Отримано: 19.03.2015 р.

Прорецензовано: 25.03.2015 р.

Прийнято до друку: 30.04.2015 р.

Примостка А. О. Особливості розробки та проектування мультиагентних систем / А. О. Примостка // Наукові записки Національного університету «Острозька академія». Серія «Економіка» : збірник наукових праць / ред. кол. : І. Д. Пасічник, О. І. Дем'янчук. – Острог : Видавництво Національного університету «Острозька академія», 2015. – Випуск 28. – С. 159–163.

УДК: 004.415.2:004.81

JEL-класифікація: C69

Примостка Андрій Олександрович,
аспірант Державного вищого навчального закладу
«Київський національний економічний університет імені Вадима Гетьмана»

ОСОБЛИВОСТІ РОЗРОБКИ ТА ПРОЕКТУВАННЯ МУЛЬТИАГЕНТНИХ СИСТЕМ

Проаналізовано трактування поняття «агент» у різних методологіях розробки мультиагентних систем. Розглянуто принципи проектування мультиагентних систем на базі вже розробленого програмного забезпечення. Запропоновано спосіб представлення агента не тільки у вигляді неподільної одиниці, але й як системи, яка сама може бути мультиагентною. Також запропоновано всі сутності, які безпосередньо не включені до агента, розглядати як зовнішнє середовище агента. Застосування такого підходу дозволяє уніфікувати процедуру розробки мультиагентних систем для різних рівнів деталізації цільової системи з використанням рекурсивної деталізації її проектування.

Ключові слова: *агент, властивості агентів, мультиагентна система, уніфікація процесу проектування мультиагентних систем.*

Примостка Андрей Александрович,
аспирант Государственного высшего учебного заведения
«Киевский национальный экономический университет имени Вадима Гетьмана»

ОСОБЕННОСТИ РАЗРАБОТКИ И ПРОЕКТИРОВАНИЯ МУЛЬТИАГЕНТНЫХ СИСТЕМ

Проанализирована трактовка понятия «агент» в разных методологиях разработки мультиагентных систем. Рассмотрены принципы проектирования мультиагентных систем на базе уже разработанного программного обеспечения. Предложен способ представления агента не только в виде неделимой единицы, но и как системы, которая сама может быть мультиагентной. Также предложено все сущности, которые непосредственно не включаются в агента рассматривать как внешнюю среду агента. Применение такого подхода позволяет унифицировать процедуру разработки мультиагентных систем для разных уровней детализации целевой системы с использованием рекурсивной детализации ее проектирования.

Ключевые слова: *агент, свойства агентов, мультиагентная система, унификация процесса проектирования мультиагентных систем.*

Andrii Prymostka,
Graduate student State higher education establishment «Vadym Hetman National Economic University of Kyiv»

PECULIARITIES OF THE DEVELOPMENT AND DESIGN OF MULTI-AGENT SYSTEMS

The analysis of the notion of «agent» in different multi-agent systems development methodologies has been made. The common for different methods principles of development of multi-agent systems based at existing developed methods of projecting of software support have been overviewed. The way of introducing the agent not only as an indivisible entity, but also as a system that could be multi-agent, have been offered. It`s also offered to overview as an external environment of the agent all entities that are directly included into the agent. The method of unification of development of multi-agent systems for different levels of detailing of target system due to unification of the notion «agent» with using of recursive detailing of system projecting has been offered with using of this view.

Keywords: *agent, multi-agent system, unification of development of multi-agent systems.*

Постановка проблеми. З розвитком та ускладненням програмних комплексів виникла необхідність спрощення представлення складних систем та урахування в них можливих дій користувача. Одним із ефективних способів вирішення цієї проблеми став мультиагентний підхід до моделювання складних систем. Але на цей час розроблені процедури проектування складних мультиагентних систем додатково ускладнюють власне сам процес проектування, пропонуючи використовувати різнорідні об'єкти, окрім самих агентів. Вивчення наукових праць показало, що вчені виокремлюють різні типи об'єктів у процесі симуляції мультиагентних систем [1-4]. Для кожного типу об'єктів зазвичай наводиться окрема методика проектування, власна онтологія, свій набір рекомендацій щодо реалізації та алгоритмів, що використовуються для симуляції. Така неузгодженість ускладнює процес проектування та взаєморозу-

міння розробників мультиагентних систем, а отже, вимагає уніфікації як термінологічного апарату, так й етапів процесу проектування та реалізації таких програмних продуктів.

Аналіз останніх досліджень та публікацій. Дослідженню мультиагентних систем присвячено багато наукових праць [1-3]. Із розвитком систем, що використовують мультиагентний підхід, було розроблено різні методики проектування таких комплексів, яким присвячено праці зарубіжних учених [4-8]. Мультиагентний підхід до моделювання складних систем є відносно новим, тому більшість понять і термінів не мають однозначного трактування. Мультиагентний тип систем розвивався як частина розподілених систем штучного інтелекту, а головною їх особливістю є здатність до виконання значного обсягу робіт високої складності, які жоден агент автономно зробити не спроможний. Зараз мультиагентні системи застосовуються у багатьох програмних продуктах різного цільового призначення, зокрема в управлінні підприємством, системах оборони, транспорті, логістиці, графіці, фільмовиробництві, комп'ютерних іграх та ін. На цей час сформувалися різні концепції та підходи до побудови мультиагентних систем, які відрізняються за логікою представлення, рівнем автоматизації, кількістю користувачів, можливостями доступу до сховищ і баз даних. Перспективним напрямом є наукові дослідження кооперативних мультиагентних систем, активізація яких була зумовлена зростанням потужності та розповсюдженості комп'ютерної техніки.

Мета і завдання дослідження. Метою статті є висвітлення сутності, особливостей і можливості застосування мультиагентного підходу до побудови інформаційно-аналітичних систем у банківській діяльності, що дозволить суттєво підвищити ефективність управлінського процесу.

Виклад основного матеріалу. Мультиагентні системи, або, як ще їх називають, агентно-орієнтовані системи, – це сукупність агентів (автономних модулів), здатних до взаємодії. Хоча єдиної думки щодо розуміння поняття «агент» на цей час не склалося, але у широкому сенсі під агентом розуміють апаратну або програмну сутність (модуль), здатну діяти в інтересах досягнення цілей, поставлених перед нею іншим учасником системи: як користувачем, так і розробником, а також іншим агентом. Уживаючи термін «агент», дослідники трактують його на власний розсуд і по-різному описують його особливості. Здебільшого, характеризуючи програмних агентів, учені акцентують увагу на різних, притаманних їм, властивостях [2], [3], [4], [5]. Найчастіше виокремлюють такі властивості агентів:

– можливість програмної реалізації агентів;

– автономність – можливості агента діяти самостійно, на основі своїх спостережень зовнішнього, відносно агента, середовища;

– раціональність – максимізація міри якості, що визначена на множині можливих результатів роботи агента;

– комунікативність – ступінь і спосіб взаємодії агентів у межах системи;

– адаптивність (сприйнятливість) – можливості агентів сприймати своє оточення та реагувати на нього;

– про-активність – виявлення агентами спрямованості на досягнення цілі.

Також у багатьох роботах згадується така характеристика, як тривалість періоду функціонування.

Агентів також називають розумними агентами (*Intelligent agent*), тому що саме ця характеристика є ключовою при їхньому створенні. Агент може одержати завдання, яке необхідно виконати, а може мати мету, якої треба досягти, й у відповідності з цим приймати потрібні рішення в процесі комунікації з іншими агентами. Загалом, вирішення певного завдання або досягнення мети можна розглядати як рівнозначні сутності, адже у кожен конкретний момент часу результатом досягнення поточної цілі може стати як вирішення отриманого завдання, так і досягнення глобальної мети функціонування агента. Це означає, що в кожен конкретний момент часу агент спрямований на отримання одного задовільного результату роботи. Під результатом мається на увазі сукупність усіх задовільних кінцевих станів роботи агента, що дозволяють досягти поточної мети.

Для складних мультиагентних систем розробляються спеціальні методики проектування та реалізації на базі вже розроблених методик або поєднання кількох підходів в одній методиці. На цей час запропоновано низку методик проектування МАС, таких як Tropos [4], KADS [6], Prometheus [7], ADELFE [8] та інші. Ці підходи базуються на класичних підходах до розробки програмного забезпечення. В класичних алгоритмах розробки програмного забезпечення (зокрема, у процесі безпосередньої розробки, без врахування організаційних етапів, таких, наприклад, як підтримка чи тестування програмних модулів) виділяють три основні етапи розробки нових програмних комплексів:

– визначення вимог до системи;

– проектування, або розробка архітектури;

– реалізація.

З урахуванням специфіки мультиагентних систем у методиках проектування передбачено деталізоване та специфічне розбиття класичних етапів на окремі фази, у кожній із яких увага акцентується на конкретних цілях і завданнях, які вважаються найбільш важливими на поточний момент процесу розробки. Наприклад, у методиці Tropos етап проектування розбивається на фази архітектурного та деталізованого

дизайну, що фокусуються на специфікації системи відповідно до вимог, виділених на попередніх фазах. Архітектурний дизайн визначає глобальну архітектуру системи в термінах підсистем, пов'язаних через потоки даних та управління. На етапі детального дизайну метою є визначення можливостей агентів та їх взаємодії [4].

Отже, у процесі проектування агентів вимоги до функціонування кожного агента визначаються в межах вимог конкретної мультиагентної системи. Це означає, що за умови зміни вимог до МАС необхідно змінити й агентів, тобто в загальному випадку агенти не можуть бути вилучені із системи для функціонування в межах іншої системи чи в автономному режимі. Таким чином, агенти не можуть мати ціль, корисну власне тільки самому агенту, або одну ціль для кількох МАС одночасно.

Етап визначення вимог може у неявному вигляді «розтягнутися» і на етап проектування, оскільки саме тут визначаються вимоги до кожного конкретного агента. Тобто формально етап проектування не передбачає визначення вимог до системи загалом, але саме на цьому етапі відбувається визначення вимог до агентів, які розглядаються як програмний продукт, що може діяти самостійно на основі своїх спостережень зовнішнього середовища. Такий підхід використовується, наприклад, у методиці Tropos, де на етапі визначення, формулюють вимоги до системи, на етапі архітектурного дизайну визначають підсистеми, що розглядаються як самостійні модулі (сутності) цієї методики – актори, яких фактично можна трактувати як агентів. Далі переходять до другої фази – поділу підсистем на агентів, для яких реалізуються етапи визначення вимог, проектування та реалізації у межах прийнятої методики, тобто відбувається повторення тих же самих етапів проектування, що і для системи в цілому. Отже, в процесі проектування МАС етапи розробки власне системи та розробки окремих агентів подібні.

Це дає підстави розглядати програмного агента як окрему систему зі специфічними вимогами до взаємодії, що накладаються на загальносистемному рівні, а отже, застосовувати ті ж техніки та прийоми проектування, що й для системи в цілому. Тобто, якщо на етапі проектування виокремлювати не кінцеві (або «атомарні») сутності агентів, а розглядати поділ на підсистеми через призму мультиагентних систем, то після виділення підсистем до них можна застосувати рекурсивно той же самий процес проектування, що і для системи загалом. Точкою зупинки в такому поділі є критерій неможливості подальшого поділу цілей агента без втрати змістовного навантаження в процесі формулювання цілей наступного рівня поділу.

В описаній уніфікації процесу проектування мультиагентної системи постає питання комунікації агентів чи підсистем між собою. Суть проблеми полягає в тому, що на різних рівнях поділу системи на складові ефективними можуть бути різні протоколи зв'язку, які відповідають рівням абстракції предметної області для такого розбиття. Тобто, на верхньому рівні ефективним протоколом може бути високорівнева мова запитів, наприклад KQML, а в процесі подальшої деталізації для комунікації в межах однорівневої підсистеми ефективною може виявитись інша мова, така, наприклад, як FIPA-ACL або, в межах підсистеми, SQL.

Таким чином, робимо висновок, що на кожному рівні деталізації структурні елементи конкретного рівня ефективно взаємодіють із суміжними елементами за допомогою мови, яка дає можливість ефективно виразити мету та цілі, притаманні агентам конкретного рівня деталізації. Також, очевидно, що за наявності можливості комунікації за допомогою різних протоколів зв'язку обирати протокол з усіх розроблених буде агент-ініціатор, а всі інші агенти, котрі беруть участь у комунікації, повинні самі трансформувати обраний протокол до зручного для них вигляду.

Оскільки практично кожен агент є не лише ініціатором, тому має модуль, який конвертує запити на обраній ініціатором мові для представлення, зручного для агента. Тобто, під час розробки онтології, яка задовольнить потреби різнорівневих агентів, кожен агент має можливість сформулювати запит у термінах онтології. Це означає, що конвертацію запиту з конкретної мови та протоколу до процедур в рамках і визначеннях онтології мультиагентної системи може здійснити сам агент-ініціатор. Сказане дозволяє зробити висновок, що реалізація багатьох різних протоколів взаємодії агентів у межах однієї системи є недоцільною. Якщо розглядати взаємодію агентів усередині системи з єдиною онтологією та можливості формулювання запитів у термінах обраної онтології, то є сенс реалізації протоколу взаємодії в термінах єдиної онтології системи.

У процесі розробки мультиагентних систем автори здебільшого виокремлюють різні типи кінцевих об'єктів, що використовуються в процесі симуляції. При чому, у різних підходах цей поділ здійснюється залежно від різних властивостей об'єктів. Наприклад, у одній роботі розмежовують агентів та об'єкти, у другій – агентів та артефакти, а в інших здійснюється групування агентів за деякою спільною ознакою у єдиний агент зі своїми внутрішніми онтологіями та протоколами зв'язку. Для кожного типу об'єктів зазвичай наводиться окрема методика проектування, власна онтологія, свій набір рекомендацій щодо реалізації та алгоритмів, що використовуються у процесі симуляції. Така неузгодженість ускладнює загальне проектування та розуміння МАС, збільшує обсяг інформації, необхідної для її сприйняття, призводить до неможливості уніфікувати як методичні підходи до розробки, так й алгоритми реалізації

системи та протоколи комунікації. Також за такого підходу значно ускладнюється можливість симуляції частини системи, оскільки різні типи об'єктів системи можуть мати різну поведінку.

У роботі [9] запропоновано підхід, відповідно до якого в процесі симуляції мультиагентних систем усі неподільні сутності розглядаються як агенти. Це дозволяє провести уніфікацію різних типів об'єктів і звести їх до єдиного розуміння поняття «агент». Оскільки агенти неоднорідні, то за такого трактування залишається поділ агентів на активних, пасивних і нестійких. Активні агенти відрізняються від пасивних тим, що самі можуть ініціювати дії.

Вище було показано, що поняття «агент» доцільно розширити також і на інші артефакти проектування системи в цілому, зокрема такі, наприклад, як підсистеми. Тобто, під час проектування системи кожен цикл формулювання вимог і проектування поточного рівня деталізації слід розглядати як такий, що не відрізняється від циклів інших рівнів за способами та методами, що використовуються, а різниця полягає лише в об'єктах, до яких застосовуються обрані методи проектування.

Цілком очевидно, що результатом роботи агента мають бути кількісні чи якісні зміни зовнішнього, щодо самого агента, середовища. Тобто кожен агент у результаті своєї роботи впливає на зовнішнє середовище чи саму систему. З погляду характеристики раціональності результат роботи агента повинен мати критерій оцінювання для того, щоб агент міг самостійно оцінити результати діяльності та, за можливості, покращити їх чи змінити свою поведінку. Отже, кожен агент повинен мати результатом своєї роботи такі прояви, що можуть бути оцінені, інакше будь-які дії, що не мають зовнішнього прояву можуть бути оцінені агентом, як раціональні, у тому числі і не виконання жодних дій. При чому не виконання жодних дій у разі відсутності результату буде сприйматись як оптимальна стратегія, оскільки за відсутності результату не затрачаються жодні зусилля.

З позицій автономності будь-який агент повинен реагувати на зміну зовнішнього середовища у межах своїх можливостей оцінки такої зміни. У загальному розумінні зовнішнім є середовище, в якому працює система в цілому. Але з позиції кожного окремого агента, зовнішнім середовищем буде також і сама система, адже агент повинен реагувати як на зміни ззовні системи, так і на вагомі для нього зміни всередині системи. Також агент повинен узаємодіяти з іншими агентами системи. Отже, з погляду проектування агента немає сенсу в диференціації середовища на зовнішнє відносно системи та внутрішнє, якщо існує можливість впливати на систему за допомогою комунікаційних каналів, оскільки для підтримки своєї автономності агент повинен однаково адекватно реагувати на будь-які зміни, значущі для його роботи.

Таким чином, під зовнішнім середовищем агента пропонуємо розуміти все, що не включено до самого агента, а саме як мультиагентну систему (за винятком самого агента), так і середовище, у якому ця система функціонує. За такого підходу агенти безпосередньо впливають на зовнішнє середовище, або безпосередньо змінюючи середовище, у якому функціонує вся система, або впливаючи на систему за допомогою комунікативних засобів самої системи. Як наслідок, виникає питання диференціації між безпосередньою зміною середовища та повідомленням усередині системи, але, під час уведення в систему модулів, що відповідають за трансляцію запитів у термінах онтології системи у безпосередній вплив на середовище різниця між зміною середовища та повідомленнями зникає. Такі модулі можуть існувати як окремо, так і у формі складових частин агента. Таким чином, кожен агент може впливати на систему через відомий усім агентам комунікативний канал системи у рамках фізично відокремленої групи агентів.

Висновки. Запропонований підхід дозволяє розглядати агента як повністю незалежну та автономну одиницю мультиагентної системи, що сама може складатися з окремих агентів. Це значно спрощує проектування та розуміння мультиагентної системи в цілому, оскільки на кожному рівні системи потрібно визначити функціональність кожного агента такого рівня окремо та визначити онтологію, що буде використовуватись усередині системи. А на рівні агентів проектування можна проводити окремо для кожного агента, базуючись на необхідній функціональності агента, цілях, яких необхідно досягти, та переліку важливих факторів зовнішнього середовища, що мають бути враховані у процесі функціонування.

Також перевагою є те, що проектування та реалізацію програмних агентів можна проводити окремо, за винятком спільної онтології, що забезпечується відповідною комунікацією. Також за цього підходу для комунікації агентів усередині системи можна використовувати існуючі засоби, що забезпечують прийнятні параметри зв'язку, такі як захищеність, швидкість, розподіленість, можливість використання в гетерогенних середовищах та інші.

Запропонований підхід до розробки мультиагентних систем дозволяє:

– уніфікувати проектування технічних аспектів мультиагентної системи;

– спростити проектування як системи в цілому, так і проектування кожного агента окремо;

– спростити розуміння процесу функціонування системи, розбивши її на повністю незалежні модулі-агенти;

– удосконалити тестування як частин системи, так і системи в цілому, за рахунок того, що агенти за такого підходу

– не впливають один на одного;

– уніфікувати та використати існуючі програмні засоби для комунікації агентів;

– паралельно розробляти різних агентів незалежно від їх функціональності.

Перевагами цього підходу є те, що мультиагентна система в автономному режимі може адаптуватись до змін зовнішнього середовища, залежно від налагоджень системи аналізувати навантаження та працювати у моменти його спадів, легко може бути розподілена для роботи у гетерогенному середовищі та добре масштабується. Оскільки кожен агент відповідає за дані лише у момент роботи, МАС може без втручання коригувати навантаження на техніку і в той же час забезпечувати високу швидкість відповіді та можливості збереження змінних за структурою даних.

Література:

1. Stone P. and Veloso M. «Multiagent Systems: A Survey from Machine Learning Perspective» / Stone P. and Veloso M. // School of Computer Science Carnegie Mellon University – Pittsburg: December 1997.

2. Городецкий В. И. и др., «MAS DK: инструментарий для разработки многоагентных систем и примеры приложений» / Городецкий В. И. и др. // Труды Междунар. Конгресса «Искусственный интеллект в XXI веке» (ICAI 2001) – 2001 – С. 249–262.

3. Швецов А. Н. и др., «Инструментальный программный комплекс для проектирования мультиагентных систем» / Швецов А. Н. // Материалы IX Междунар. Конференции. «Интеллектуальные системы и компьютерные науки» – М.: 2006 – Т. 2, Ч. 2. – С. 309–312.

4. «Tropos: An Agent-Oriented Software Development Methodology» / BRESCIANI PAOLO, PERINI ANNA, GIORGINI PAOLO, FAUSTO GIUNCHIGLIA и JOHN MYLOPOULOS // Autonomous Agents and Multi-Agent Sytems – 2004 – 8 – pp. 203-206.

5. «Software agents: An overview» / Nwana H. // The Knowledge Engineering Review Journal – 1996 – vol. 11, № 3 – pp.1-40.

6. Schreiber, G., Weilinga, B. и Breuker, J., «KADS: A Principled Approach to Knowledge-based Systems» / Schreiber, G., Weilinga, B. и Breuker, J. // Academic Press – London: 1993.

7. «The Prometheus Methodology» / Michael Winikoff, Lin Padgham // Multiagent Systems, Artificial Societies, and Simulated Organizations – 2004 – Volume 11 – pp. 217-234.

8. «ADELFE: A Methodology for Adaptive Multi-agent Systems Engineering» / Carole Bernon, Marie-Pierre Gleizes, Sylvain Peyruqueou, Gauthier Picard // Engineering Societies in the Agents World III, Lecture Notes in Computer Science – 2003 – Volume 2577 – pp. 156-169.

9. «Everything can be Agent! (Extended Abstract)» / Kubera Y. et al. // Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010) – Toronto: 2010 – Volume 1 – pp. 1547–1548.