навіть Managed Lisp Managed чи Prolog10.1.

Repast має власний планувальник, який підтримує дискретно-подійне моделювання. Repast дозволяє використовувати великий набір комунікаційних механізмів з різноманітними топологіями взаємодії, включає повний набір утиліт для зберігання і відображення стану агентів. У систему також включено утиліти для автоматичної інтеграції як з комерційними, так і вільно доступними географічними інформаційними системами (ГІС). Інтеграція з комерційними ГІС включає автоматичне підключення до таких широковживаних географічних інформаційних систем, як ESRI та ArcGIS.

Repast є абсолютно об'єктно-орієнтованим.

Swarm уперше запущено в 1994 році. Swarm – це відкритий і безкоштовний набір бібліотек з відкритим кодом, який і наразі підтримується Swarm Development Group (SDG). Swarm прагне надати розподілену платформу для моделювання АМ і сприяє розробці широкого кола моделей. Користувач створює моделі включенням компонент з бібліотек Swarm у свої програми. Більше інформації про Swarm, так само як і завантаження, можна знайти на головній сторінці SDG [15].

Система моделювання Swarm має дві основні складові.

Компоненти ядра запускають код моделювання, написаний мовою загального призначення Objective-C, Tcl / Tk і Java. На відміну від Repast, Swarm-планувальник підтримує тільки часове просування через фіксовані проміжки. Swarm підтримує повний набір комунікаційних механізмів і може моделювати всі основні топології. Система включає хороший набір утиліт для зберігання і відображення стану агентів. Оскільки Swarm базується на комбінації Java і Objective-C, то вона – об'єктно-орієнтована. Але суміш використовуваних мов є причиною труднощів з

інтеграцією до деяких великомасштабних середовищ розробки, наприклад Eclipse. Swarm підтримує ГІС через бібліотеку Kenge.

NetLogo є програмним середовищем моделювання природних і соціальних явищ, створеним у 1999 році.

NetLogo надзвичайно добре підходить для моделювання складних систем, що розвиваються з плином часу. Дизайнер моделей може давати вказівки сотням або тисячам «агентів», що діють незалежно. Це створює можливість дослідити зв'язок між поведінкою на мікрорівні окремих індивідуальностей і макрорівні моделей, які виникають в результаті взаємодії багатьох індивідуальностей.

NetLogo має велику документацію та підручники. Існує бібліотека, яка містить велику колекцію попередньо написаних моделей, які можна використовувати і модифікувати. Ці симуляції стосуються багатьох областей у природних і соціальних науках, а також біології та медицини, фізики і хімії, математики та інформатики, економіки і соціальної психології.

NetLogo має інструмент для спільної участі в моделюванні, він називається HubNet.

NetLogo працює на віртуальній машині Java, тому працює на всіх основних платформах (Mac, Windows, Linux та інші.) Він запускається як окремий додаток або з командного рядка. Моделі і HubNet можуть бути запущені як аплети Java у веб-браузері.

Світ NetLogo складається з агентів. Агенти є істотами, які можуть виконувати інструкції. Усі існуючі агенти можуть виконувати свою власну діяльність одночасно.

У NetLogo існують чотири типи агентів:

- черепахи – turtles – рухливі агенти;
- плями або точки поверхні – patches – нерухомі точки екрана. Двовимірний світ NetLogo розділений на безліч квадратних плям. Всі ці плями створюють поверхню, якою пересуваються черепахи;
- посилання – links – з'єднують пару черепах;
- спостерігач – observer – не має певного положення у світі NetLogo, він спостерігає за плямами і черепахами. Через нього можна керувати іншими агентами.

На самому початку в світі немає черепах, але їх може створити оглядач. Черепахи, створювані оглядачем, з'являються в точці з координатами [0 0] – в самому центрі екрана. Черепахи можуть бути створені будь-якою точкою (patch) екрана. Кожна черепаха завжди має координати – xcor ycor – і ці координати не обов'язково цілі числа. Координати плям – pxcor pycor – обов'язково цілі числа.

Посилання не має координат, але завжди пов'язано з двома черепахами.

У NetLogo є команди, які вказують агентам, що робити, – переміститися, створити черепашку, встановити зв'язок і таке інше. Зазвичай, ці команди починаються з дієслова – die, jump, inspect та інші. Окрім того, є процедури, які отримують на вході дані, обробляють їх і видають результат.

Висновки

Сучасні інструменти імітаційного моделювання дозволяють ефективно застосовувати його не тільки в наукових дослідженнях, а й як засоби для побудови систем підтримки прийняття рішень у бізнесі. Для досягнення практично значущих результатів необхідно знати про особливості та обмеження кожного з існуючих підходів. Вибір тієї чи іншої парадигми повинен обумовлюватися не стільки предметною областю моделювання, а й необхідним ступенем деталізації системи та даними, що є у розпорядженні.

Більшість сучасних програмних засобів робить процес симуляції задоволенням для розробника, надаючи широкі можливості для анімації та оптимізації модельованих процесів.

Агентне моделювання дозволяє змоделювати систему максимально наближену до реальності, зробити значний крок у розумінні та управлінні сукупністю складних соціальних процесів. Агентний підхід – найкращий для моделювання соціальної поведінки: можна швидко створювати моделі з агентами, які взаємодіють як між собою, так і з навколишнім середовищем.

Існує багато засобів розробки агентних моделей. Одним із подальших досліджень може бути побудова систем АМ з прекрасними засобами візуалізації розробленої моделі.

Список літератури

1. Замятина Е. Б. Современные теории имитационного моделирования : специальный курс / Е. Б. Замятина. – Пермь : ПГУ. – 2007. – 119 с.
2. Мейтус В. Ю. Интеллектуальные компоненты в системах управления производством / В. Ю. Мейтус. – Кибернетика и систем. анализ. – 2003 – № 3. – С. 29-44.
3. Allen Newell. Human problem solving / A. Newell, H. A. Simon. – Prentice-Hall, 1972. – 920 p.
4. A. M. Uhrmacher. Simulation for Agent-Oriented Software Engineering [Електронний ресурс] / A. Uhrmacher. – Режим доступу : http://www.thesimguy.com/GC/papers/WMC02/G067_UHRMACHER.pdf – Назва з екрана.

5. Dunin-Keplicz B., Treuer J. Compositional Formal Specification of Multi-Agent System / B. Dunin-Keplicz, J. Treuer – Amsterdam, August 8-9, 1994 // Intelligent Agents. ECAI-94 Workshop on Agent Theories, Architecture and Languages / eds. M.J.Wooldridge and N.R.Jennings. – Proceedings. Springer Verlag. – P. 102–117.
6. Charles M. Macal. Tutorial on agent-based modeling and simulation part 1 [Електронний ресурс] / C. M. Macal, M. J. North // Proceedings of the 38th conference on Winter simulation. – P. 73–83. – Режим доступу: http://www.palgrave-journals.com/jos/journal/v4/n3/full/jos20103a.html. – Назва з екрана.
7. Herbert A. Simon. Models of Man: Social and Rational / H. Simon. – New York : John Wiley and Sons, Inc., 1957. – 279 p.
8. Grebel T. Agent-based modeling – A methodology for the analysis of qualitative development processes / T. Grebel, A. Pyka. – Discussion Paper Series 251 : Universitaet Augsburg, Institute for Economics, 2003.
9. Social Mechanisms and Generative Explanations: Computational Models with Double Agents / M. Macy, D. Centola, A. Flache Arnout van de Rijt, Robb Willer // Analytic Sociology and Social Mechanisms / ed. Pierre Demeulenaere. – Cambridge University, 1971. – P. 250–265.
10. Schelling Thomas C. Dynamic Models of Segregation / T.Schelling // Journal of Mathematical Sociology. Vol. 1. – 1971. – P. 143–186.
11. The Repast Suite [Електронний ресурс]. – Режим доступу: http://repast.sourceforge.net – Назва з екрана.
12. Swarm – Summary [Електронний ресурс]. – Режим доступу: http://www.swarm.org/. – Назва з екрана.
13. NetLogo [Електронний ресурс]. – Режим доступу: http://ccl.northwestern.edu/netlogo/. – Назва з екрана.
14. Mason-2.20 [Електронний ресурс]. – Режим доступу: http://search.cpan.org/~jswartz/Mason-2.20/. – Назва з екрана.
15. LinkedIn [Електронний ресурс]. – Режим доступу: http://www.linkedin.com/company/swarm-developme. – Назва з екрана.

A. Glybovets

AGENT-BASED MODELING

Article is about key methodology in agent-based modeling and main agent-based products.

Keywords: agents modeling, agents, agent-based model, environment for development of agents models.

Матеріал надійшов 01.09.2013

УДК 519.8

R. Trygub, O. Trygub, V. Gorborukov

RESEARCHING SEMISTRUCTURED PROBLEMS OF MULTICRITERIA OPTIMIZATION USING THE SOFTWARE SYSTEM

Develop the optimal decision support system for solving semistructured problems of multicriteria optimization, which can be used by individual or collegial body who take responsible decisions. Currently existing software tools, which solve this class problems are limited only by finding the best alternative, whereas the proposed system also (in addition to solving this problem) allows to develop instructions ("guidelines for actions") for any of losing alternatives so that the observance of them will guarantee the winning for this alternative.

Keywords: semistructured problems, multicriteria optimization, analytic hierarchy process, criteria, alternatives.

Among multicriteria problems connected with decision support [1–4], which very often occur in practice, the problems of alternatives choosing stay actual [5–7]. Mathematically, such problems are described by a set of alternatives and all of them are given the values of certain parameters (criteria).