

УДК 681.3, 004.85, 004.89



К.А. Ручкін

ДонНТУ, м. Донецьк, Україна, c_ruchkin@mail.ru

АНАЛІЗ ТА ПРОЕКТУВАННЯ БАГАТОАГЕНТНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ПОВЕДІНКИ НЕЛІНІЙНОЇ ДИНАМІЧНОЇ СИСТЕМИ У РЕАЛЬНОМУ ЧАСІ

У роботі продовжені дослідження [1], які пов'язані з розробкою багатоагентної системи прогнозування поведінки нелінійної динамічної системи у реальному часі. Складена загальна схема вирішення задачі, описана загальна архітектура багатоагентної системи, розроблена структура бази даних, спроектовані агенти багатоагентної системи. Запропонована багатоагентна система спроектована на основі методу розподіленого штучного інтелекту з використанням даних, одержаних з середовища моделювання Modeler [2] в режимі реального часу. Система складається з сімнадцяти агентів: інтерфейсного агента, інформаційного агента, агента-планувальника, керуючого агента, тактового агента і дванадцяти спеціалізованих агентів-розпізнавачів. Наведено специфікацію багатоагентної системи, її агентів та бази даних у вигляді діаграм UML, а також опис структури, функцій і порядку роботи системи та примітки щодо її програмної реалізації в майбутньому.

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ, БАГАТОАГЕНТНІ СИСТЕМИ, ЗАДАЧА ПРОГНОЗУВАННЯ,
НЕЛІНІЙНІ ДИНАМІЧНІ СИСТЕМИ

Вступ

Задача побудови багатоагентної системи (БАС) дослідження та прогнозування поведінки динамічних систем є досить новою та складною задачею. Її суть полягає у наступному. Необхідно дослідити та спрогнозувати поведінку нелінійної автономної багатопараметричної динамічної системи на основі графічної інформації (двовимірних та тривимірних перетинів Пуанкаре), яка отримується у реальному часі. Складність розв'язування цієї задачі полягає в тому що, вона погано формалізується та може бути віднесена до задач штучного інтелекту. Так, наприклад, проведення окремих досліджень двовимірних або тривимірних перетинів Пуанкаре недостатньо для вирішення початкової проблеми. Також для побудови перетинів Пуанкаре нефіксований час, який необхідно для розрахунку перетинів. Малий час – не дає змоги побудувати адекватні, великий час – дозволяє накопичувати помилки. Задача ускладнюється і тим, що динамічна система є багатопараметричною. У залежності від параметрів вона може мати протилежні властивості: бути добре прогнозованою або ні. Для вирішення цих проблем пропонується застосувати багатоагентно-орієнтовану технологію та розробити відповідну багатоагентну систему прогнозування поведінки нелінійної динамічної системи у реальному часі [1].

Як відомо, у сучасних дослідженнях систем штучного інтелекту агентно-орієнтовані технології знаходять все більш широке застосування. Ці технології можуть бути розглянуті як нова парадигма задля аналізу, проектування, розробки та реалізації складних програмних систем інтелектуального напрямку. В агентно-орієнтованих технологіях агенти – це складні комп'ютерні програми (або підпрограми), які можуть діяти автономно (від імені користувача) для вирішення поставлених задач

згідно з обраною стратегією поведінки. Проте все більше задач потребує застосування декількох різноманітних стратегій, також як і декількох агентів. В мультиагентних системах агенти повинні взаємодіяти кооперативно між собою та виробляти більш адекватні кооперативні рішення. Багато-агентні інтелектуальні системи мають більш вагомі переваги перед класичними системами штучного інтелекту та здатні вирішувати більш складні задачі, наприклад, задачі пошуку, розпізнавання та прогнозування. Але їх проектування та моделювання є досить складною задачею та залежить від вибраної методології проектування БАС. Розглянемо сучасний стан напряму багатоагентно-орієнтованого аналізу та проектування і його методи.

1. Сучасний стан напряму багатоагентно-орієнтованого аналізу та проектування

Серед останніх досліджень з використанням багатоагентних систем для вирішення різноманітних завдань, пов'язаних з комплексними і розподіленими інтелектуальними обчисленнями, відзначимо біоінформаційний аналіз [3,4], пошук та вилучення інформації [5-9], електронну комерцію та управління [11-13], логістику та імітаційне моделювання динамічних систем та організацій [10, 14], розпізнавання образів [15-19], експертні системи та системи підтримки прийняття рішень [21,22].

Особливий інтерес для цього дослідження представляє собою багатоагентні системи розпізнавання графічних образів та багатоагентні системи реального часу. Розглянемо деякі з них детальніше.

Так А. Аддіс, Г. Арман і Е. Варгю розробили загальну архітектуру багатоагентної системи для пошуку та обробки інформації [5]. Дана архітектура передбачає наявність у системи таких можливостей, як пошук інформації в безлічі різнорідних

джерел (зокрема в мережі Інтернет), динамічна інтеграція нових джерел в процесі пошуку, контроль релевантності результатів, врахування відгуків та уподобань користувача. Отримана архітектура лягла в основу кількох багатоагентних систем, а саме системи керування мультимедіа-ресурсами, системи моніторингу човнів і системи класифікації вмісту Вікіпедії.

Г. Пур провела дослідження зі створення гнучкої, адаптивної, розширюваної і робастної системи для контролю якості в режимі реального часу з метою застосування в сфері електронної комерції [13]. Результатом дослідження стала архітектура і прототип багатоагентної системи, заснованої на принципах автономних комп'ютерних систем (autonomic computing). Дана архітектура передбачає наявність у системи таких можливостей, як моніторинг параметрів якості, оцінка рівня якості відповідно до заданих стандартами, вироблення і виконання планів для регулювання рівня якості.

Д. Рахалем, Ф. Рахалем і М. Чекруном була розроблена SMAST – багатоагентна система для моделювання транспортних систем [14]. Дана БАС призначена для моніторингу, діагностики та підтримки управління трафіком в режимі реального часу. У SMAST кожен суб'єкт дорожнього руху представляється у вигляді незалежного діючого агента з певними задачами і переконаннями. За допомогою додаткових агентів для моніторингу та аналізу переміщення транспорту SMAST дозволяє передбачати дорожні інциденти, моделювати їх протікання і пропонувати можливі шляхи їх вирішення.

Дж. Каселлою, В. Деуфеміей і В. Маскарді була розроблена AgentSketch – багатоагентна онлайн-система для розпізнавання намальованих від руки діаграм в режимі реального часу [15]. Дана БАС призначена для використання в сукупності з планшетними і пір'яними (pen-based) інтерфейсами. AgentSketch використовує багатоагентний підхід для управління багатьма розпізнавачами символів і підвищення ефективності розпізнавання шляхом аналізу контекстної інформації. Передбачається, що отриману систему можна буде застосовувати в різних предметних областях шляхом створення розпізнавачів символів для цих областей. Розроблена БАС була успішно протестована в задачі розпізнавання діаграм UML.

О. Гемпелем, У. Бекером і Дж. Хартманном була розроблена Visent – загальна архітектура багатоагентної системи для паралельного розпізнавання об'єктів [16]. Дана архітектура призначена для створення систем візуального розпізнавання, що діють на основі бази знань. Процес розпізнавання розподіляється між багатьма агентами, кожен з яких намагається розпізнати задану частину потрібованого об'єкта. Тестування отриманого

прототипу багатоагентної системи продемонструвало збільшення швидкості розпізнавання в порівнянні з аналогічним багатопотоковим рішенням за рахунок обміну інформацією між агентами.

Р. Джума, П. Леклерк і Ш. Азар розробили EsqQUIsE – багатоагентну онлайн-систему для захвату та інтерпретації архітектурних креслень, намальованих від руки [17]. Дана система призначена для підтримки ранніх етапів проектування, коли архітектурний проект являє собою намальований від руки ескіз. БАС побудована за багаторівневою схемою з розбивкою комплексної задачі розпізнавання на простіші підзадачі, аж до інтерпретації найпростіших графічних примітивів. Отриманий прототип дозволяє проводити розпізнавання креслень в режимі реального часу, тобто по мірі їх малювання формувати семантичну і тривимірну моделі будівлі, що проектується.

Дж. Ржевським, П. Скобелевим, І. Мінаковим, С. Вольманом був розроблений багатоагентний метод динамічної кластеризації [18]. У даній БАС кожен окремих підлягаючий класифікації об'єкт асоціюється з агентом, а кластери формуються в процесі переговорів і об'єднання цих агентів з метою збільшення оцінки системи (загального показника кластеризації). Головними особливостями розробленої системи є здатність працювати з динамічними даними (нові об'єкти можуть у будь-який момент часу включатися в процес кластеризації) і здатність до еволюції (кластери можуть виникати, зникати і міняти свій склад у процесі досягнення максимальної оцінки системи).

Г. Саламою, А. Фахмі і М. Ельбаром була розроблена багатоагентна модель і прототип системи для розпізнавання осіб [19]. Цю БАС передбачається використовувати для біометричної ідентифікації особистості по мережі. Дана система комбінує кілька методів розпізнавання, розподіляючи завдання ідентифікації між кількома незалежними агентами і виводячи підсумковий результат розпізнавання на основі колективного результату цих агентів. Тестування системи показало як збільшення ефективності, так і зменшення витрат часу на процес розпізнавання.

К. Сас, Г. О'Харе і Р. Рейллі провели дослідження перспективи створення адаптивних віртуальних середовищ (virtual environment) [21]. У процесі дослідження була проаналізована можливість створення інтелектуальної системи підтримки віртуальної навігації та динамічної адаптації віртуального середовища на основі багатоагентної підходу. Був розроблений ранній прототип такої системи, здатної аналізувати і класифікувати траєкторії рухів користувачів у віртуальному просторі.

В. Розколом, А. Левассьєром, О. Чабролом, С. Грузе, Є. Данчіном і П. Понтаротті була розроблена CASSIOPE (Clever Agent System for Synteny

Inheritance and Other Phenomena in Evolution) – багатоагентна біоінформаційна система для пошуку консервативних ланцюжків в геномах в режимі реального часу [22]. Розроблена система дозволяє автоматично ідентифікувати потрібні ланцюжки у великому обсязі динамічних вхідних даних, проводити оцінку значущості отриманих результатів, а також виконувати реверсивний пошук.

Таким чином проведений аналіз показує ефективність багатоагентної технології задля вирішення різноманітних науково-технічних задач. Розглянемо методи побудови багатоагентних систем.

2. Дослідження методів агентно-орієнтованого проектування і аналізу

Поряд з дослідженнями, пов'язаними із застосуванням багатоагентних систем та розробкою нових багатоагентних архітектур, варто відзначити роботи по створенню нових методологій [23, 24, 25] і фреймворків [17, 22, 26] для багатоагентного проектування та аналізу.

М. Вулдріджем, Н. Дженнінгсоном і Д. Кінні була розроблена GAIA – універсальна методологія для агентно-орієнтованого аналізу і проектування [25]. Дана методологія базується на уявленні багатоагентної системи у вигляді обчислювальної організації, що складається з різних взаємодіючих ролей. В розробленій системі кожен агент виконує одну або кілька таких ролей, реалізуючи пов'язані з ними функції та властивості. GAIA не враховує предметну область і архітектуру окремого агента, тому вона може бути застосована для розробки широкого кола багатоагентних систем.

К. Берноном, М. Глейзе, С. Пейрукву і Г. Пікардом була розроблена ADELFE – методологія для розробки адаптивних багатоагентних систем [27]. Ця методологія є доповненням до більш загальної методології розробки програмного забезпечення Rational Unified Process. ADELFE призначена для створення систем з високим рівнем стійкості до внутрішніх і зовнішніх змін. Дана методологія приділяє основну увагу середовищу функціонування системи, робить акцент на забезпеченні можливості самоорганізації агента для адаптації до змін у середовищі, а також на розпізнаванні та вирішенні «некооперативних» (виняткових) ситуацій при взаємодії агентів один з одним.

А. Омідіні була розроблена SODA – агентно-орієнтована методологія для розробки та аналізу Інтернет-систем [28]. Дана методологія робить акцент на проектуванні спільнот агентів і розглядає індивідуального агента перш за все як частину такої спільноти, яка має певний набір соціальних функцій і якостей. Використовуючи такі поняття, як завдання, роль, ресурс, сервіс, права і протокол взаємодії SODA дозволяє відобразити соціальну

модель агента на його інфраструктуру. Таким чином, головна риса SODA полягає в тому, що внутрішнє влаштування індивідуального агента виникає з його соціальної ролі і зовнішніх якостей.

На основі зробленого огляду типів агентів, деяких агентних архітектур та моделей поведінки, можна зробити висновок, що найбільш підходящим для розробляємої системи варіантом побудови індивідуального агента є реактивна архітектура, що визначає діяльність агента у вигляді реакцій на зміни у зовнішньому середовищі. Використання цієї архітектури не дозволяє домогтися високого рівня самоврядування, але значно спрощує процес побудови агента та забезпечує надійність і передбачуваність його поведінки.

3. Постановка завдання

Метою даної роботи є розробка багатоагентної системи, що заснована на використанні методу розподіленого штучного інтелекту і призначена для прогнозування поведінки деякої нелінійної динамічної системи у реальному часі [1,2].

Задля цього необхідно виконати наступне:

- описати загальну схему вирішення поставленої задачі;
- спроектувати агентів системи, описати їх ролі та функції;
- розробити структуру бази даних системи;
- описати загальну архітектуру та технічні аспекти реалізації розробленої багатоагентної системи;

До багатоагентної системи пред'являються наступні вимоги:

- 1) автономність (робота без втручання користувача);
- 2) функціонування в режимі реального часу (паралельно з роботою Modeler);
- 3) розпізнавання різних типів траєкторій на графічних уявленнях моделі, яка досліджується;
- 4) регулювання часу прорахунку кожної окремої траєкторії;
- 5) визначення початкових точок для розрахунку траєкторій;
- 6) оптимізація процесу розпізнавання шляхом регулювання числа задіяних в ньому компонентів системи;
- 7) мінімізація помилок розпізнавання шляхом використання багатьох автономних розпізнавачів;
- 8) ведення бази даних, яка містить опис виявлених траєкторій динамічної системи.

4. Аналіз та проектування загальної схеми БАС

Багатоагентна система (БАС) – це система, що складається з багатьох інтелектуальних агентів, сполучених один з одним (рис. 1). Метою використання багатоагентної системи є вирішення задачі, яку надто складно або неможливо вирішити за

допомогою індивідуального агента або монолітної системи.

Архітектуру багатоагентної системи має сенс розглядати в двох площинах: архітектура складаючих її агентів і організація взаємодії цих агентів між собою.

Агент – це система, що знаходиться у деякому середовищі і здатна виконувати деякі автономні дії для виконання закладених у неї цілей. Агент може виконувати певний набір дій, за допомогою чого він впливає на стан середовища, у якому він діє; список можливих дій агента в деякий момент часу залежить від поточного стану середовища. Основною проблемою в роботі агента є вибір найбільш підходящої для поточних умов дії, яка б дозволила виконати завдання агента найкраще.

Основними моделями організації обчислень за допомогою багатоагентних систем в даний час є розподілений штучний інтелект і штучне життя.

Загальна модель розподіленого штучного інтелекту (РШІ) визначає рішення деякої загальної задачі за допомогою її декомпозиції на декілька підзадач, що вирішуються індивідуальними інтелектуальними агентами, і формування загального результату на основі аналізу локальних результатів, отриманих агентами (рис. 1). Розподілений штучний інтелект передбачає узгодження цілей, інтересів і стратегій окремих агентів та координацію їх діяльності.

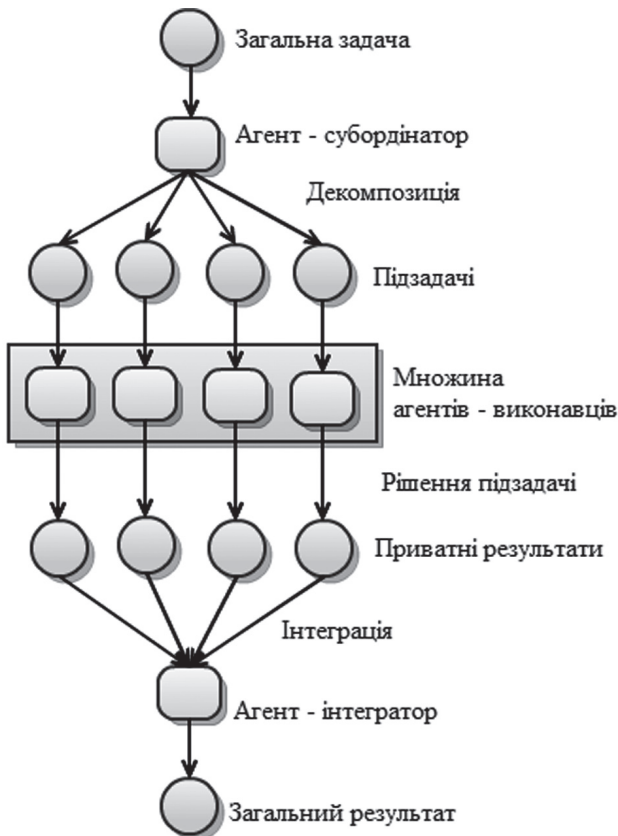


Рис. 1. Схема вирішення задачі методом розподіленого штучного інтелекту

Основною задачею програмної системи є своєчасне розпізнавання та інтерпретація траєкторій, представлених у вигляді сукупностей точок на візуальних відображеннях перетинів Пуанкаре.

Середовище моделювання Modeler надає чотири різних графічних відображення досліджуваної моделі (зовнішня і внутрішня секція, сфера і поверхня), які користувач може використовувати для дослідження таких траєкторій. Саме необхідність використання всіх чотирьох відображень моделі для розуміння загальної картини руху обумовлює застосування багатоагентного підходу і формує вигляд потрібної багатоагентної системи.

Очевидно, що «множинність» і неоднорідність процесу розпізнавання у даному випадку природньо вписується в багатоагентну парадигму – кожне відображення досліджуємої моделі може бути асоційоване з окремим агентом або спільнотою агентів. Це дозволяє розбити комплексний процес дослідження на простіші підзадачі, які розподіляються між спеціалізованими агентами. Дані агенти потім автономно вирішують виділені їм частини загальної задачі і формують локальні (приватні) результати, які використовуються для редукції загального (підсумкового) результату розпізнавання.

Таким чином, це обумовлює використання для дослідження і розпізнавання методу розподіленого штучного інтелекту. Існуючі на даний момент багатоагентні системи розпізнавання, як правило, побудовані схожим чином [5, 15, 16, 17, 19]; єдиним істотним відхиленням від даної схеми можна вважати лише роль агента-інтегратора, яка може бути відсутня, якщо агенти-виконавці досить самостійні для колективної редукції загального результату за допомогою переговорів або редукція просто не передбачена (як в Visent [16]).

Сам процес розпізнавання траєкторії на окремо взятому графічному відображенні моделі також може бути розбитий на підзадачі і, відповідно, виконуватися групою агентів.

Загалом траєкторії руху можна розділити на типи: регулярна, з малою хаотичністю, з великою хаотичністю, хаотична. Кожен з випадків володіє своїми відмінними візуальними характеристиками. Регулярна траєкторія являє собою концентричні замкнуті криві, траєкторія з малою хаотичністю – група неконцентричних замкнутих кривих, хаотична траєкторія – сукупність точок, що утворюють множину.

Таким чином, для розпізнавання кожного випадку може знадобитися окремий метод або навіть група методів (наприклад, різні методи для розпізнавання на площині і на поверхні об'ємної фігури). Це дозволяє виділити спеціалізовані типи агентів (A1, A2, A3, A4) для розпізнавання певних випадків траєкторії. У підсумку отримуємо локальну групу агентів-розпізнавачів (A11, A12, A13, A21 і т.д.) для

кожного з чотирьох графічних відображень моделі і додатковий рівень ієрархії в розподіленому процесі розпізнавання (рис. 2).

Також варто враховувати, що окрім розпізнавання траєкторій існують й інші задачі, які мають вирішувати, як розробляється багатоагентна система: визначення оптимальної початкової точки для прорахунку нової траєкторії, своєчасне припинення прорахунку поточної траєкторії і процесу дослідження моделі в цілому.

Для вирішення цих задач має сенс заснувати блок з одного або кількох агентів для забезпечення інтелектуального регулювання процесу дослідження. В рамках схеми роботи розподіленого штучного інтелекту такий блок або агент управління (АУ) повинен грати роль як агента-субординатора, так і агента-інтегратора, тобто формувати початкові дані та умови для роботи агентів-розпізнавачів та інтерпретувати отримані ними результати (P11, P12, P13 і т.д.).

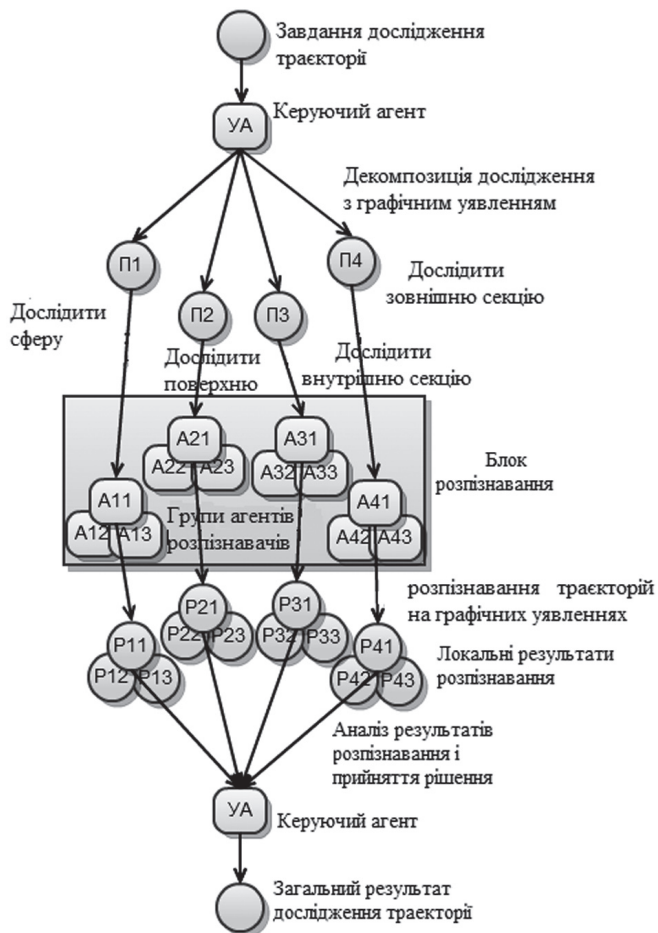


Рис. 2. Схема розпізнавання траєкторії за методом РШІ

В узагальненому вигляді роботу блоку управління можна розглядати як циклічний процес ініціації, моніторингу та регулювання прорахунку і розпізнавання окремої траєкторії.

Наступним аспектом роботи багатоагентної системи є ввід і вивід даних. У даному випадку вхідними даними є графічні відображення досліджуємої

моделі, що генеруються Modeler, а вихідними – опис цієї моделі, отриманий в результаті її дослідження. Для забезпечення модульності і уніфікації вводу/виводу даних має сенс організувати інтерфейсний агент.

Інтерфейсні агенти досить часто зустрічаються в багатоагентних системах [5, 15, 19], відіграючи роль «заглушок» (stubs) між системою і зовнішнім середовищем. Задача такого агента – надання абстрактного інтерфейсу для вводу/виводу даних, що не залежить від конкретного джерела або приймача даних. Це дозволяє організувати уніфікований ввід/вивід, забезпечує адаптивність і розширюваність системи в рамках її зв'язків із зовнішнім середовищем. Більш того, багатоагентна система може містити кілька інтерфейсних агентів, зокрема для отримання інформації з різних джерел.

В розроблюваній системі інтерфейсний агент перш за все може бути використаний для посилки запитів і команд Modeler, отримання від нього даних, перетворення даних в структурований вигляд і обробки можливих помилок комунікації.

Також має сенс виділити в системі окремого агента для передобробки вхідних даних. Таких агентів можна зустріти в деяких багатоагентних системах розпізнавання [15, 19] поряд з інтерфейсними агентами. Як правило, агент передобробки даних служить для перетворення «сирих» даних, отриманих безпосередньо від інтерфейсного агента, в систематизовану та структуровану інформацію з метою подальшого використання іншими компонентами багатоагентної системи.

В розроблюваній системі може знадобитися приводити отримані з Modeler зображення досліджуваної моделі до такого вигляду, в якому їх буде зручно обробляти агентам-розпізнавачам. Зокрема, перед розпізнаванням окремої траєкторії має сенс прибрати з зображення всі «зайві» елементи: сітку та інші статичні елементи, точки знайдених раніше траєкторій та інше. Також залежно від обраних методів розпізнавання може знадобитися масштабувати зображення, змінювати спосіб його кольорового кодування, апроксимувати лінії траєкторій і т.п.

Таким чином, усі перераховані задачі можна покласти на спеціалізованого агента передобробки даних або, скоріше, на інформаційного агента. Такий агент буде надавати високорівневий доступ до даних Modeler за допомогою звернення до інтерфейсного агента.

Інформаційний агент повинен виконувати наступне:

- 1) одержувати високорівневі запити від інших агентів системи;
- 2) посилати низькорівневі запити інформаційного агента для збору даних, необхідних для виконання запиту;

3) отримувати та обробляти надіслані інтерфейсним агентом дані;

4) вислати отриману інформацію замовникові в необхідному форматі.

Таким чином, укрупнена структура розроблюваної багатоагентної системи включає в себе наступні компоненти (рис. 3): інтерфейсний агент, інформаційний агент, блок розпізнавання, блок управління.

Призначення перерахованих компонентів наступне:

1) інтерфейсний агент служить для забезпечення взаємодії з Modeler;

2) інформаційний агент використовується для передобробки графічних даних і організації високорівневого доступу до них;

3) блок розпізнавання виконує розпізнавання траєкторій на зображеннях моделі (внутрішньої і зовнішньої секції, сфери і поверхні);

4) блок управління координує загальний процес дослідження.

Вхідними даними БАС є графічні відображення досліджуємої моделі (зображення траєкторій руху), що генеруються в середовище Modeler.

Вихідними даними системи є база даних, що містить аналітичну і статистичну інформацію, отриману в процесі дослідження моделі, зокрема опис знайдених траєкторій.

4. Розробка структури бази даних

В розроблюваній багатоагентній системі база даних містить інформацію щодо дослідження моделі. В процесі роботи БАС база даних є проміжним результатом дослідження, який керуючий агент використовує для прийняття рішення щодо класифікації системи та регулювання інтенсивності процесу розпізнавання. Після закінчення функціонування

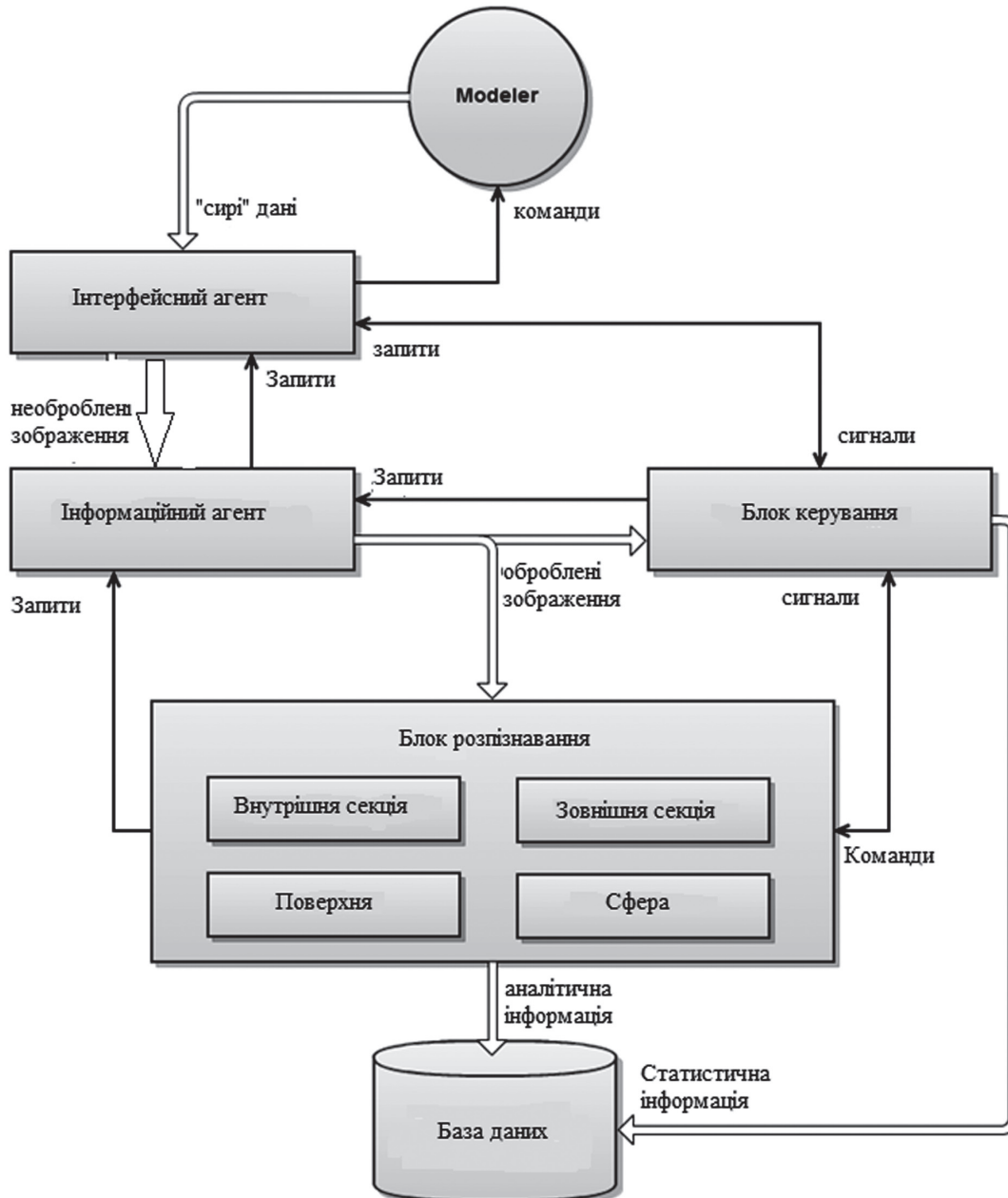


Рис. 3. Загальна схема розроблюємої багатоагентної системи

багатоагентної системи база даних являє собою загальний результат дослідження, що надається користувачеві.

У загальному випадку база даних може містити такі дані про траєкторію:

- тип (регулярна, з малою хаотичністю, хаотична);
- початкова точка розрахунку;
- точність кінцевого результату розпізнавання
- обсяг часу, витрачений на розрахунок і розпізнавання;
- склад активних розпізнавачів на момент припинення розрахунку;
- кінцеві результати всіх задіяних агентів-розпізнавачів;
- зображення кінцевого вигляду траєкторії на різних графічних відображеннях моделі;

– оцінена швидкість приросту точок на початку і наприкінці прорахунку.

Структура бази даних наведена на рис. 4.

5. Опис агентів та кооперативна взаємодія між ними

Розроблена система складається з сімнадцяти агентів: інтерфейсного агента, інформаційного агента, агента-планувальника, керуючого агента, тактового агента і дванадцяти спеціалізованих агентів-розпізнавачів.

Інтерфейсний агент є відповідальним за взаємодію системи із середовищем моделювання Modeler. Він надає іншим агентам системи інтерфейс для передачі в Modeler запитів, команд і отримання згенерованих в ньому зображень, а також забезпечує обробку помилок комунікації.

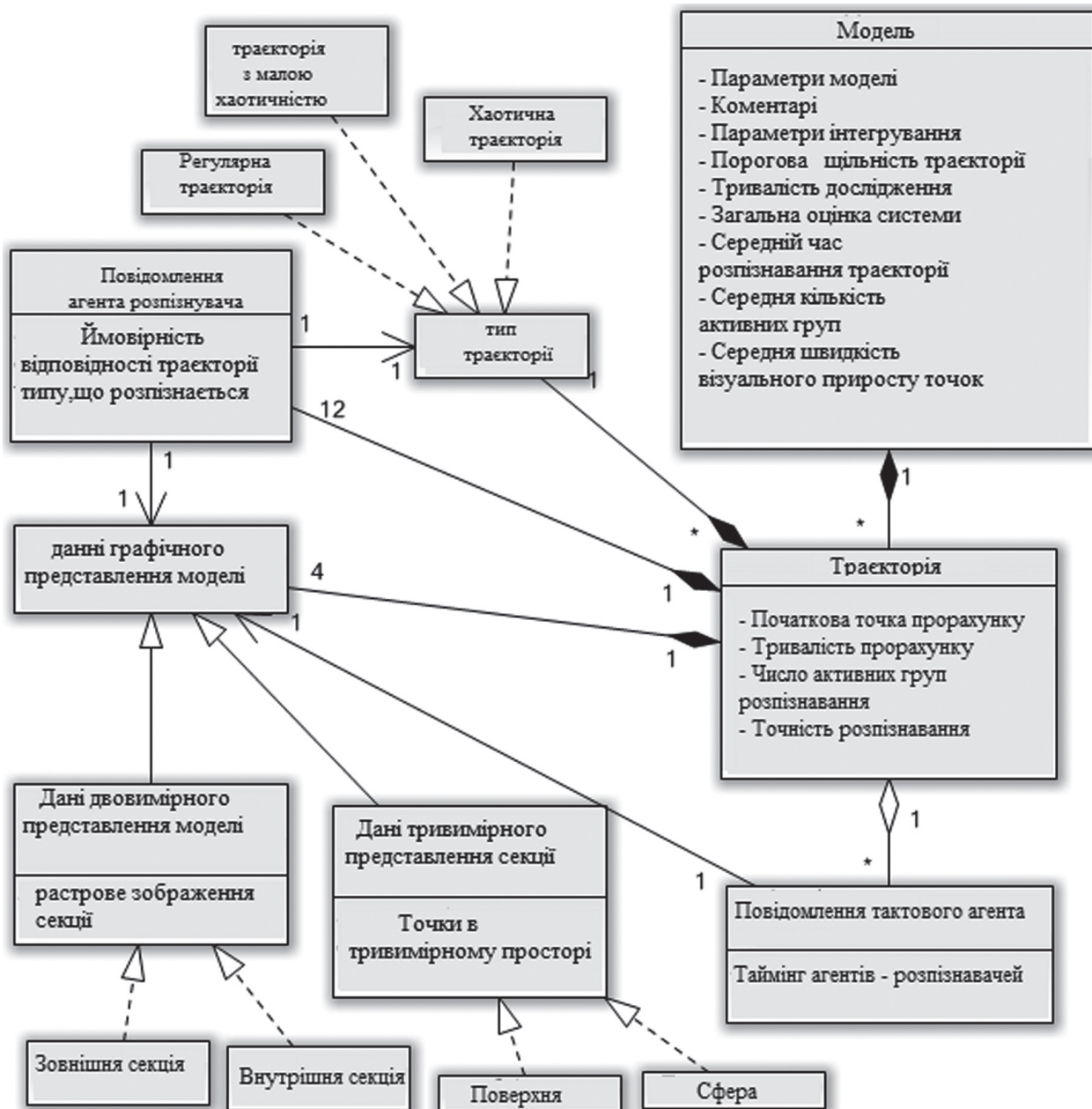


Рис. 4. Схема структури бази даних процесу дослідження

Інформаційний агент надає високорівневий інтерфейс до даних, отриманих з Modeler за допомогою інтерфейсного агента. Він відповідальний за перетворення формату і предобробку графічних даних, фільтрацію і кешування запитуваних зображень.

Агент-планувальник вирішує задачу вибору початкової точки для прорахунку нової траєкторії, базуючись на розташуванні знайдених раніше траєкторій. Також цей агент визначає, коли знайдено досить траєкторій і дослідження моделі можна завершити.

Агент-розпізнавач виконує розпізнавання одного типу траєкторії (регулярна, з малою хаотичністю, хаотична) на певному графічному відображенні моделі (зовнішній чи внутрішній секції, сфері або поверхні). Всі агенти-розпізнавачі організовані в чотири групи, по три агенти на кожне відображення моделі. Результатом роботи агента даного класу є ймовірність відповідності поточної траєкторії типу, що розпізнається.

Тактовий агент регулює частоту періодичного спрацьовування агентів-розпізнавачів в кожній окремій групі розпізнавання в залежності від швидкості приросту точок на підконтрольних групі зображеннях моделі.

Керуючий агент координує роботу агентів системи в цілому. Зокрема, він відповідальний за

регулювання розрахунку і розпізнавання окремої траєкторії, прийняття рішення щодо її класифікації та занесення відомостей про знайдені траєкторії в базу даних системи.

Узагальнена схема взаємодії агентів в системі приведена на рис. 5.

Передбачається, що розроблювана БАС повинна бути релізована у складі окремого програмного забезпечення з графічним інтерфейсом користувача, що взаємодіє з середовищем Modeler за допомогою стандартних комунікаційних засобів операційної системи (сокети, канали, міжпроцесні повідомлення).

Також варто зазначити, що структура спроектованих агентів передбачає, що вони будуть сполучатися за принципом сигналів і слотів, тобто за допомогою асинхронного зворотного виклику обробників подій.

Для звернення до інших агентів (передачі запиту на отримання даних або виконання дії) у кожного агента передбачено набір відповідних методів-сигналів.

У свою чергу, для отримання таких запитів від інших агентів використовується набір методів-слотів, що спрацьовують при виникненні пов'язаних з ними сигналів. Кожен сигнал може бути підключений до необмеженої кількості слотів, і навпаки.

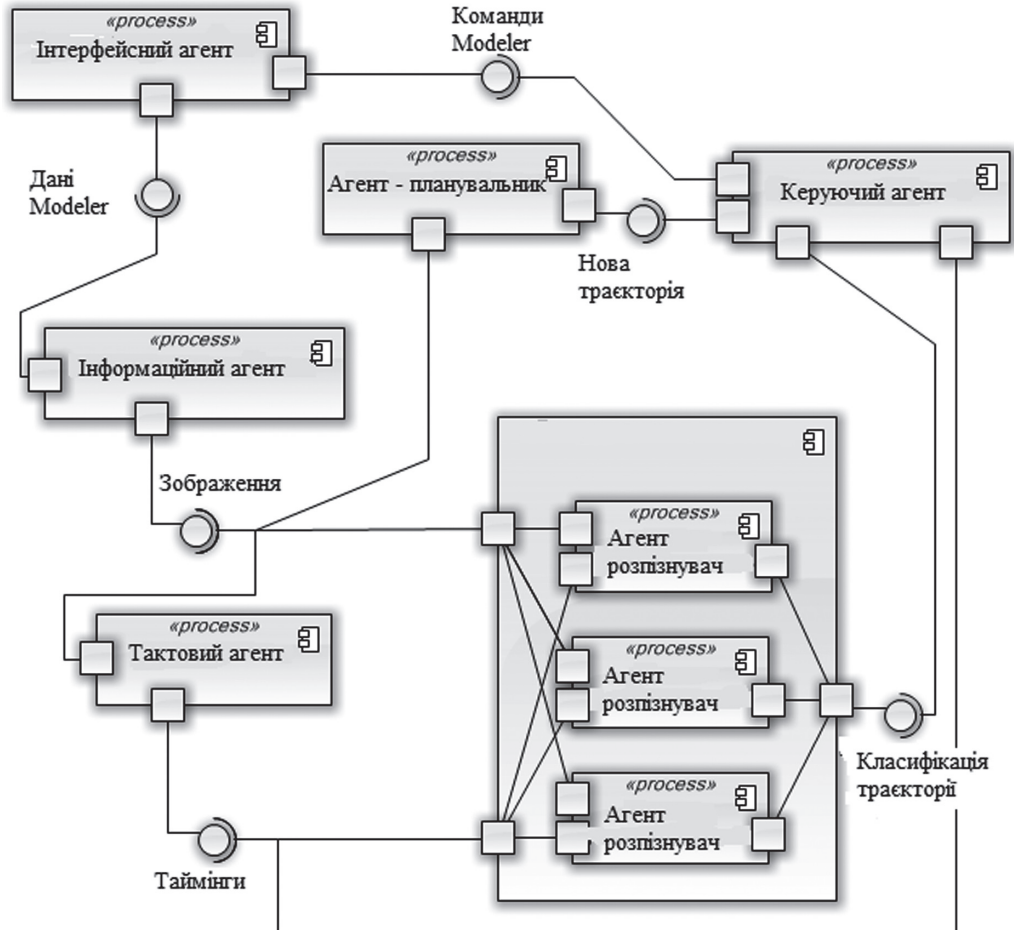


Рис. 5. Діаграма компонентів БАС

Для з'єднання сигналу зі слотом їх сигнатури повинні бути сумісні (слот повинен приймати на вхід тільки параметри, зазначені в сигналі).

Структура і склад класів агентів наведені на рис. 6.

Застосування такої схеми взаємодії дозволяє домогтися гнучкості, простоти і абстракції в організації передачі повідомлень між автономними агентами в багатоагентній системі.

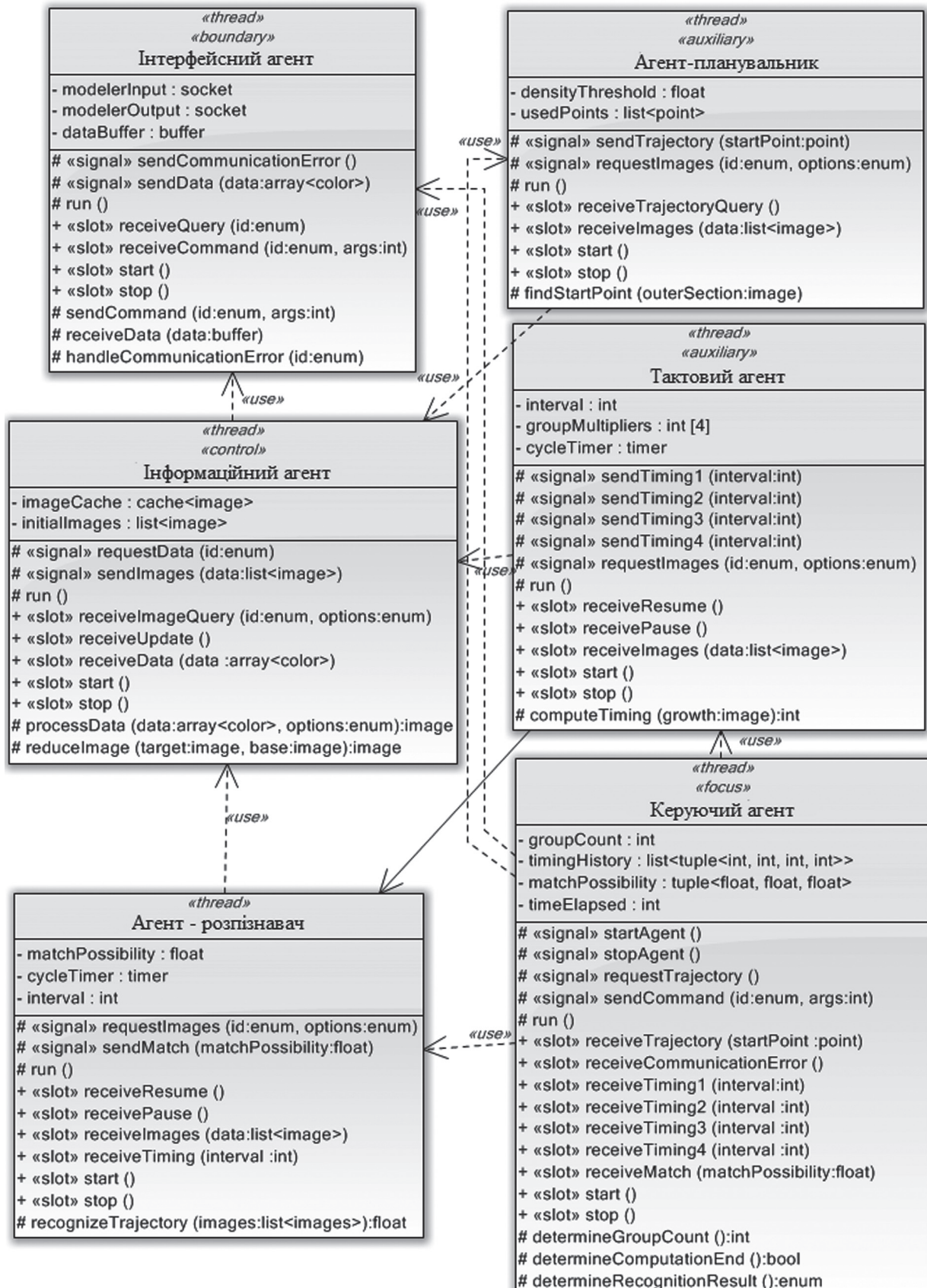


Рис. 6. Діаграма класів БАС

Таким чином, оптимальною низькорівневою основою для реалізації спроектованих агентів є потоки (threads), засновані на транзакціях, тобто працюючі в циклі обробки подій (events). Виділення агентів в окремі процеси в даному випадку найімовірніше буде зайвим, оскільки додаткові витрати і затримки при обміні даними між такими агентами навряд чи будуть компенсовані розподілом обчислювального навантаження між декількома машинами в мережі.

Реалізація системи можлива і за допомогою застосування будь-якого високорівневого набору програмних засобів, що надає готову основу для створення агентів і організації їх спільної роботи (агентного фреймворку).

Висновки

В даній роботі була розроблена багатоагентна система для вирішення задачі дослідження моделі руху на основі методу розподіленого штучного інтелекту з використанням даних, одержаних з середовища моделювання Modeler, в режимі реального часу. Була складена загальна схема вирішення задачі, спроектовані агенти багатоагентної системи, розроблена структура бази даних, описана загальна архітектура багатоагентної системи.

Результатом розробки є специфікація потрібної багатоагентної системи, складаючих її агентів та бази даних у вигляді діаграм UML, а також опис структури, функцій і порядку роботи системи та примітки щодо її програмної реалізації.

Одержане рішення має наступні переваги:

- повна автоматизація процесу знаходження траєкторій;

- висока ефективність розпізнавання траєкторій завдяки використанню багатьох розпізнавачів статистичних даних, накопичуваних під час дослідження;

- наявність механізмів саморегуляції для мінімізації похибки розрахунку траєкторій та оптимізації споживання системних ресурсів.

Розроблена схема багатоагентної інтелектуальної системи може в подальшому бути реалізована у складі прикладного програмного забезпечення. Це програмне рішення може застосовуватися для підвищення ефективності та автоматизації певних видів діяльності, пов'язаних з дослідженням моделей динамічних систем, спільно з середовищем моделювання Modeler або іншими подібними програмними продуктами.

Список літератури: 1. Ручкин К.А. Разработка мультиагентной системы для прогнозирования поведения динамической системы в режиме реального времени / К.А. Ручкин, А.В. Данилов // Искусственный интеллект. – 2011. – №4. – С. 192–199. 2. Ручкин К.А. Разработка компьютерной системы для построения и анализа сечений Пуанкаре / К.А. Ручкин // Искусственный интеллект. – 2009. – №1.

– С. 300–304. 3. Decker K. Extending a Multi-Agent System for Genomic Annotation / K. Decker, S. Khan, C. Schmidt, D. Michaud // AAMAS 2008. – 2008. – pp. 13–25. 4. Rascol V. CASSIOPE: An expert system for conserved region searches / V. Rascol, A. Levasseur, O. Chabrol, S. Grusea, P. Gouret, E. Danchin, P. Pontarotti // BMC Bioinformatics 2009. – 2009. – pp. 128–135. 5. Addis A. From a Generic MultiAgent Architecture to MultiAgent Information Retrieval Systems / A. Addis, G. Armano, E. Vargiu // AAMAS 2008. – 2008. – pp. 48–61. 6. Kanteev M. Multi-Agent Meta-Search Engine Based on Domain Ontology / M. Kanteev, I. Minakov, G. Rzevski, P. Skobelev, S. Volman Skobelev // AAMAS 2007. – 2007. – 13c. 7. Rao V. S. Distributed Agent Mining and Agent Mining Interaction and Integration: A Novel Approach / V. S. Rao, S. Vidiyavathi, G. Ramaswamy // IJRRAS 4. – 2010. – pp. 57–69. 8. Rao V. S. Multi Agent-Based Distributed Data Mining: an Over View / V.S. Rao // International Journal of Reviews in Computing. – 2010. – №5. – pp. 67–82. 9. Sajja P. S. Multi-Agent System for Knowledge-Based Access to Distributed Databases / P. S. Sajja // Interdisciplinary Journal of Information, Knowledge, and Management. – 2008. – №3. – pp. 29–41. 10. Rzevski, G. Magenta Technology: A Family of Multi-Agent Intelligent Schedulers / G. Rzevski, J. Himoff, P. Skobelev. – 2006. – 11c. 11. Camilio C. A Multi-Agent Framework for E-Commerce Automated Negotiation Protocols Evaluation / C. Camilio, R. Nogueira, C. Vinhal // AAMAS 2009. – 2009. – pp. 29–35. 12. DiPippo L. C. A Real-Time Multi-Agent System Architecture for E-Commerce Applications / L. C. DiPippo, V. Fay-Wolfe, L. Nair, E. Hodys, O. Uvarov // AAMAS 2006. – 2006. – pp. 47–58. 13. Pour G. Multi-Agent System Architecture for Quality Control: Moving Toward Autonomic System for E-Commerce / G. Pour // IADIS International Conference e-Commerce 2004. – 2004. – pp. 125–141. 14. Rahal D. Multi-Agent System for Modeling Transport Systems / D. Rahal, F. Rahal, M. Chekroun // European Journal of Scientific Research. – 2010. – №1. – pp. 80–89. 15. Casella J. A Multi-Agent System for Hand-drawn Diagram Recognition / J. Casella, V. Deufemia, V. Mascardi // AAMAS 2007. – 2007. – pp. 54–59. 16. Hempel O. Agent-Based Object Recognition / O. Hempel, U. Boker, G. Hartmann // ICEIS 2000. – 2000. – pp. 33–40. 17. Juchmes R. A Multi-Agent System for the Interpretation of Architectural Sketches / R. Juchmes, P. Leclercq, S. Azar // EUROGRAPHICS 2004. – 2004. – pp. 79–87. 18. Rzevski G. Dynamic Pattern Discovery using Multi-Agent Technology / G. Rzevski, P. Skobelev, I. Minakov, S. Volman Skobelev // TELE_INFO 07. – 2007. – 9c. 19. Salama G. A Distributed Multi-Agent Model for Face Recognition over Network / G. Salama, A. Fahmy, M. Elbar // ASAT-13. – 2009. – pp. 75–92. 20. Zarandi M. A Multi-Agent Expert System for Steel Grade Classification Using Adaptive Neuro-fuzzy Systems / M. Zarandi, M. Avazbeigi, M. Anssari, B. Ganji // Expert systems. – 2010. – №1. – pp. 161–180. 21. Sas C. Virtual environment trajectory analysis: a basis for navigational assistance and scene adaptivity / C. Sas, G. O'Hare, R. Reilly // Future Generation Computer Systems. – 2005. – №4. – pp. 19–32. 22. Brazier M. T. DESIRE: Modeling Multi-Agent Systems in a Compositional Formal Framework / M. T. Brazier, B. M. Dunin-Keplicz, N. R. Jennings // International Journal of Cooperative Information Systems. – 1997. – №1. – pp. 66–76. 23. Bauer B. UML 2.0 and Agents: How to Build Agent-based Systems with the new UML Standard / B. Bauer, J. Odell // Interdisciplinary Journal of Information, Knowledge, and Management.

– 2010. – №5. – pp. 15-28. **24. Kinny D.** A Methodology and Modelling Technique for Systems of BDI Agents / D. Kinny, M. Georgeff, A. Rao // AAMAS 2001. – 2001. – pp. 101-117. **25. Wooldridge M.** The Gaia Methodology for Agent-Oriented Analysis and Design / M. Wooldridge, N. R. Jennings, D. Kinny // Knowledge Engineering Review. – 1997. – №9. – pp. 27-51. **26. Mylopoulos J.** TROPOS: A Framework for Requirement-Driven Software Development / J. Mylopoulos, J. Castro // Lecture Notes in Computer Science. – 2000. – pp. 55-68. **27. Bernon C.** ADELFE, a Methodology for Adaptive Multi-Agent Systems Engineering / C. Bernon, M. Gleizes, S. Peyruqueou, G. Picard // AAMAS 2006. – 2006. – pp. 95-113. **28. Omicini A.** SODA: Societies and Infrastructures in the Analysis and Design of Agent-based Systems/ A. Omicini // AAMAS 2005. – 2005. – pp. 37-51.

Поступила в редколегію 25.02.2013

УДК 681.3, 004.85, 004.89

Анализ и проектирование многоагентной системы прогнозирования поведения нелинейной динамической системы в реальном времени / К.А. Ручкин / / Бионика интеллекта: науч.-техн. журнал. – 2013. – № 1 (80). – С. 117-127.

В работе продолжены исследования [1], связанных с разработкой многоагентной системы прогнозирования поведения нелинейной динамической системы в реальном времени (ППНДСРЧ). Составлена общая схема решения задачи, описана общая архитектура многоагентной системы, разработана структура базы данных, спроектированы агенты многоагентной системы. Предложенная многоагентная система спроектирована на основе метода распределенного искусственного интеллекта с использованием данных, полученных из среды

моделирования Modeler [2] в режиме реального времени. Система состоит из семнадцати агентов: интерфейсного агента, информационного агента, агента-планировщика, управляющего агента, тактового агента и двенадцати специализированных агентов-распознавателей. Приведены спецификацию многоагентной системы, ее агентов и базы данных в виде диаграмм UML, а также описание структуры, функций и порядка работы системы и замечания по ее программной реализации в будущем.

Ил. 6. Библиогр.: 28 назв.

UDC 681.3, 004.85, 004.89

Analysis and design of multi-agent system predict the behavior of nonlinear dynamical systems in real time / К.А. Ручкин // Bionics of Intelligence: Sci. Mag. – 2013. – № 1 (80). – P. 117-127.

In the continuation of the study [1], are associated with the drafting of multi-agent system predict the behavior of nonlinear dynamic systems in real time (PPNDSRCH). Ckladena general scheme for solving the problem described overall multi-agent system architecture, developed database structure designed multi-agent system. Over-the proposed multi-agent system is designed based on the method of distributed artificial intelligence using data obtained from the simulation environment Modeler [2] in real time. The system consists of seventeen agents: interface agent, information agent, agent scheduling, managing agent, agent clock and twelve-tsyaty specialized agents recognizers. An specificity certification for multi-system, its agents and database diagrams in UML, and a description of the structure, functions, and on-line system operation and notes about its software implementation in the future.

Fig. 6. Ref.: 28 items.