НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ННК «Інститут прикладного системного аналізу»
(повна назва інституту/факультету)

Системного проектування
(повна назва кафедри)

«На правах рукопису»
УДК 004.82

«До захисту допущено»
Завідувач кафедри
_____ А.І. Петренко
(підпис) (ініціали, прізвище)
“___”_____ 2017 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 8.05010103 Системне проектування
(код і назва)

на тему: Технології формування знань та обміну знаннями в інтелектуальних комп’ютерних системах

Виконала: студентка VI курсу, групи ДА-52м
(шифр групи)

Науменко Тетяна Олександрівна _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник професор, д.т.н., Рогоза В.С. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант Розроблення стартап-проекту проф., д.т.н., Рогоза В.С. _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2017 року

Національний технічний університет України
«Київський політехнічний інститут»

Інститут (факультет) ННК «Інститут прикладного системного аналізу
(повна назва)

Кафедра Системного проектування
(повна назва)

Рівень вищої освіти Другий (Магістерський)

Спеціальність 8.05010103 Системне проектування
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ А.І. Петренко
(підпис) (ініціали, прізвище)
«___»_____2017 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Науменко Тетяні Олександрівні
(прізвище, ім'я, по батькові)

1. Тема дисертації «Технології формування знань та обміну знаннями в інтелектуальних комп'ютерних системах»

науковий керівник дисертації Рогоза В.С., д.т.н., професор,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «20» березня 2017 р. №32-ст

2. Термін подання студентом дисертації 01.06.2017

3. Об'єкт дослідження процес керування та самостійного функціонування групи інтелектуальних об'єктів «Розумного будинку»

4. Предмет дослідження Мультиагентна система як засіб для організації роботи «Розумного будинку»

5. Перелік завдань, які потрібно розробити
1. Аналіз використання онтологій в побудові інформаційних систем, що накопичують та обмінюються знаннями на прикладі мультиагентних систем.
2. Побудова онтології для накопичення та обміну знаннями.
3. Створення мультиагентної системи та впровадження в неї онтології.

4. Створення системи керування смарт-елементами «розумного будинку» на основі мультиагентної системи
5. Тестування створеної системи на основі розроблених сценаріїв
6. Оформлення роботи на основі отриманих результатів

6. Орієнтовний перелік публікацій

1. Naumenko T.O. The analysis of fields of using ontologies in the construction of information systems / Tetiana Naumenko // 19-th International conference on System Analysis and Information Technology SAIT 2017, Kyiv, Ukraine , May 22 – 25, 2017. – 2017. – p.234-235
2. Науменко Т.О. Використання онтологій для зберігання та обміну заннями в мультиагентних системах / Науменко Тетяна Олександрівна // Міжнародний науковий журнал «Інтернаука». – 2017. – №6(28). – 2017. – с. 50-52

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розробка стартап-проекту	Рогоза В.С. професор		
Основна частина	Рогоза В.С. професор		

8. Дата видачі завдання 01.02.2017

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
	Отримання завдання	01.02.2017	
	Збір інформації та аналіз літератури	15.02.2017	
	Розробка онтологій	27.03.2017	
	Розбробка мультиагентної системи	09.04.2017	
	Розробка системи керування елементами «розумного будинку»	26.04.2017	
	Тестування розроблених сценаріїв	13.05.2017	
	Оформлення дипломної роботи	30.05.2017	
	Отримання допуску до захисту та подача роботи в ДЕК	08.06.2017	

Студент _____ Т.О. Науменко
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ В.С. Рогоза
(підпис) (ініціали, прізвище)

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Технології формування знань та обміну знаннями в інтелектуальних комп'ютерних системах

студенткою: Науменко Тетяною Олександрівною

Робота виконана на 110 сторінках, містить 24 ілюстрації, 23 таблиць. При підготовці використовувалася література з 74 різних джерел.

Актуальність

У зв'язку зі стрімким розвитком систем, заснованих на знаннях, які займають широку область в інформаційних технологіях, формуючи власні методи і принципи, значного розвитку зазнали й інформаційні системи, що накопичують знання та обмінюються ними. Одним із прикладів таких систем, є мультиагентні системи.

Отже дослідження мультиагентних систем, їх застосування у різних галузях науки та промисловості, а також способи формування та обміну знаннями в них є актуальним напрямком досліджень на сьогоднішній день.

Не можна не звернути увагу на те, що розвиток технологій досяг настільки високого рівня, що смарт-пристроїв стає все більше у будинках користувачів. Тому ще одним важливим і актуальним питанням є покращення систем керування інтелектуальними елементами «Розумного будинку».

Мета і завдання

Метою магістерської дисертації є дослідження мультиагентних систем, їх застосування у різних галузях науки та промисловості, способів формування та обміну знаннями в них за допомогою онтологій, а також створення системи керування інтелектуальними об'єктами «Розумного будинку». Для цього визначено наступні завдання:

1. Дослідити онотології, основні елементи онтологій, принципи проектування онтологій.
2. Дослідити поняття агента та мультиагентні системи.
3. Дослідити застосування онтологій в мультиагентних системах.
4. Побудувати онтології для накопичення та обміну знаннями.
5. Створити мультиагентну системи із застосуванням онтології.
6. Створити систему керування смарт-елементами «Розумного будинку» на основі мультиагентної системи.

Рішення поставлених завдань та досягнуті результати

В роботі були розглянуті та проаналізовані поняття інформаційних комп’ютерних систем, застосування в них мультиагентних систем, поняття онтології та засоби її впровадження в побудову мультиагентних систем. Було побудовано онтології, створена мультиагентна система із вбудованою в неї онтологією. Також створено програмне забезпечення керування інтелектуальними об’єктами «Розумного будинку». Також така система забезпечує спілкування між агентами та самостійність у прийнятті рішень щодо функціонування.

Розроблену систему було протестовано на попередньо розроблених сценаріях, що перевіряють головні вимоги, щодо нового підходу в даній області. Тестування проводилося з використанням таких смарт-девайсів, як пилосос, телевізор та холодильник. Результатом є підтвердження теорії щодо полегшення керування такою системою, зменшення недоліків у роботі кожного пристрою та можливість їх самостійного адаптивного функціонування.

Об’єкт досліджень

Об’єктом дослідження обрано процес керування та самостійного функціонування групи інтелектуальних об’єктів «Розумного будинку»

Предмет досліджень

Предметом дослідження обрано мультиагентну система як засіб для організації роботи «Розумного будинку»

Методи досліджень

Пошук і аналіз інформації про побудову, розвиток, сучасний стан, проблеми та перспективи застосування мультиагентних систем та використання в них онтологій; синтез для винайдення рішення поставленої задачі; спостереження, дедукція та порівняння для визначення необхідних функцій системи, що покращать роботу; експеримент для тестування.

Наукова новизна

Наукова новизна роботи полягає у винайдені нового підходу до створення системи керування інтелектуальними об’єктами «Розумного будинку», що покращує керування ними та забезпечує самостійність та адаптивність функціонування.

Практичне значення

Отримані результати можуть використовуватися у майбутніх дослідженнях за напрямками: вдосконалення систем керування «Розумним будинком» та розповсюдження нової технології у підрозділи «Інтернету речей». Також використання обраного підходу значно полегшить користувачам процес керування інтелектуальними пристроями та завдяки самостійності об’єктів заощадить час власників.

Апробації результатів дисертації

Результати досліджень оприлюднені на 19-й Міжнародній науково-технічній конференції SAIT 2017, а також – у міжнародному науковому журналі «Інтернаука», випуск №6 2017 року.

Ключові слова:

Інформаційна система, агент, мультиагентна система, онтологія, інтелектуальний пристрій, «Розумний будинок».

РЕФЕРАТ

на магистерскую диссертацию

выполнена на тему: Технологии формирования знаний и обмена знаниями в интеллектуальных компьютерных системах

студенткой: Науменко Татьяной Олександровной

Робота виконана на 110 сторінках, містить 24 ілюстрації, 23 таблиць. При підготовці використовувалася література з 74 різних джерел.

Актуальность

В связи со стремительным развитием систем, основанных на знаниях, которые занимают широкую область в информационных технологиях, формируя собственные методы и принципы, значительное развитие получили и информационные системы, накапливают знания и обмениваются ими. Одним из примеров таких систем является мультиагентные системы.

Итак исследования мультиагентных систем, их применение в различных областях науки и промышленности, а также способы формирования и обмена знаниями в них является актуальным направлением исследований на сегодняшний день.

Нельзя не обратить внимание на то, что развитие технологий достиг столь высокого уровня, что смарт-устройств становится все больше в домах пользователей. Поэтому еще одним важным и актуальным вопросом является улучшение систем управления интеллектуальными элементами «Умного дома».

Цель и задания

Целью магистерской диссертации является исследование мультиагентных систем, их применение в различных областях науки и промышленности, способов формирования и обмена знаниями в них с помощью онтологий, а

также создание системы управления интеллектуальными объектами «Умного дома». Для этого определены следующие задачи:

1. Исследовать агенты и мультиагентные системы.
2. Исследовать онотологии, основные элементы онтологий, принципы проектирования онтологий.
3. Исследовать применение онтологий в мультиагентных системах.
4. Построить онтологии для накопления и обмена знаниями.
5. Создать мультиагентную системы с применением онтологии.
6. Создать систему управления смарт-элементами «Умного дома» на основе мультиагентной системы.

Решения поставленных заданий и достигнутые результаты

В работе были рассмотрены и проанализированы понятие информационных компьютерных систем, применение в них мультиагентных систем, понятие онтологии и средства ее внедрения в построение мультиагентных систем. Было построено онтологии, создана мультиагентная система со встроенной в нее онтологии. Также создано программное обеспечение управления интеллектуальными объектами «Умного дома». Также такая система обеспечивает общение между агентами и самостоятельность в принятии решений по функционированию.

Разработанную систему было протестировано на предварительно разработанных сценариях, проверяющие главные требования, предъявляемые к новому подходу в данной области. Тестирование проводилось с использованием таких смарт-девайсов, как пылесос, телевизор и холодильник. Результатом является подтверждение теории по облегчению управления такой системой, уменьшение недостатков в работе каждого устройства и возможности их самостоятельного адаптивного функционирования.

Объект исследований

Объектом исследования избран процесс управления и самостоятельного функционирования группы интеллектуальных объектов «Умного дома»

Предмет исследований

Предметом исследования избран мультиагентную система как средство для организации работы «Умного дома»

Методы исследований

Поиск и анализ информации об устройстве, развитие, современное состояние, проблемы и перспективы применения мультиагентных систем и використатння в них онтологий; синтез для нахождения решения поставленной задачи; наблюдения, дедукция и сравнения для определения необходимых функций системы, улучшат работу; эксперимент для тестирования.

Научная новизна

Научная новизна работы заключается в изобретены нового подхода к созданию системы управления интеллектуальными объектами «Умного дома», что улучшает управление ими и обеспечивает самостоятельность и адаптивность функционирования.

Практическое значение

Полученные результаты могут использоваться в будущих исследованиях по направлениям: совершенствование систем управления «Умным домом» и распространение новой технологии в подразделения «Интернета вещей». Также использование выбранного подхода значительно облегчит пользователям процесс управления интеллектуальными устройствами и благодаря самостоятельности объектов сэкономит время владельцев.

Апробации результатов диссертации

Результаты исследований опубликованы в 19-й Международной научно-технической конференции SAIT 2017, а также - в международном научном журнале «Интернаука», выпуск №6 2017 года.

Ключевые слова:

Информационная система, агент, мультиагентная система, онтология, интеллектуальное устройство, «Умный дом».

ABSTRACT

On master's thesis

on topic: Technologies of knowledge forming and exchange in intelligent computer systems

student: Naumenko Tetiana

Work carried out on 110 pages containing 24 figures, 23 tables. The paper was written with references to 74 different sources.

Topicality

Due to the rapid development of knowledge-based, which occupy a wide area of information technology, creating their own methods and principles underwent significant development of information systems that accumulate knowledge and sharing. One example of such systems is a multi-agent system.

Thus the study of multi-agent systems and their applications in various fields of science and industry, as well as methods of formation and sharing of knowledge they have important area of research to date.

It is impossible not to draw attention to the fact that technological development has reached such a high level that smart devices is becoming increasingly in homes of users. Therefore, another important and urgent issue is to improve the management of intellectual elements of "Smart home".

Aims and objectives

The purpose of the master's thesis is to study multi-agent systems and their applications in various fields of science and industry, methods development and knowledge sharing in them using ontologies, and creating intelligent facilities management system "smart home". The following tasks are proposed:

1. Investigate the agents and multi-agent systems.

2. Investigate onotolohiyi essential elements of ontologies, ontology design principles.
3. To investigate the use of ontologies in multi-agent systems.
4. Build an ontology for collecting and sharing knowledge.
5. Create multi-agent systems using ontology.
6. Create a smart control system elements of "Smart home" based on multi-agent system.

The solutions of the tasks and results

The paper reviewed and analyzed the concept of computer information systems, they use multi-agent systems, ontology concepts and tools for its implementation in the construction of multi-agent systems. It was built ontologies created multiagent system with built-in ontology it. The software also controls intelligent objects "smart home". Also, this system provides communication between agents and independence in making decisions on operations.

The developed system was tested on pre-designed scenarios that examine major requirements for a new approach in this area. Testing was conducted using such smart of devices like vacuum cleaner, TV and refrigerator. The result is a confirmation of the theory to facilitate the management of such a system, reducing defects in the work of each device and the possibility of self-adaptive functioning.

The object of research

The object of study of the process control and independent operation group of intelligent objects of "Smart home"

Subject of research

The subject of study chosen multi-agent systems as a means for the organization of the "smart home"

Research methods

Search and analysis of information on the construction, development, current status, problems and prospects of multi-agent systems and vykorysatnnya they ontologies; Synthesis invention for solving the task; observation, deduction and comparison to determine necessary system functions that improve work; Experiment to test.

Scientific novelty

Scientific novelty invented is a new approach to creating intelligent facilities management system "smart home" that improves management and ensures the independence and adaptive functioning.

The practical value of the research

The results obtained can be used in future studies in the areas of: improving the management system "Smart Home" and the spread of new technology in the "Internet of things". Also, the use of the chosen approach will greatly facilitate the users' management of intelligent devices and, due to the independence of objects, will save time for owners

Research results approbation

The research results presented at the 19th International Scientific Conference SAIT 2017, and - in the international scientific journal "Internauka", Issue №6 2017.

Keywords

Information system, agent, multi-agent system, ontology, intelligent device "Smart House".

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ІС	інформаційна система
ПЗ	програмне забезпечення
МАС	мультиагентна система
ІС	інтелектуальна система
DL	дескрипційна логіка
АП	агентна платформа
GADT	Global Agent Descriptor Table
LADT	Local Agent Descriptor Table
БЛА	безпілотний літальний апарат
RDF	Resource Description Framework
OWL	Web Ontology Language
JADE	Java Agent Development Framework
ПМ-мережі	мережі потреб та можливостей

ВСТУП

У зв’язку зі стрімким розвитком систем, заснованих на знаннях, які займають широку область в інформаційних технологіях, формуючи власні методи і принципи, значного розвитку зазнали й інформаційні системи, що накопичують знання та обмінюються ними.

Побудова і впровадження інтелектуальних систем, що ґрунтуються на формалізації та повторному використанні знань, є перспективним напрямом практичного застосування методів штучного інтелекту у програмних системах. В основу таких систем покладено формалізоване подання знань про предметну область, наприклад, у формі онтології. Визначення онтології Грубером як специфікації певної концептуалізації залишає відкритим питання вибору формального апарату та мовних засобів для побудови такої специфікації. Не до кінця вирішеними залишаються завдання аналізу та виявлення суті онтологічного опису предметної області, визначення властивих йому обмежень та переваг. Це вимагає побудови та дослідження формальних моделей для різноманітних аспектів онтологічного моделювання.

Сьогодні надзвичайно актуальне питання створення інтелектуальних систем у певних галузях науки та промисловості, котрі могли б значною мірою спростити інженерам та науковцям пошук та доступ до потрібної (релевантної) інформації у сховищах величезних об’ємів. Систему, що вирізняється вмінням визначати, зберігати та використовувати в потрібний момент необхідні релевантні знання, називатимемо інтелектуальною системою (ІС). В основі архітектури сучасних ІС лежать онтології, що формуються для заданої предметної області. У процесі реалізації ІС найскладнішим є етап початкового формування такої онтології та подальшого її наповнення новими знаннями.

В сучасних умовах підвищення складності інформації та процесу обробки все частіше виникає ситуація, при якій не тільки людина, але навіть алгоритмічний метод обробки стають неефективними. Ці ситуації вимагають застосування

гнучких методів обробки. Одним з таких методів є метод агентів і мультиагентних систем [Безгубова 2015, с. 41-49].

Хорошим прикладом використання інтелектуальних агентів служать системи управління роботами. Роботи можуть мати широкий асортимент штучних органів почуттів (сенсорні датчики) і штучних ефекторів (маніпулятори, педіпулятори). Йдеться про робототехнічних пристроях, що виконують завдання, пов'язані з переміщеннями в просторі. Активність і автономність роботів тісно пов'язані з наявністю коштів мети й планування дій, систем підтримки вирішення завдань, а інтелектуалізація, крім володіння системою обробки знань, передбачає розвинені засоби комунікації різних рівнів, аж до засобів природньомовного спілкування.

Невід'ємним атрибутом інтелектуальних роботів є наявність спеціальної підсистеми планування, складовою програму дій робота в реальних умовах навколишнього середовища, які визначаються сенсорами робота. Для планування діяльності робот повинен мати знання про властивості навколишнього середовища і шляхи досягнення цілей в цьому середовищі.

Далі в дипломному проекті буде описана побудова мультиагентної системи керування групою інтелектуальних роботів з використанням вбудованих онтологій застосованих в якості баз знань для сувмісної автономної, адаптивної, продуктивної роботи.

1 ВИКОРИСТАННЯ ОНТОЛОГІЙ У ПОБУДОВІ КОМП’ЮТЕРНИХ ІНФОРМАЦІЙНИХ СИСТЕМ, ЩО НАКОПИЧУЮТЬ ТА ОБМІНЮЮТЬСЯ ЗНАННЯМИ

«Ручні ІС» («без комп'ютера») існувати не можуть, оскільки існуючі визначення наказують обов'язкову наявність у складі ІС апаратно-програмних засобів. Внаслідок цього поняття «автоматизована інформаційна система», «комп'ютерна інформаційна система» і просто «інформаційна система» є синонімами. Тож далі буде використано поняття інформаційної системи.

1.1 Визначення інформаційної системи.

1.1.1 Поняття інформаційної системи

Система – це сукупність елементів, що взаємодіють один з одним та утворюють певну цілісність, єдність.[Перегудов Ф.И., 1989]

Архітектура системи – сукупність властивостей системи, істотних для користувача. [Грицунов 2010,с. 222]

Елемент системи – частина системи, що має певне функціональне призначення. Елементи, що складаються з простих взаємопов'язаних елементів, часто називають підсистемами.

Організація системи – внутрішня впорядкованість, узгодженість взаємодії елементів системи, що виявляється, зокрема, в обмеженні різноманітності стану елементів в рамках системи.

Структура системи – склад, порядок і принципи взаємодії елементів системи, що визначають основні властивості системи. Якщо окремі елементи системи рознесені по різних рівнях і характеризуються внутрішніми зв'язками, то йдеться про ієрархічну структуру системи.

Додавання до поняття *система* слова *інформаційна* відображає мету її створення і функціонування. Інформаційні системи забезпечують збір, зберігання, обробку, пошук, видачу інформації, необхідної в процесі прийняття

рішень завдань з будь-якої області. Вони допомагають аналізувати проблеми і створювати нові інформаційні продукти. [Методи та засоби мультимедійних інформаційних систем 2015, с. 413-416.]

Інформаційна система – це взаємопов'язана сукупність засобів, методів і персоналу, що використовуються для зберігання, обробки та видачі інформації в інтересах досягнення поставленої мети.[Избачков, 2008, с.213]

Найбільш важливими принципами побудови ефективних інформаційних систем є наступні [Основи інформаційних систем, 2001, с. 187]:

- *Принцип інтеграції*, що полягає в тому, що оброблювані дані, одного разу введені в систему, багаторазово використовуються для вирішення великої кількості завдань.
- *Принцип системності*, що полягає в обробці даних в різних аспектах, щоб отримати інформацію, необхідну для прийняття рішень на всіх рівнях управління.
- *Принцип комплексності*, що полягає в механізації і автоматизації процедур перетворення даних на всіх етапах функціонування інформаційної системи.

1.1.2 Процеси, що протікають в інформаційних системах

Інформаційний процес – процес створення, збору, обробки, накопичення, зберігання, пошуку, розповсюдження та споживання інформації.

Інформаційний ресурс – це окремі документи і окремі масиви документів, документи і масиви документів в інформаційних системах (бібліотеках, архівах, фондах, банках даних, інших видах інформаційних систем.[Проектування інформаційних систем 2002, с. 286]

У нормативно-правовому аспекті документ визначається як зафіксована на матеріальному носії інформація з реквізитами, що дозволяють її ідентифікувати.

Процес *документування* перетворює інформацію в інформаційні ресурси.

Процеси, що забезпечують роботу інформаційної системи будь-якого призначення, умовно можна представити як такі, що складаються з наступних блоків:

- введення інформації із зовнішніх чи внутрішніх джерел;
- обробка вхідної інформації і представлення її в зручному вигляді;
- висновок інформації для представлення споживачам або передачі в іншу систему;
- зворотній зв'язок – це інформація, перероблена людьми даної організації для корекції вхідної інформації.

Інформаційні процеси реалізуються за допомогою інформаційних процедур, що реалізують той чи інший механізм переробки вхідної інформації в конкретний результат. [Маслов 2005, с. 98]

Розрізняють такі типи інформаційних процедур:

1. Повністю *формалізовані*, при виконанні яких алгоритм переробки інформації залишається незмінним і повністю визначений (пошук, облік, зберігання, передача інформації, друк документів, розрахунок на моделях).
2. *Неформалізовані* інформаційні процедури, при виконанні яких створюється нова унікальна інформація, причому алгоритм переробки вихідної інформації невідомий (формування множини альтернатив вибору, вибір одного варіанта з отриманої множини).
3. *Погано формалізовані* інформаційні процедури, при виконанні яких алгоритм переробки інформації може змінюватися і повністю не визначений (завдання планування, оцінка ефективності варіантів економічної політики).

Функції інформаційних підрозділів, що створюють і підтримують інформаційні системи (служба адміністратора): оповіщення і обробка запитів; підтримання цілісності та збереження інформації; періодична ревізія інформації; автоматизація індексування інформації.

В цілому інформаційні системи визначається наступними властивостями:

- будь-яка інформаційна система може бути піддана аналізу, побудована і керована на основі загальних принципів побудови систем;
- інформаційна система є динамічною і розвивається;
- при побудові інформаційної системи необхідно використовувати системний підхід;
- вихідною продукцією інформаційної системи є інформація, на основі якої приймаються рішення;
- інформаційну систему варто сприймати як людино-машинну систему обробки інформації.

Впровадження інформаційних систем може сприяти [Избачков, 2008, с.213]:

- отриманню більш раціональних варіантів вирішення управлінських завдань за рахунок впровадження математичних методів;
- звільнення працівників від рутинної роботи за рахунок її автоматизації;
- забезпечення достовірності інформації;
- вдосконалення структури інформаційних потоків (включаючи систему документообігу);
- надання споживачам унікальних послуг;
- зменшення витрат на виробництво продуктів і послуг (включаючи інформаційні).

1.2 Класифікація інформаційних систем

1.2.1 Типи інформаційних систем

Тип інформаційної системи залежить від того, чиї інтереси вона обслуговує і на якому рівні управління. За характером уявлення і логічної організації інформації, що зберігається, інформаційні системи підрозділяються на фактографічні, документальні та геоінформаційні.

Фактографічні інформаційні системи накопичують і зберігають дані у вигляді множини екземплярів одного або декількох типів структурних елементів (інформаційних об'єктів). Кожен з таких екземплярів або деяка їх

сукупність відображають відомості з якого-небудь факту, події окремо від усіх інших відомостей і фактів.

Структура кожного типу інформаційного об'єкта складається з кінцевого набору реквізитів, що відображають основні аспекти та характеристики об'єктів даної предметної області. Комплектування інформаційної бази в фактографічних інформаційних системах включає, як правило, обов'язковий процес структуризації вхідної інформації.

Фактографічні інформаційні системи припускають задоволення інформаційних потреб безпосередньо, тобто шляхом подання споживачам самих відомостей (даних, фактів, концепцій).

У *документальних (документованих) інформаційних системах* одиничним елементом інформації є нерозчленований на більш дрібні елементи документ і інформація при введенні (вхідний документ), як правило, не структурується, або структурується в обмеженому вигляді. Для документа, що вводиться, можуть встановлюватися деякі формалізовані позиції (дата виготовлення, виконавець, тематика).

Деякі види документальних інформаційних систем забезпечують встановлення логічного взаємозв'язку документів, що вводяться - підпорядкованість за смисловим змістом, взаємні посилання з яких-небудь критеріями і т.д. Визначення і встановлення такого взаємозв'язку є складною многокритериальную і багатоаспектну аналітичну задачу, яка не може бути формалізована в повній мірі.

У *геоінформаційних системах* дані організовані у вигляді окремих інформаційних об'єктів (з певним набором реквізитів), прив'язаних до загальної електронної топографічній основі (електронній карті). Геоінформаційні системи застосовуються для інформаційного забезпечення в тих предметних областях, структура інформаційних об'єктів і процесів в яких має просторово-географічний компонент (маршрути транспорту, комунальне господарство).[Гайдамакин 2002, с. 128]

1.2.2 Класифікація інформаційних систем за функціональною ознакою

Функціональна ознака визначає призначення підсистеми, а також її основні цілі, завдання та функції. На рис. 1.1 подано класифікацію інформаційних систем за влучним висловом їх функціональних підсистем.

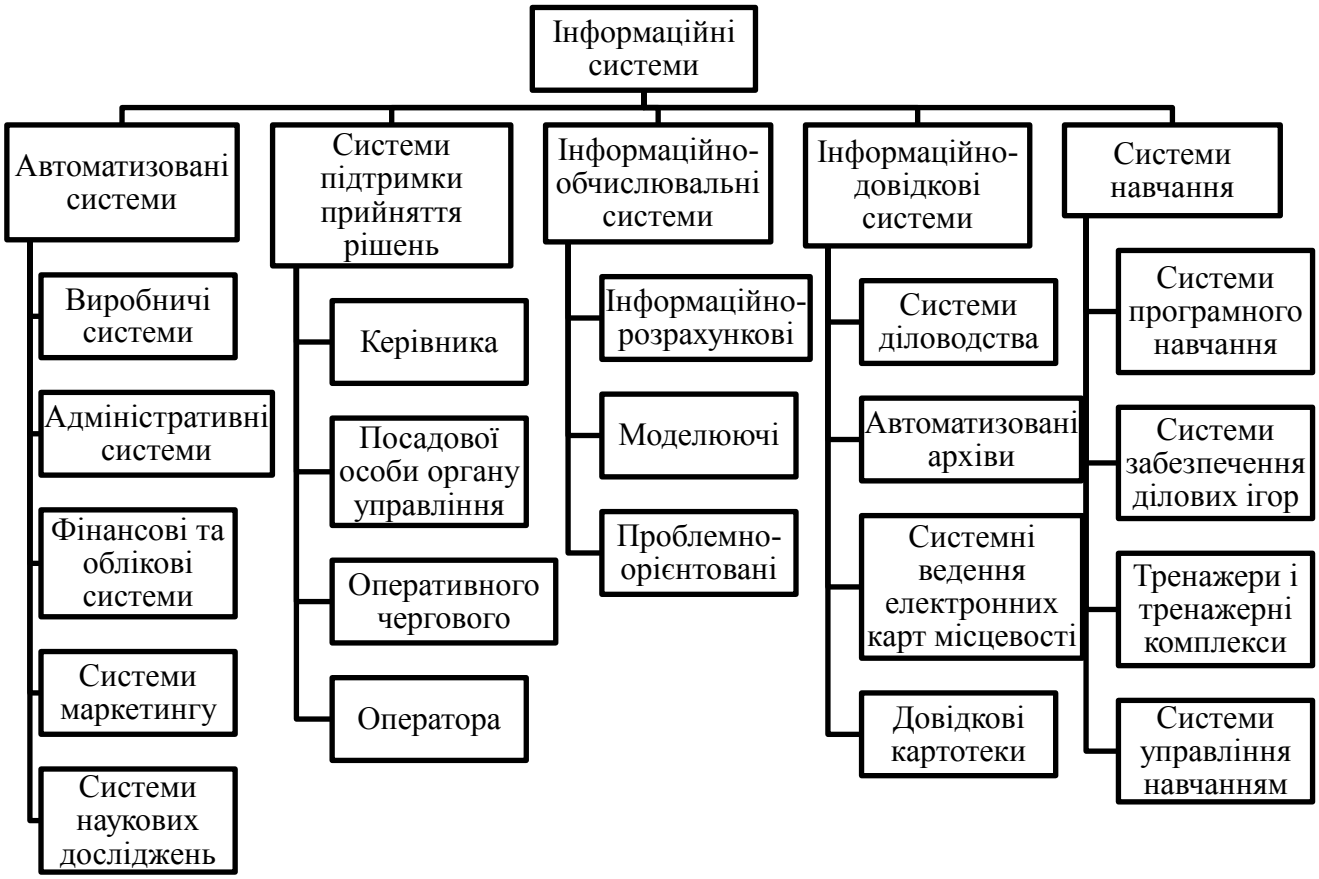

Рисунок 1.1 – Класифікація інформаційних систем за функціональною ознакою[Информационные системы в экономике 2008, с. 231]

1.2.3 Класифікація інформаційних систем за рівнями управління

Виділяють[Гайдамакин 2002, с. 128]:

- інформаційні системи оперативного (операційного) рівня - бухгалтерська, банківських депозитів, обробки замовлень, реєстрації квитків, виплати зарплати;

- інформаційна система фахівців - офісна автоматизація, обробка знань (включаючи експертні системи);
- інформаційні системи тактичного рівня (середня ланка) - моніторинг, адміністрування, контроль, прийняття рішень;
- стратегічні інформаційні системи - формулювання цілей, стратегічне планування.

1.3 Застосування мультиагентних систем в інформаційних системах

Субстанціональний підхід до аналізу інформаційних систем однозначно визначає інформаційні моделі як основу обробки інформації в інформаційних системах [Цветков 2005, с. 118-122.]. Однак процес обробки пов'язаний не тільки з об'єктами обробки, але і з процесами. В сучасних умовах підвищення складності інформації та процесу обробки все частіше виникає ситуація, при якій не тільки людина, але навіть алгоритмічний метод обробки стають неефективними. Ці ситуації вимагають застосування гнучких методів обробки. Одним з таких методів є метод агентів і мультиагентних систем [Безгубова 2015, с. 41-49.]. Агент може бути розглянутий як комп'ютерна система, яка знаходиться в деякому динамічному середовищі, і яка здатна на автономні дії в цьому середовищі.

1.3.1 Агенти як інструмент зниження складності

Проблема складності пов'язана як зі структурною складністю так і з об'ємною складністю. Вона зустрічається як при задачах обробки так і в завданнях управління. Мультиагентні системи можуть застосовуватися як для обробки так і для управління. Один з підходів до управління пов'язаний з побудовою мережних систем (Networking Organizations), підрозділи яких можуть розглядатися як автономні підприємства [Todd 2012, с. 229-245.]. Мережева організація є відкритою і входять до її складу підприємства можуть

взаємодіяти з іншими організаціями. Домінуючими процесами в відкритих організаціях є: навчання, розвиток та адаптація, - вимагають узгодженого, гнучкого і оперативного прийняття рішення. Це підвищує складність таких систем і створює інформаційний бар'єр.

Новий підхід до завдань оперативної обробки інформації в мережевих і складних системах зв'язуються із застосуванням мультиагентних технологій [Тарасов 1998, с. 5-63]. Вони розвиваються на базі: методів штучного інтелекту, об'єктно-орієнтованого програмування, паралельних обчислень і телекомунікацій. В основі цих технологій лежить поняття «агента», програмного об'єкта, здатного сприймати ситуацію, приймати рішення і комунікувати з собі подібними. Ці можливості відрізняють адаптивні і перебудовуються мультиагентні системи від «жорстко» організованих систем. Мультиагентні системи здатні до саморозвитку і самоорганізації. Агенти можуть діяти від імені та за дорученням осіб, які приймають рішення, та на основі даних їм повноважень в автоматичному режимі вести переговори, знаходити варіанти рішень та узгоджувати свої рішення один з одним. Тут слід зазначити тенденцію субсидіарного управління [Цветков 2012, с. 40-43.], яка реалізується в агентних системах. Цікавим є дослідження агентів як нової форми вирішення управлінських і наукових завдань.

1.3.2 Мультиагентний підхід

В основі мультиагентного підходу лежить поняття мобільного програмного агента, який реалізований і функціонує як самостійна спеціалізована комп'ютерна програма або елемент штучного інтелекту.

На зміну таким системам, що копіюють централізовану ієрархію, швидко прийшли розподілені системи, в яких знання і ресурси розподілялися між досить "самостійними" агентами, але зберігався загальний орган командного управління, який приймає рішення в критичних або конфліктних ситуаціях. Подальшим кроком в цьому напрямку стала парадигма повністю децентралізованих систем, в яких управління відбувається тільки за рахунок

локальних взаємодій між агентами. При цьому вузька функціональна орієнтація агента на рішення якоїсь однієї окремої частини загального завдання поступово стала поступатися місцем універсальної цілісності (автономності). Прикладами таких децентралізованих організацій частково можуть служити колонії комах, наприклад, бджіл або мурах.[Городецкий 1998, с. 64–116]

Суть мультиагентних технологій полягає в принципово новому методі вирішення завдань. На відміну від класичного способу, коли проводиться пошук деякого чітко визначеного (детермінованого) алгоритму, що дозволяє знайти найкраще вирішення проблеми, в мультиагентних технологіях рішення виходить автоматично в результаті взаємодії безлічі самостоя тільних цілеспрямованих програмних модулів - так званий мих програмних агентів.

Найчастіше класичні методи вирішення завдань або незастосовні до реального життя (неважко уявити собі, що значить спробувати вирішити задачу управління підприємством в непербачуваній динамічній обстановці сучасного бізнесу, навіть з по міццю вищої математики і найбільш просунутих економічних моделей), або вони вимагають величезних обсягів розрахунків, або вони зовсім відсутні.

Чи означає це, що ситуація, коли точний алгоритм рішення відсутня, безнадійна? Ні - відповідають мультиагентні технології. Зрештою, людям у повсякденному житті постійно доводиться в умовах дефіциту часу і коштів вирішувати завдання, які не мають точного формального рішення - і вони вирішуються часто не найгіршим чином.[Shoham 2009, с. 14–37.]

На рис. 13.1 показані в порівнянні дві схеми побудови про граммного забезпечення: традиційна і на базі мультиагентної системи. У МАС кожної сутності ставиться у відповідність про програмний агент, який представляє її інтереси.

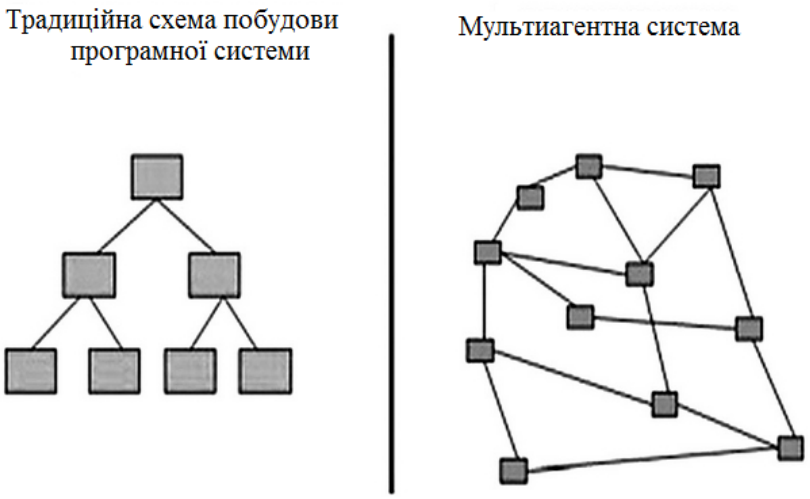

Рисунок 1.2 – Традиційне і мультиагентних побудова програмного продукту

Агенти дуже схожі на членів команди, які можуть змагатися один з одним або співпрацювати в процесі прийняття рішення. Ключова особливість Емерджентні інтелекту - динаміка і непередбачуваність процесу прийняття рішень. На практиці це означає, що рішення досягається за рахунок сотень і тисяч взаємодій, які майже неможливо відстежити. Але це і не потрібно, оскільки агентам дають цілі, які вони повинні досягати, але не визначають сценарії виконання завдань по досягнення цих цілей.

Ці сценарії формуються і виконуються агентами самостійно. На кожному кроці агенти розглядають входи системи і реагують на непередбачувані події (затримки, збої, зміни). Реакція може бути самостійною, або здійснюватися у взаємодії з оператором. Таким чином, емерджентним інтелект - це не є якийсь один новий і спеціально сконструйований унікальний блок-вирішувач, доданий до системи. Навпаки, це щось (результат самоорганізації), що виникає як би "з повітря" (за рахунок безлічі прихованих або яв них умов, що склалися в ситуації), спонтанно і в заздалегідь не передбачений момент часу, і так само несподівано зникає, але в процесі свого існування визначальним чином керує роботою всієї системи. Тут ми маємо справу з виникненням поряд ка з хаосу, з одним з тих явищ, які вивчали і описували такі видатні вчені, як Олександр Богданов (теорія організації), Ілля Пригожий (самоорганізація в фізичних

системах), Марвін Мінський (психологія і теорія мислення), Артур Кестлер (біологія).

1.3.3 Агенти і мультиагентні системи

На початку XXI століття група провідних світових вчених склала список пріоритетних завдань кібернетики на найближчі 50 років Серед них:

- динамічно реконфігурованих інтелектуальне управління складними системами;
- асинхронна теорія управління;
- управління розподіленими об'єктами через Інтернет;
- перепрограмування системи управління бактеріями;
- створення футбольної команди роботів, яка виграє у переможця кубка світу серед людей.

Мультиагентна система (МАС) кардинально відрізняються від традиційних "жорстко" організованих систем, і, в перспективі, здатні допомогти у вирішенні цих завдань.

Відкритий характер сучасного інформаційного суспільства та глобальної ринкової економіки призводить до прискорення науково-технічного прогресу і загострення конкуренції на ринках. Це змушує підприємства шукати нові методи і засоби організації і управління, спрямовані на більш якісне і ефективне задоволення індивідуальних запитів споживачів. Більшість сучасних систем характеризуються відсутністю коштів своєчасної ідентифікації нових потреб і можливостей в середовищі, що дозволяють підприємству оперативно приймати ефективні рішення по реконфігурації виробничих, кадрових, фінансових та інших ресурсів.

Необхідність модифікації схеми прийняття рішень в традиційних системах виявляється складною і трудомісткою завданням, яке вимагає високої кваліфікації виконавців. Це робить розробку і експлуатацію таких систем вкрай дорогими. Відповідно, ще однією актуальною проблемою сьогодення стає зростання обсягів інформації і ступеня складності опису систем.

Для вирішення подібних проблем застосовуються мультиагентні технології, в основі яких лежить поняття "агента", яке останнім часом було адаптовано до багатьох областям як прикладного і системного програмування, так і до досліджень в областях штучного інтелекту і розподілених інтелектуальних систем. Причому в кожному конкретному випадку поняттю надається дещо різне значення.

Спочатку ідея створення інтелектуального посередника (агента) "виникла в зв'язку з бажанням спростити стиль спілкування кінцевого користувача з комп'ютерними програмами, оскільки домінуючий, в основному, і нині стиль взаємодії користувача з комп'ютером передбачає, що користувач запускає завдання явно і управляє її рішенням. Але це абсолютно не підходить для недосвідченого користувача ". Інакше кажучи, спочатку ідея інтелектуального посередника виникла як спроба інтелектуалізації призначеного для користувача інтерфейсу.

Ще однією важливою властивістю агента є те, що він поміщений в зовнішнє середовище, з якої він здатний взаємодіяти. Зазвичай, середовище не контролюється агентом, він лише здатний впливати на неї. Поділ намірів і бажань необхідно, так як агент може мати несумісні бажання або бажання можуть бути недосяжні. Оскільки агент обмежений в ресурсах і не може досягти всіх бажань одночасно, природно вибирати найбільш значущі цілі - наміри. Отже, агент - розумна сутність, вміщена в зовнішнє середовище, здатна взаємодіяти з нею, роблячи автономні раціональні дії для досягнення цілей, тобто, Інтелектуальний агент - це агент, що володіє наступними властивостями:

- реактивність (англ. reactivity) - агент відчуває зовнішнє середовище і реагує на зміни в ній, здійснюючи дії, спрямовані на досягнення цілей;
- проактивность (англ. pro-activeness) - агент показує кероване цілями поведінку, проявляючи ініціативу, здійснюючи дії спрямовані на досягнення цілей;

- соціальність (англ. social ability) - агент взаємодіє з іншими сутностями зовнішнього середовища (іншими агентами, людьми і т. д.) для досягнення цілей.

При розробці системи кожне з перших двох властивостей досягається досить легко. Найбільшу складність представляє поєднання в системі обох властивостей в потрібних пропорціях. Буде не дуже ефективно, якщо агент жорстко слід сценарієм досягнення мети, не реагуючи на зміни у зовнішньому середовищі і не володіючи здатністю помітити необхідності коригування плану. Але також неефективною буде і поведінка, обмежена лише реакцією на що надходять із зовні стимули, без будь-якого планування цілеспрямованих дій.

Складність формулювання змістовних практично значущих завдань і неможливість апріорного точного завдання всіх умов функціонування висувають адаптивні постановки проблем, окремо виділяючи таку особливість агентів, як адаптивність - здатність автоматично пристосовуватися до невизначених і мінливих умов в динамічному середовищі.

Попередниками програмних агентів можна вважати складні адаптивні системи, які вміють підлаштовуватися під ситуацію або обставини і принциповим чином змінювати свою поведінку або характеристики, щоб забезпечити вирішення поставлених перед ними завдань. Однак у випадках, коли агент функціонує в складній, постійно мінливому середовищі, взаємодіючи при цьому з іншими агентами, така мультиагентна система значно складніше просто адаптивної системи, так як вона швидше навчається і може діяти ефективніше за рахунок перерозподілу функцій або завдань між агентами.

Складні системи часто розглядають як середовище дії агентів. З поняттям складних систем пов'язані такі фундаментальні ідеї, які безпосередньо впливають на функціонування МАС:

- в складних системах існують автономні об'єкти, які взаємодіють один з одним при виконанні своїх визначених завдань;

- агенти повинні мати можливість реагувати на мінливі умови середовища, в якій вони функціонують і, можливо, змінювати свою поведінку на основі отриманої інформації;
- складні системи характеризуються виникають структурами - логічно пов'язаними схемами, які формується в результаті взаємодії між агентами;
- складні системи з виникаючими структурами часто існують на межі порядку і хаосу;
- при створенні складних систем на базі агентів має сенс розглядати біологічні аналогії, такі як: паразитизм, симбіоз, репродукцію, генетику, мітоз і природний відбір (наприклад, компанія British Telecom при формуванні мережі напрямки дзвінків використовує модель діяльності колонії мурах).

Концепція агентів, розроблена в рамках мультиагентних технологій і мультиагентних систем, передбачає наявність активного поведінки агентів, тобто здатності комп'ютерної програми самостійно реагувати на зовнішні події і вибирати відповідні дії. Сьогодні агентні технології пропонують різні типи агентів, моделі їх поведінки і властивості, сімейство архітектур і бібліотеки компонентів, орієнтовані на сучасні вимоги.

Програмні інтелектуальні агенти - це новий клас систем програмного забезпечення, яке діє або від імені користувача, або від імені системи делегувала агенту повноваження на виконання тих чи інших дій. Вони є, по суті, новим рівнем абстракції, відмінним від звичних абстракцій типу - класів, методів і функцій. Але при цьому, розробка МАС дозволяє створювати системи володіють розширюваністю / масштабованість, мобільністю / переносимістю, інтероперабельність, що безсумнівно дуже важливо при розробці систем, заснованих на знаннях.

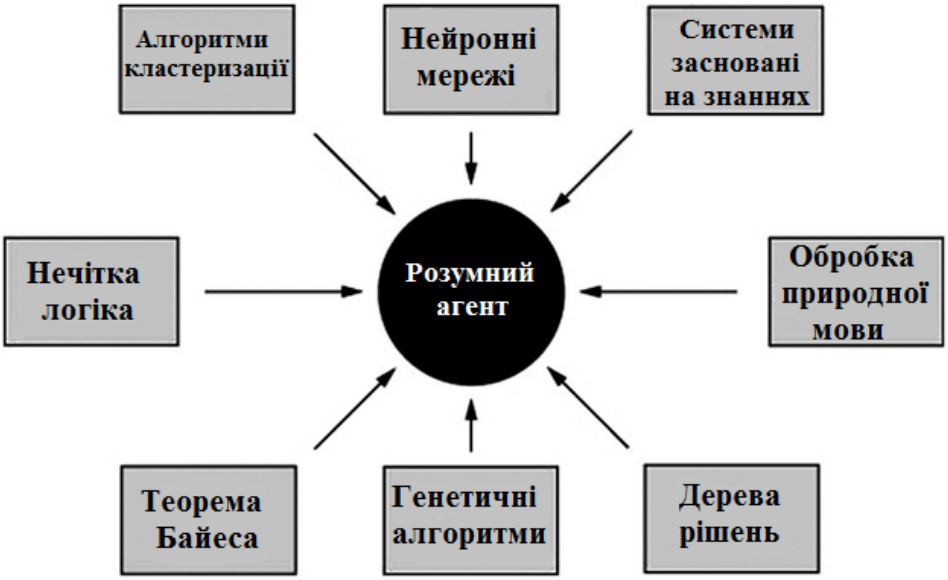

Рисунок 1.3 – Області знання і технології, використовувані інтелектуальними агентами

Проста комп'ютерна програма відрізняється від агента тим, що "не обтяжує" себе цільовим поведінкою і аналізом досягнутих результатів. Навпаки, агент, що представляє інтереси користувача, "зацікавлений" у тому, щоб завдання було виконано. У разі невдачі або якогось збою він повинен повторити спробу пізніше або мати про запас альтернативний варіант вирішення проблеми. Агенти в процесі відпрацювання завдань завжди формує список виконаних дій, результати тестування і верифікації і відсилають його в керуючу систему.

На підставі викладеного вище ми можемо скомпілювати наступне визначення: "агент - це самостійна програмна система,

1. має можливість приймати вплив із зовнішнього світу;
2. визначальна свою реакцію на цей вплив і формує відповідь дія;
3. змінює свою поведінку з плином часу в залежності від накопиченої інформації і витягнутих з неї знань,
4. володіє мотивацією і здатна після делегування повноважень користувачем поставити себе на його місце і прийняти рішення, відповідне ситуації ".
5. Інтелектуальний агент повинен мати наступні властивості:

6. автономність - здатність функціонувати без втручання з боку свого власника і здійснювати контроль внутрішнього стану і своїх дій;
7. адаптивність - агент має здатність навчатися;
8. коллаборативного - агент може взаємодіяти з іншими агентами декількома способами, граючи різні ролі;
9. здатність до міркувань - агенти можуть володіти частковими знаннями або механізмами виведення, а також спеціалізуватися на конкретну предметну область;
10. комунікативність - агенти можуть спілкуватися з іншими агентами;
11. мобільність - здатність передачі коду агента з одного сервера на інший;
12. соціальну поведінку - можливість взаємодії і комунікації з іншими агентами;
13. реактивність - адекватне сприйняття середовища і відповідні реакції на її зміни;
14. активність - здатність генерувати цілі і діяти раціонально для їх досягнення;
15. наявність базових знань - знання агента про себе, навколишньому середовищу, включаючи інших агентів, які не змінюються в рамках життєвого циклу агента;
16. наявність переконань - змінна частина базових знань, які можуть змінюватися в часі;
17. наявність мети - сукупність станів, на досягнення яких спрямоване поточна поведінка агента;
18. наявність бажань - стану і / або ситуації, досягнення яких для агента важливо;
19. наявність зобов'язань - завдання, які бере на себе агент на прохання і / або дорученням інших агентів;
20. наявність намірів - те, що агент повинен робити в силу своїх зобов'язань та / або бажань

Мультиагентна система (МАС) - складна система, в якій функціонують два або більше інтелектуальних агентів. Процес самоорганізації в мультиагентних системах - внутрішня впорядкованість, узгодженість, взаємодія більш або менш диференційованих і автономних агентів агентної системи, зумовленої її будовою. Таким чином, в МАС кілька агентів можуть спілкуватися, передавати один одному деяку інформацію, взаємодіяти між собою і вирішувати поставлену. У такій системі завдання (або підзадачі) розподілені між агентами, кожен з яких розглядається як член групи або організації. Розподіл завдань передбачає призначення ролей кожному з членів групи, визначення міри його "відповідальності" і вимог до "досвіду".

Основою формою організації взаємодії між агентами, що характеризується об'єднанням їхніх зусиль для досягнення спільної мети при одночасному поділі між ними функцій, ролей і обов'язків є кооперація.

У загальному випадку це поняття можна визначити формулою: {кооперація = співпрацю + координація дій + вирішення конфліктів}. Під координацією зазвичай розуміється управління залежностями між діями. Комунікація між штучними агентами залежить від обраного протоколу, який представляє собою безліч правил, що визначають, як синтезувати значущі і правильні повідомлення. Фундаментальними особливостями групи, складеної з агентів, які співпрацюють для досягнення спільної мети, є соціальна структура і розподіл ролей між агентами.

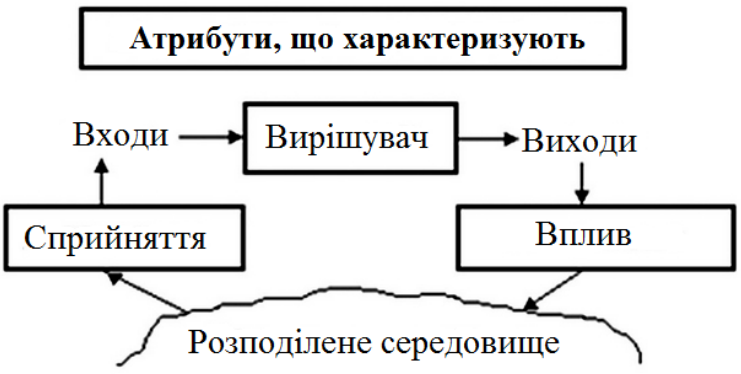

Рисунок 1.4 – Укрупнена структура агента

Основою архітектури агента є контекст, або серверне середовище, в якому він виконується. Кожен агент має постійний ідентифікатор - ім'я. У серверному середовищі може виконуватися не тільки вихідний агент, а й його копія. Агенти здатні самостійно створювати свої копії, розсилаючи їх по різних серверів для виконання роботи. Після прибуття агента на наступний сервер його код і дані переносяться в новий контекст і стираються на попередньому місцезнаходження. У новому контексті агент може робити все, що там не заборонено. Після закінчення роботи в контексті агент може переслати себе в інший контекст або по вихідний адресою відправника. Агенти здатні також вимикатися ("вмирати") самі або за командою сервера, який переносить їх після цього з контексту в місце, призначене для зберігання.

На рисунку 1.4 показана укрупнена структура типового агента. Входами є внутрішні параметри агента і дані про стан середовища. Виходи - параметри, що впливають на середу і інформують користувача (або програму, що виконує роль менеджера в системі) про стан середовища та прийняті рішення. Вирішувач - процедура прийняття рішень. Вирішувач може бути досить простим алгоритмом або елементом системи штучного інтелекту.

Загальна методологія висхідного еволюційного проектування МАС може бути представлена ланцюжком: {довкілля - функції МАС - ролі агентів - відносини між агентами - базові структури МАС - модифікації}, і включає наступні етапи:

- формулювання призначення (цілі розробки) МАС;
- визначення основних і допоміжних функцій агентів в МАС;
- уточнення складу агентів і розподіл функцій між агентами, вибір архітектури агентів;
- виділення базових взаємозв'язків (відносин) між агентами в МАС;
- визначення можливих дій (операцій) агентів;
- аналіз реальних поточних або передбачуваних змін зовнішнього середовища.

1.3.4 Підхід "Агентів і Світів" в розробці МАС

Одним із сучасних підходів до створення МАС є підхід, заснований на концепції "Агентів і Світів" (далі - агентів і світів).

На відміну від інших відомих підходів, цей підхід реалізує формування загального миру діяльності сторін, що кооперуються і світів діяльності кожної з них, шляхом створення єдиної комплексної середовища. Це середовище є основою діяльності менеджерів і фахівців і забезпечує відтворення основних компонент процесу діяльності підприємства і взаєморозуміння між людьми. Використання інтелектуальних агентів в комбінації з світами дій і міркувань дозволяє менеджерам моделювати три "кити" кооперації: колективне мислення, обґрунтоване поведінку і комунікацію сторін-учасниць. За рахунок цього кожна зі сторін покладає на свого агента місію узгодження більшості проблем, що виникають. Це найчастіше реалізується у вигляді реалізації віртуального круглого столу, що пропонує агентам загальний мир дії.

У розглянутому підході світ дій - це модель середовища діяльності, що базується на знаннях. Головна її відмінність від традиційних систем моделювання полягає в тому, що ця система формує модель віртуального простору і надає прямий доступ до об'єктів світу в цьому просторі для виконання дій, моделюючи реакцію на ці дії відповідно до законів світу.

Принципи побудови і функціонування світів коротко можуть бути описані в такий спосіб:

- світ складається з об'єктів, здатних взаємодіяти один з одним відповідно до законів світу;
- для користувача світи представляються сценами, що складаються з деяких сцен і заданих об'єктів, з певними відносинами між ними, і об'єктами, потенційно застосовними в сцені;
- об'єкти світу визначаються властивостями, що забезпечують їх здатність вступати у взаємодію з іншими об'єктами; стану об'єктів визначаються їх

властивостями і відносинами; потенційно можливі властивості об'єктів визначаються законами світу, що діють в сцені;

- закони світу задаються сценаріями дій, які визначаються як правила змін стану об'єктів світу; прості сценарії дозволяють складати більш складні;
- відносини між об'єктами визначають зв'язку між ними; найбільш поширеними відносинами є "ціле-частини", "приналежність", "міра" і ряд інших;
- основні концепти виражаються в атрибутах речовини, простору і часу, енергії та інформації.

Ці базові категорії дозволяють конструювати світи дій в різних предметних областях. Розглянуті принципи дозволяють також створювати світи міркувань, такі як економіка і політика, технології і торгівля і т. п.

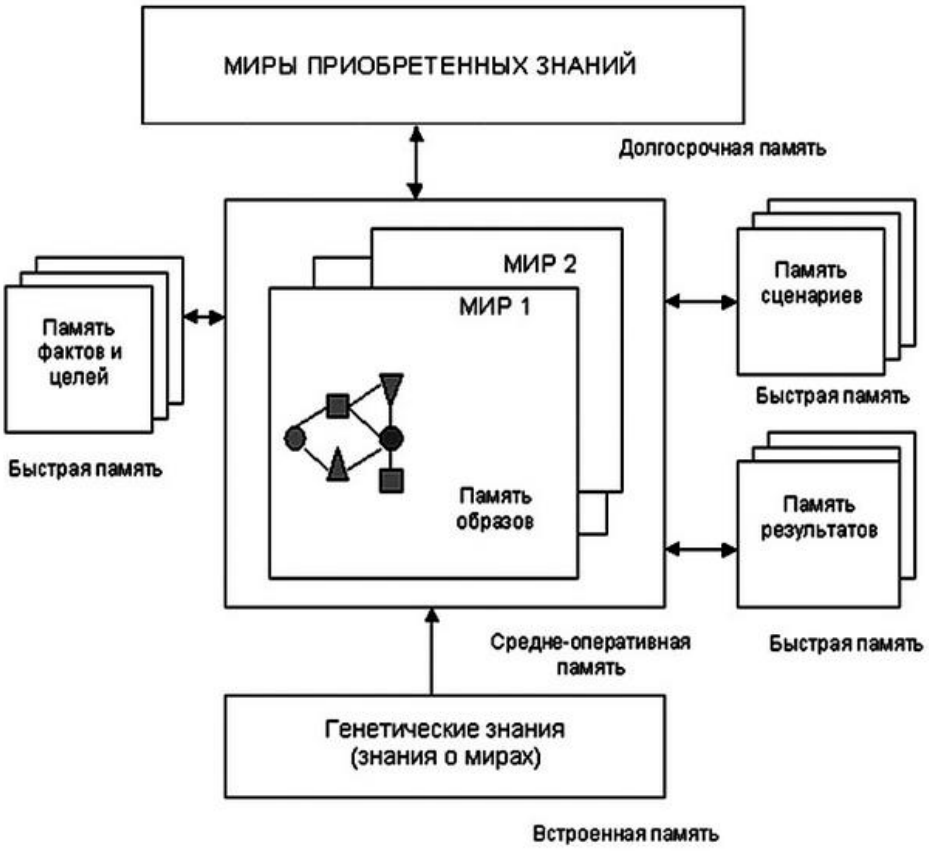

Рисунок 1.5 – Модель пристрою пам'яті інтелектуального агентар[Управление на базе мультиагентных систем, 2010, с. 7]

Розглянемо відповідні моделі пристрою пам'яті і мислення агента. У структурі пам'яті агента виділені наступні компоненти (рис. 1.5):

- довгострокова пам'ять агента, що містить повні семантичні мережі предметних областей знань; ця пам'ять поповнюється знаннями в процесі навчання агента і постійно трансформується і систематизується, через неї, як через сито, пропускаються всі вхідні факти;
- пам'ять (простір) свідомості, що містить образи об'єктів світів, що є середньостроковій. У цій пам'яті містяться описи сцени в кожному з світів (також у формі семантичної мережі) і тут же виконуються основні розумові операції над образами об'єктів;
- пам'ять фактів, а також пам'ять сценаріїв - найбільш часто змінювані структури змісту пам'яті (оперативна пам'ять). У пам'яті фактів знаходяться вихідні і кінцеві, а також всі проміжні факти, одержувані в процесі міркувань і розрахунків; пам'ять сценаріїв також піддається перетворенням, в першу чергу, пов'язаних з узагальненням і конкретизацією сценаріїв;
- пам'ять генетичних знань - це жорстко вбудовані в систему і незмінні знання, тут - знання про конструкції і функціонування світів.

Загальні принципи мислення агента є цілком традиційними і включають наступні три основні фази (рис. 1.6):

- сприйняття - отримання даних і побудова моделі сцени в завантаженому світі;
- пізнання - аналіз і формується сценарій дій суб'єкта для досягнення поставлених цілей;
- виконання - наміченого сценарію з постійним порівнянням очікуваних і спостережуваних результатів.

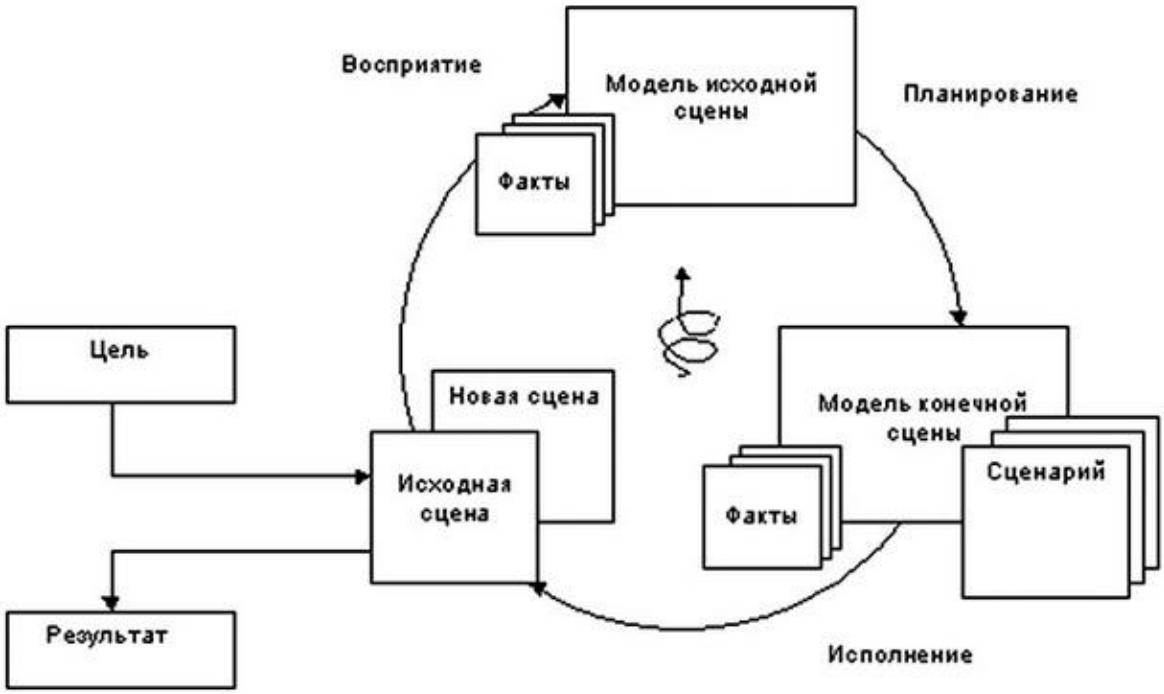

Рисунок 1.6 – Схема мислення інтелектуального агента [Управление на базе мультиагентных систем, 2010, с. 9]

На відміну від інших подібних систем, в даній системі реалізація цих фаз здійснюється через два базових механізми: абстрагування і конкретизації, тісно пов'язаних між собою. У цьому сенсі мислення агента нагадує рух поршнів в двигуні: рух вгору - шляхом абстрагування, вниз - шляхом конкретизації.

При "пошарових" міркуваннях основний час витрачається на виконання "розумових" операцій - дій з образами об'єктів (поняттями), які змінюють стан сцени і, тим самим, обмежують застосування дедуктивних міркувань. Використовувана при цьому логіка дій також істотно відрізняє пропоновану модель від традиційних дедуктивних систем можливостями вибору.

1.4 Онтології в мультиагентних системах

Онтології - засновані на дескрипційних логіках системи подання знань, що набули широкого поширення у вирішенні задач каталогізації, класифікації та подання у дружньому для людини вигляді інформації: існує безліч проектів, що використовують онтології для представлення знань. Крім вищевказаних

застосувань, в яких бази знань в першу чергу орієнтовані на використання їх людиною, на сьогоднішній день все частіше онтології використовуються для зберігання уявлень про світ агентів в мультиагентних системах. Це дозволяє застосовувати алгоритми виводу в дескрипційних логіках безпосередньо на базі знань агента, а також забезпечує більшу стандартизацію і впорядкованість знань, що дозволяє спростити комунікацію між агентами.[Науменко 2017, с. 50-52]

1.4.1 Дескрипційна логіка

У загальному розумінні, дескрипційна логіка (далі DL) - мова представлення знань, що дозволяє описувати поняття предметної області у недвозначному, формалізованому вигляді. Вони поєднують в собі, з одного боку, багаті виражальні можливості, а з іншого – хороші обчислювальні властивості, такі як розв'язність і відносно невисока обчислювальна складність основних логічних проблем, що робить можливим їх застосування на практиці. Таким чином, DL являють собою компроміс між виражальністю і розв'язністю.[Бочаров 2001, с. 196]

Описову логіку можна розглядати як розв'язні фрагменти логіки предикатів, синтаксично ж вони близькі до модальних логік. Нехай C - множина концептів (аналог унарних предикатів), R - множина ролей (аналог бінарних предикатів), I - множина імен індивідів (тобто об'єктів предметної області) DL. Тоді термінологією (або скорочено TBox) називається набір тверджень виду $C \equiv D$(еквівалентність) або $C \subseteq D$ (включення), де C і D - довільні концепти; системою фактів (ABox) називається набір тверджень виду a: C і aRb, де $a, b \in I$ є індивіди, C - довільний концепт, R - роль.

Описи концептів інтерпретуються у класичному сенсі теорії моделей: інтерпретація I - це пара (Y,f), де Y - непорожня множина індивідів, а f - функція інтерпретації, яка відображає множину концептів C в Y і множину імен ролей в $Y \times Y$.

Інтерпретація I називається моделлю для термінології T, якщо для будь-якого включення $A \subseteq D \in T$ вірно $f(A) \subseteq f(D)$ і для будь-якої еквівалентності $A \equiv B$ вірно $f(A)=f(B)$. Концепт C називається здійсненним в термінології T, якщо існує така інтерпретація (Y,f), що є моделлю T, що f(C) непорожня.[Найданов 2014, с. 9046]

1.4.2 Аналіз областей використання онтологій при побудові інформаційних систем

Перший крок в концептуальному моделюванні діяльності інформаційних систем є перетворення сприйняття реального світу в моделі світу, що він має намір представити [Soares 2010, с. 127]

Онтологія - на основі опису логічної системи подання знань, широко використовується при вирішенні задач каталогізації, класифікації та подання інформації, орієнтованої на людину. Ми зацікавлені в онтології в контексті онтологічного керування інформаційними системами, де вони можуть бути використані під час розробки і під час виконання [Soares 2010, с. 127].

Існує велика область використання онтології для представлення знань в інформаційних системах, наприклад:

- завдання аналізу текстів - BIOAP (Basic Integrated Ontological Analytic Processor) 1.0

Онтологія визначає терміни, що використовуються для опису і представлення знань про конкретну предметну область. Це необхідно для людей, для додатків, систем баз даних і різних інших інформаційних систем, які поділяють конкретну інформацію в будь-якій предметній області. Онтологія включає для визначення комп'ютерної обробки основних концепцій в області і відносин між ними [Захарова 2011].

- Оцінка інформаційної системи охорони здоров'я - FI-STAR (Future Internet Social and Technological Alignment Research)

Онтологічні уявлення інформаційної системи охорони здоров'я дає обчислювальну структуру, з якої кілька ознак, включаючи аспекти оцінки,

можуть бути вилучені. Функції можуть бути визначені на цій онтології, як кількісні, що об'єднують, порівнюють або вибирають деякі з вузлів або гілок. Сама онтологія може бути розширена шляхом присвоєння значень її вершин і ребер, даючи можливість подальших висновків [Eivazzadeh 2016].

- побудова семантичних інформаційних систем

Використання семантичних технологій дає незаперечну перевагу на всіх основних етапах аналізу, проектування, реалізації, тестування і супроводу інформаційних систем, в тому числі опису семантики предметної області з використанням онтологій як змінної частини інформаційної системи. Підхід до створення семантичних інформаційних систем на основі Сховища IT-інфраструктури досліджень енергетики дозволяє досить швидко і досить просто розвивати їх [Копайгородский 2014, с. 78-89].

Сьогодні все більше і більше онтологій використовуються для зберігання уявлень про світ в мультиагентних системах (МАС). Це дозволяє застосовувати алгоритми виводу в описі логіки, заснованої на знаннях агента, а також забезпечує більшу стандартизацію і впорядкування знань, що спрощує взаємодію між агентами. Завдання полягає у вивченні методів застосування онтологій в разі розробки і реалізації МАС, проблем і переваги подання знань агентів у вигляді онтологій, а також огляд перспектив подальшого розвитку інтеграції представлення знань системи в блоці управління агентами МАС.

Онтології широко застосовуються на різних стадіях розробки і експлуатації мультиагентних систем. Онтології використовують для проектування різних МАС, а також при реалізації МАС для вирішення завдань управління продажами і орендою нерухомості, торгівлі на біржі, підтримки безпеки інформаційних ресурсів, інформаційного пошуку та інших завдань.

- архітектура для веб-додатків датчиків - ODMAS SWAP (Ontology Driven MultiAgent System Sensor Web Agent Platform)

У той час як система на основі онтології показала деякий успіх у моделюванні просторових даних, ніхто не забезпечує всеосяжні рамки для подання всіх аспектів геопросторових даних (простір, час, тема і

невизначеність) і систем суб'єктів (агенти, послуги та процеси), які будуть служити і обробляти ці дані. Підходи агента на основі забезпечують найкращі абстракції для моделювання та управління системними сутностей. Тим не менш, у даний час інфраструктура мультиагентної системи є негнучкою і статичної, і не враховує поточне розгортання додатків і реконфігурацію. Крім того, більшість архітектур на основі агентів припускають, що онтологія вже існує, і забезпечити невелику підтримку для створення і управління онтологій [Deshendran 2009].

- Інтелектуальна система розподіленого управління малим космічним апаратом групи - Smart Satellite

Принципи застосування онтологій для вирішення нового класу задач інтелектуальних рухомих об'єктів управління поведінкою, які можуть діяти як повністю автономно, так і в якості групи в контексті вирішення завдань дистанційного зондування Землі групою малих космічних апаратів. Структура онтології та редактор онтології/сцени описують основні функції. Онтологічний опис спостережуваних об'єктів і правил розпізнавання мішеней і методи об'єктів пошуку задовольняє онтологічний опис.[Скобелев 2015]

- Мультиагентна система для контролю продукції в режимі реального часу [Иващенко 2011]

У реалізаціях мультиагентних систем онтології застосовуються для систематизації предметної області і доступних агентам знань. Можна виділити три якісно різні підходи до інтеграції онтології в роботу мультиагентной системи: або кожен агент зберігає свою онтологію, що містить доступні саме йому знання і поняття (будемо називати такі МАС розподіленими), або онтологія єдина для всіх агентів і зберігається централізовано (як правило, на спеціального агента, будемо називати такі МАС централізованими), або онтологія частково єдина, а частково - розподілена (будемо називати такі МАС гібридними) [Маркелов 2014, с. 82-87.].

Як правило, для роботи з МАС застосовуються таксономические онтології, фактично представляють знання у вигляді ієрархії. Перспективний

напрямок застосування онтологій в МАС є група роботів: об'єкти компонент системи розумний будинок, роботи помічники людей і виробництва у всіх сферах, комп'ютерного тестування, а також безпілотні літальні апарати (БЛА).

Аналіз більшості завдань, що вирішуються з використанням одиночних БЛА, показує, що вони можуть більш ефективно вирішуватися групою, так як у групи взаємодіючих літальних апаратів з'являються додаткові корисні властивості. Таким чином, для безпілотних літальних апаратів найближчим часом можуть стати актуальними проблеми групової взаємодії [21]. Одним їх можливих способів вирішення цієї проблеми і планування стратегії поведінки кожного окремого агента є використання онтологій не тільки як баз знань про зовнішній світ, але і як основи системи управління, яка застосовує автоматичний висновок на основі онтологічного знання як природний і свідомо коректний спосіб вибору поведінки для агента .

Також застосування онтологій забезпечить стандартизацію подання знань, що, імовірно, спростить обмін інформацією між гетерогенними агентами, а також дозволить при часто зустрічається в реальних проектах ситуації неможливості зв'язку між агентами відновити або передбачити наперед поведінку іншого агента з деякою точністю на основі відомих частин його онтології.[Naumenko 2017, с. 234-235]

1.5 RDF для формування середовища обміну знаннями

Мова RDF. Як зазначалося вище, в семантичній мережі інформація представлена як набір тверджень, що складається з трьох частин (трійки або триплети): суб'єкт, предикат та об'єкт. Твердження такого роду природно формують орієнтований граф, чиї вузли є суб'єкти і об'єкти кожного твердження і дуги, що їх поєднують – це предикати. Така модель даних формалізована мовою Resource Description Framework (RDF). Графи не мають коріння. Об'єднання графів концептуально є те саме, як розміщення їх поруч один з одним. Триплети самі є потужним інструментом для об'єднання даних.

Триплети є тільки колекції URI і літералів, і кожен URI або літерал має глобальну область дії. Використання глобального простору імен має вирішальне значення, оскільки це означає, що триплети завжди можуть бути об'єднані без перекладу імен. Оскільки кожен компонент графа може бути використаний без перекладу, всі графи можна транспортувати і комбінувати без перекладу, що є величезною перевагою при обміні. Оскільки твердженя RDF не треба перекладати при передачі від однієї системи до іншої, вони діють у будь-якому контексті. Вони є повністю самодостатні і, як такі, вони не залежать один від одного. Ця незалежність означає, що порядок, в якому вони наведені, не має значення. Як саме поняття графа, RDF як таке не є мовою, але є абстрактною моделлю, яка може бути серіалізованна (конкретизована) багатьма способами. Деякі з реалізованих форматів серіалізації сьогодні є досить популярні, наприклад, нотація Turtle (Terse RDF Triple Language) або нотація XML/RDF. Існує два типи вузлів: ресурси та літерали. Літерали представляють конкретні значення даних і можуть виступати тільки як об'єкти тверджень. Ресурси, навпаки, це все, що можна назвати, вони можуть бути і суб'єктами, і об'єктами. Ресурс – це ім'я, яке представляє об'єкт або поняття. Імена ресурсів мають форму згаданих вище URI. Існують особливі вузли, що називаються пустими вузлами. Пусті вузли є змінні, які використовуються для узагальнення понять. Вони дозволяють пов`язувати триплети, не турбуючись про пов`язування ресурсів. Предикати, які також називають властивостями, у свою чергу є ресурси і можуть бути представлені як URI. Розглянемо предикати RDF, які є наперед визначеними. Предикат RDF: type дозволяє віднести ресурс до певного типу, тобто ствердити, що ресурс-1 має тип, визначений ресурсом ресурс-2. Будь-який ресурс може мати багато тверджень щодо нього, у тому числі мати багато тверджень щодо його типу або взагалі не мати жодного. RDF є розширенням XML, він використовує стандартні угоди щодо введення скорочень (так звані префікси для URI простору імен). Літерали можуть представляти предикат. Атрибут rdf:datatype може вказувати, як інтерпретувати значення літералу. Літералу може бути призначений попередньо визначений

тип даних XML Schema, або тип даних користувача, визначений URI. Для строкових літералів може бути вказано мову (наприклад, англійська, російська тощо.). Твердження в RDF може бути про що завгодно, навіть про інше твердження. За допомогою твердження про твердження зручно проводити кваліфікацію або анотування інформації. Наприклад, одне твердження може містити теги джерел інформації або часові позначки додавання інформації до системи. RDF має кілька конструкцій для групування інформації. Контейнери є групи ресурсів, що можна поповнювати або зменшувати під час виконання. В RDF можна визначати три типи ресурсів, які є контейнерами: rdf:Bag використовується для представлення групи невпорядкований ресурсів, rdf:Seq представляє впорядковану множину ресурсів, rdf:Alt, як і Bag, визначає невпорядковану колекцію, але її екземпляри є еквівалентними альтернативами. RDF дозволяє групувати ресурси як списки, які не змінюватимуться навіть при злитті RDF-графів. Список можна обходити: предикат rdf:first посилається на перший елемент у списку, предикат rdf:rest відноситься до решти списку, яка першим елементом має другий ресурс первинного списку. Цей процес продовжується рекурсивно, поки список rdf:rest не стане rdf:nil (пусто). [Андон 2012]

1.6 Огляд існуючих засобів побудови онтології

Для створення та редагування онтологій розроблено ряд спеціалізованих середовищ розробки, редакторів, парсерів та засобів об’єднання онтологій, найбільш ефективними з яких є: KAON, OilEd, OntoEdit, Ontosaurus, OpenCyc, Protégé.

Серед цих інструментів для побудови предметно-орієнтованої онтології виді- лимо редактор Protégé-OWL, як гнучке, незалежне від платформи середовище з своїми особливостями та перевагами, яке забезпечує наочний та зручний у використанні графічний інтерфейс користувачу, реалізує масштабованість, тобто модульне нарощування системи в рамках уніфікованої

архітектури, дає змогу нарощувати архітектуру за допомогою додатково розроблених підпрограм – плагінів (plug-in). Також Protégé-OWL дає змогу описувати класи з використанням нових можливостей. Зокрема, мова OWL (Ontology Web Language) має великий набір операторів і базується на логічній моделі, яка дозволяє давати визначення поняттям так, як вони описані, тому складні комплексні поняття у визначеннях можуть бути створені з простіших. До того ж логічна модель дає змогу використовувати механізм міркувань (Reasoner), котрий у свою чергу дає змогу перевірити чи твердження і визначення в онтології є взаємно несуперечливими, а також розпізнати відповідність визначень певним поняттям. Завдяки цьому механізму підтримується правильність ієрархії онтології.

Редактор також підтримує більшість реляційних баз даних (Oracle, MySQL, Microsoft SQL Server, Microsoft Access).

У редакторі Protégé-OWL забезпечено можливість вибору однієї з трьох розроблених на цей час версій мови OWL: OWL-Lite, OWL-DL, OWL-Full.[Досин 2008]

OWL — мова опису онтологій для семантичної павутини. Мова OWL дозволяє описувати класи і відношення між ними, властиві для веб-документів і додатків[Hayes 2004, с. 136]. OWL заснована на більш ранніх мовах OIL і DAML OIL і в наш час є рекомендованою консорціумом Всесвітньої павутини. В основі мови — уявлення дійсності в моделі даних «об'єкт — властивість». OWL придатна для опису не тільки веб-сторінок, але і будь-яких об'єктів дійсності. Кожному елементу опису в цій мові (в тому числі властивостями, що зв'язує об'єкти) ставиться у відповідність URI.

1.6.1 Різновиди мови

OWL Lite призначена для користувачів, які потребують передусім класифікаційної ієрархії і простих обмежень. Наприклад, при тому, що вона підтримує обмеження кардинальності (кількості елементів), допускаються значення кардинальності тільки 0 або 1. Для розробників повинно бути

простіше в своїх продуктах забезпечити підтримку OWL Lite, чим виразніших варіантів OWL. Зокрема, OWL Lite дозволяє швидко перенести існуючі тезауруси і інші таксономії. OWL Lite також має нижчу формальну складність, ніж OWL DL.

OWL DL призначена для користувачів, яким потрібна максимальна виразність при збереженні повноти обчислень (всі логічні висновки, що припускаються тією чи іншою онтологією, будуть гарантовано обчислюваними) і розв'язуваності (всі обчислення завершаться за певний час). OWL DL включає всі мовні конструкції OWL, але вони можуть використовуватися лише згідно з певним обмеженням (наприклад, клас може бути підкласом багатьох класів, але не може сам бути представником іншого класу). OWL DL так названий з-за його відповідності дескрипційній логіці — дисципліні, в якій розроблені логіки, що складають формальну основу OWL.

OWL Full призначена для користувачів, яким потрібна максимальна виразність і синтаксична свобода RDF без гарантій обчислення. Наприклад, у OWL Full клас може розглядатися одночасно як сукупність індивідів і як один індивід у своєму власному значенні. OWL Full дозволяє будувати такі онтології, які розширюють склад зумовленого (RDF або OWL) словника. Малоймовірно, що будь-яке програмне забезпечення зможе здійснювати повну підтримку кожної особливості OWL Full.

1.6.2 Структура мови

Онтології OWL складаються з окремих компонентів, таких як індивіди, класи, властивості об'єкта.

Індивіди(Individuals). Являють собою конкретні об'єкти певної предметної області. У OWL не використовується припущення про унікальність імен (Unique Name Assumption - UNA). Це означає, що два різних імені можуть фактично посилатися на один і той же індивід. В OWL два індивіда можуть позначати один і той же об'єкт предметної області якщо явно не вказано інше.

Властивості (Properties). Це бінарні відношення на індивідах. Іншими словами, відношення по'єднують двох індивідів.

Класи (Classes). Це множини, елементами яких є індивіди. Вони описуються, використовуючи формальні (математичні) конструкції, які декларують вимоги для членства в класі. Класи можуть бути організовані в ієрархію відношень виду «підкласс-суперклас», яка також відома як таксономія. Підкласи є підмножинами свого суперкласу. Одна з головних особливостей OWL-DL - те, що ці відношення підкласу і суперкласу (відношення категоризації) можуть бути обчислені автоматично за допомогою різонерів(детальніше в розділі 2). В OWL класи мають описи, які визначають умови, яким повинен задовольняти індивід для того, щоб бути членом класу. Всі класи – підкласи класу owl:Thing, клас-корінь онтології, у DL позначається як ⊤. Найнижчий у ієрархії клас owl:Nothing є підкласом усіх класів онтології. Позначається як ⊥. Їх використовують для отримання певної інформації про всі об’єкти онтології або про жоден [Hayes 2004, с. 136].

1.7 Аналіз семантичних різонерів

1.7.1 Поняття різонера

Семантичний різонер - частина програмного забезпечення, що здатна виводити логічні висновки з набору вибраних фактів та аксіом, а також надає можливість автоматичної підтримки таких завдань як: класифікація, налагодження, формування запитів. Правила виведення зазвичай задаються засобами мови онтологій, і часто засобами мов описової логіки. Використання різонерів для автоматичного виведення ієрархії класів є одним з основних переваг побудови онтології з використанням мови OWL DL [Cornet 2003, с. 329]. При проектуванні дуже великих онтологій (понад декілька тисяч класів) без різонера дуже важко обслуговувати великі, складні онтології, і зберігати в коректному, належному вигляді.

В OWL реазонери працюють на основі концепції OWR – «Open World Reasoning». Open World Reasoning буквально перекладається з англійської як «міркування про відкритість світу». В OWL DL - це категорія, що позначає відкритість баз знань, заснована на припущенні про відкритість світу (open world assumption), що принципово відрізняє їх від баз даних, заснованих на припущення про замкнутість світу (closed world assumption) [Bock 2008, с. 54]. По суті це означає неповноту знань по відношенню до фактів, які можна вивести на їх основі шляхом логічних міркувань. Тобто побудова повної класифікації за описами окремих класів, які здавалося б, не дають повного опису онтології. Це можна розуміти так: якщо «щось» не було сказано, для того щоб бути фактом, не можна припустити його хибним - вважається, що «знання просто не було додано до бази знань». Створення явних припущень у предметній області, що лежать в основі реалізації, дає можливість легко змінити ці припущення при зміні наших знань про предметну область. Жорстке кодування припущень про світ на мові програмування призводитьдо того, що ці припущення не тільки складно знайти і зрозуміти, але і також складно змінити, особливо непрограмістам [Schalkoff 2009, с. 351]. Без розуміння принципів побудови онтологій, робота реазонера може здаватися непомітною.

1.7.2 Основні задачі різонерів

Серед основних задач різонерів можна виділити такі:

1. Проводити класифікацію і виводити ієрархію класів.
2. Перевіряти консистентність онтології.
3. Визначати тип індивіда - належність до певного класу.
4. Визначати класи, що не перетинаються з заданим класом.
5. Визначати підкласи вибраних класів.

2.3 Основні алгоритми побудови різонерів

1.7.3 Загальні відомості про правила виведення

У логіці правило виведення, правило виводу або правило перетворення — це логічна форма, що складається з функції, яка отримує передумови, аналізує їхній синтаксис і повертає висновок (або висновки). Наприклад, правило виведення, що називається modus ponens, отримує дві передумови, одну у формі «Якщо p тоді q», а другу у формі «p», і повертає висновок «q». Це правило є чинним відносно семантики класичної логіки (як і відносно семантик багатьох інших некласичних логік), у тому сенсі що якщо передумови є істинними (в межах інтерпретації), то істинним є і висновок.

Зазвичай правило виведення зберігає істинність, семантичну властивість. У багатозначній логіці воно зберігає узагальнене значення. Але дія правила виведення є винятково синтаксичною, і не потребує зберігання ніякої семантичної властивості: будь-яка функція з множин формул до формул вважається правилом виведення [Hayes-Roth 1983, с. 109].

Modus ponens (латиною: метод що підтверджує) - коректна, проста форма аргументації:

Якщо P, то Q.

P.

Звідси Q.

або у логіко-операторному записі:

$$p \to q,$$

$$p \vdash q \tag{1.1}$$

де $\vdash$ означає логічний висновок. Аргумент має два вихідних твердження. Перше це умова «якщо-то» або «умовне» твердження, а саме що із P слідує Q. Друге твердження це те що P, умовна частина першого твердження, є істиною. Із цих двох умов логічно слідує що Q, висновок першого твердження, мусить бути істиною також [Hayes-Roth 1983, с. 109]

Той факт, що висновок є коректним не ґарантує істиності вихідних тверджень. Коректність modus ponens каже нам тільки те, що висновок є

істинним тоді і тільки тоді, коли всі вихідні твердження є істинними. Слід нагадати, що коректний логічний висновок, у якому одне або більше вихідних тверджень не є істинними, називають необґрунтованим, інакше, якщо усі твердження є істинними, такий аргумент називають обґрунтованим. У більшості логічних систем Modus ponens вважається коректним, хоча кожен окремий випадок застосування може бути або обґрунтованим або ні. Висновок логіки висловлювань із використанням modus ponens є дедуктивним.

Modus tollens (латиною: спосіб, що заперечує) це формальна назва для доведення від супротивного.

Modus tollens є простою, часто вживаною формою аргументації:

Якщо P, то Q.

Q є фальш.

Тому P є фальш.

Використовуючи логіко-операторну нотацію:

$$p \to q,$$
$$\neg q$$
$$\vdash \neg p. \qquad (1.2)$$

1.7.4 Пряме виведення

Пряме виведення (forward chaining) є одним з двох основних методів логічного виведення. Є популярною стратегією впровадження експертних систем, бізнесових та продукційних моделей, базованих на правилах [Hayes-Roth 1983, с. 109]. Алгоритм починається з формування ланцюжка з наявними даними і використовує правила виведення для вилучення більшої кількості даних (від кінцевого користувача, наприклад), поки мета не буде досягнута. У системах з прямим виведенням за відомимми фактами відшукується факт, який з них випливає. Якщо такий факт вдається знайти, то він записується в базу фактів. Пряме виведення називають також виведенням, керованим даними або виведенням, керованим посиланнями правил.

У механізму прямого виведення архітектура складається з таких частин:

- Робоча пам’ять, у якій зберігаються наведені факти
- Правила, яким задовольняють ті чи інші факти за певних умов
- Дії, які включають в себе додавання або видалення фактів з робочої пам’яті

Однією з переваг прямого логічного виведення є те, що отримання нових даних може спровокувати виникнення нових висновків, що робить програмний юніт більш гнучким для систем, які динамічно змінюються [Kaczor 2010, с 134].

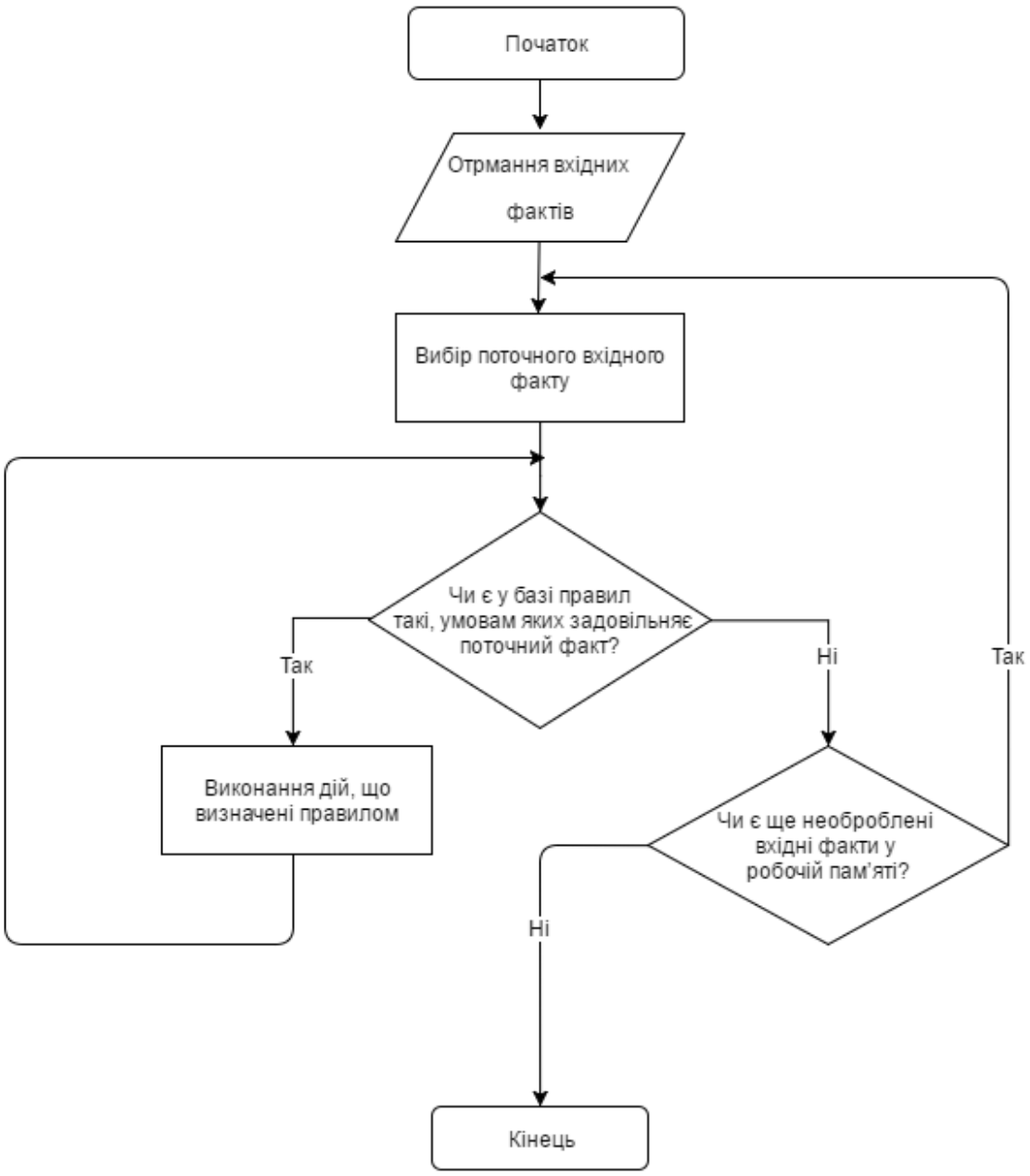

Рисунок 1.7 – Блок-схема алгоритму прямого виведення

1.7.5 Зворотне виведення

Зворотний вивід (або зворотне міркування) - це метод отримання висновку, який працює в зворотному напрямку від мети. Він використовується в автоматичному доведенні теорем, машинному виведенні та інших напрямках штучного інтелекту [Hayes-Roth 1983, с. 109].

Детальніше роботу алгоритму зворотного виведення можна розглянути на рисунку 1.8, де зображена його блок-схема.

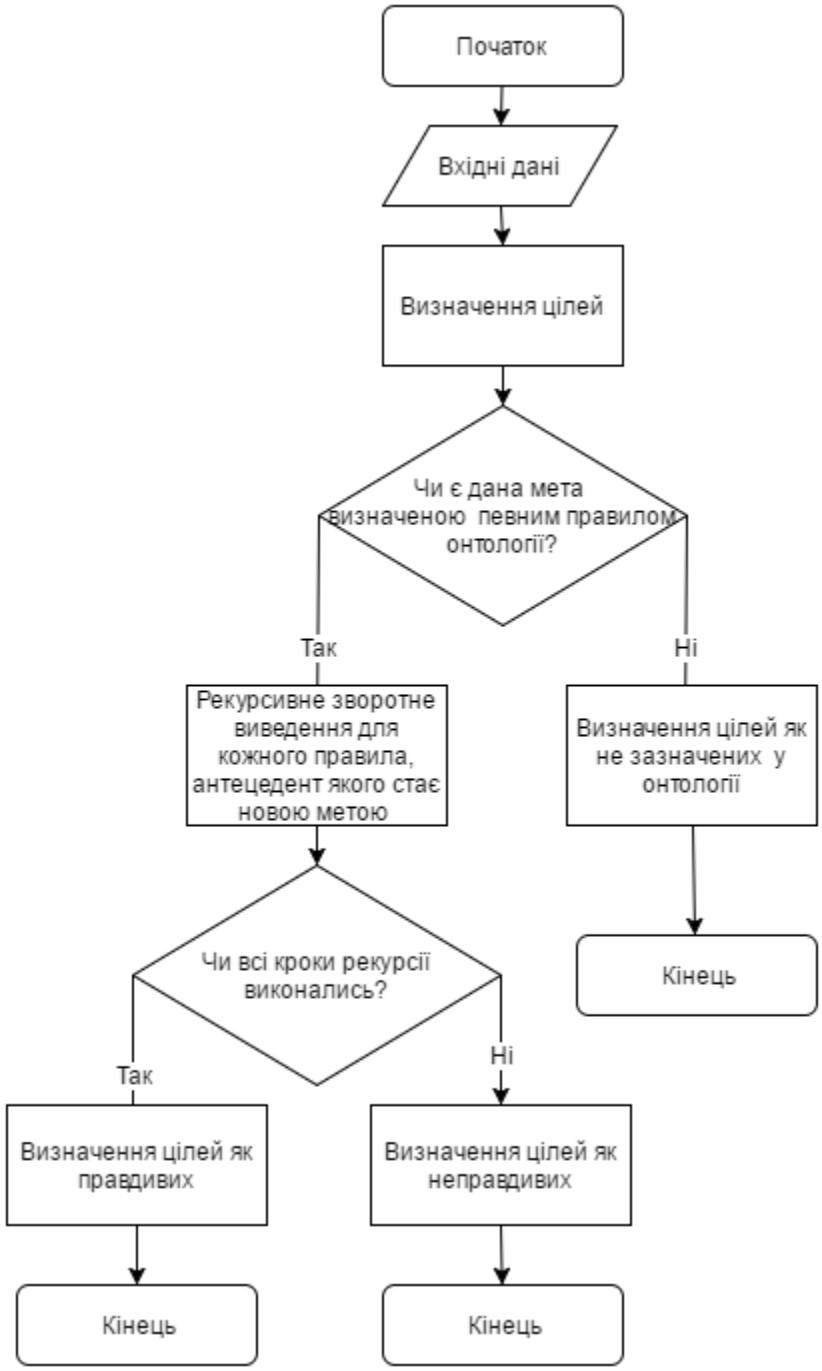

Рисунок 1.8 – Блок-схема алгоритму зворотного виведення

Оскільки саме список цілей визначає, які правила вибирати і використовувати, цей метод називається методом, керованим метою, на відміну від прямого виводу, що є методом, керованим даними. Зворотний вивід часто використовується в експертних системах [Kaczor 2010, с 134].

Недоліком алгоритму є те, що мета повинна бути точно вказаною, що унеможливлює його використання для обширних запитів. Тому зазвичай для реалізації різонера на базі алгоритмів виведення використовують і зворотне, і пряме виведення.

1.7.6 Алгоритм семантичних таблиць

Метод семантичних таблиць - це формальна роздільна процедура для формул логіки висловлювань і логіки предикатів, що дозволяє чисто синтаксичними засобами вирішувати семантичні проблеми формалізованих обчислень.

Семантична таблиця - це дерево, вершинами якого є досліджувана формула і все її подформули. Вершина називається особливою, якщо вона відмінна від атома, тобто її логічне значення залежить від останньої виконуваної логічної зв'язки. Вершина називається звичайною, якщо в ній знаходиться атом [Бет 1967, с. 223]. Останні вершини кожної гілки - це обов'язково атоми, і такі вершини називаються листками. Семантична таблиця складного висловлювання К будується індуктивно, виходячи з семантичних таблиць підформул, що входять в висловлювання К. Кожній логічній зв'язці, що виконується у відповідній даній вершині подформулі, зіставляється елементарна семантична таблиця у вигляді дерева, що розкриває логічну інтерпретацію зв'язки. Приклади найбільш вживаних в логіці висловлювань зв'язок наведені на рисунку 1.9

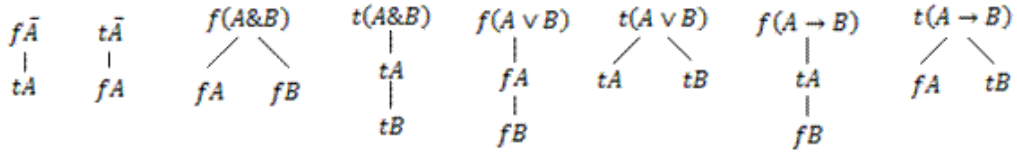

Рисунок 1.9 - Елементарні семантичні таблиці [Бет 1967, с. 223]

Семантична таблиця складного висловлювання К - це граф, який не має циклів (дерево), з єдиним коренем, в який поміщається позначена формула К. Кожна гілка дерева інтерпретується як кон'юнкція всіх підформул, записаних в вершинах цієї гілки. Семантична таблиця будується зверху вниз, виходячи з індуктивного правила визначення формули: неподільні висловлювання (літерали) - це атоми; з атомів будуються формули з використанням логічних зв'язок. Кожна формула є результатом застосування одно- або двомісної логічної зв'язки до підформул, що її утворюють [Драгалин 1982, с. 75].

1.7.7 Огляд існуючих різонерів

В даний час існує величезна кількість різонерів, проте зародження їх почалося в період з 1975 року по 2009 рік. Можна аналізувати існуючі різонери за чотирма ознаками:

- Підтримка інтерфейсу
- Вид алгоритму і завершеність
- Мова розробки
- Підтримка мов розробки семантичних мереж

Існує, щонайменше, дев'ятнадцять різонерів. Серед них можна виділити ті, які є досить популярними, а саме: Pellet, RACER, FACT ++, Snorocket, HermiT, CEL, ELK, SWRL-IQ і TrOWL, Structural Reasoner [Gardiner 2006. – 8 с.].

Pellet є вільно поширюваним OWL-DL різонером, реалізований на Java. Розробники, The Mind Swag Group, в основу даного reasoner-а поклали табличні алгоритми і підтримку виразного опису логіки. Це перший в світі різонер, який підтримує всі OWL DL SHOIN і був розширений до OWL2 . Імплементує OWL API інтерфейс. Також є підтримка пояснення помилок. В основі роботи лежить алгоритм семантичних таблиць.

RACER - Renamed ABoxes and Concept Expression Reasoner – це один з популярних різонерів, який був розроблений Яном Хорроксс. RACER, так само відомий як RacerPro є першим в історії reasner-ом. Підтримка оптимізаційних

технологій FACT, а так само підтримка нових методів для роботи з чисельним обмеженням і ABoxes. Racer реалізує TBOX і ABOX різонери для описової логіки SHIQ.

FACT ++. Ян Хоррокс представив різонер, що відомий під назвою FACT ++. Даний блок міркувань може бути використаний як класифікатор описової логіки і тестування досяжності модальної логіки. Система FACT озвучена і повністю оснащена підтримкою алгоритмів для описової логіки. Після випущених оновлень FaCT перетворився в FaCT ++. Відмінність між ними полягає у внутрішній структурі, яка написана на C ++. Перша версія FaCT ++ підтримувала тільки SHOIQ, OWL-DL. Однак, в останніх версіях з'явилася підтримка OWL цілком і за основу взята описова логіка SROIQ.

SnoRocket являє собою високопродуктивний поліноміальний алгоритм класифікації для описової логіки EL +. Реалізований даний алгоритм на мові Java. Його розроблено як частину «CSIRO's Health Informatics» і «Clinical Terminologies research program».

SWRL-IQ. Розшифровується як Semantic Web Rule Language Inference and Query tool, і є плагіном для Protégé 3.5, який дозволяє користувачеві змінювати, зберігати, записувати запити та передавати їх до базового механізму логічного висновку, заснованого на XSB Prolog.

ELK є у вільному доступі. Reasoner для мови онтологій OWL 2 EL. ELK різонер написаний на Java і контролюється за допомогою OWL API. Доступний тільки під Apache License 2.0. Кросплатформність здійснюється за рахунок підтримки Java 1.5 або вище.

Hermit - це перший загальнодоступний OWL-різонер, який написаний за допомогою OWL API. З його допомогою можна перевірити правильність складання онтологій в OWL-файлах і скласти ієрархічну структуру класів. Обчислення будуються за допомогою гіпертабличного числення .

CEL. Даний реазонер був створений на мові програмування LISP. Володіючи простим інтерфейсом у вигляді командного рядка, CEL надає

користувачам всі необхідні функції, включаючи прості інтерактивні команди допомоги. В основу різонера лягли алгоритми прямого і зворотного виводу.

TrOWL є загальним інтерфейсом для ряду різонерів, які розроблені в університеті Абердіна. Виділяють два основних види: TrOWL Quill і TrOWL REL. Quill надає зв'язок зі службами OWL 2 QL. TrOWL REL це свого роду оптимізований CEL алгоритм, який забезпечує зв'язок з OWL 2 EL. Для підтримки повного OWL DL, TrOWL використовує інші різонери, такі як FACT ++, Pellet, Hermit, RacerPro і т.д.

Structural Reasoner. Це структурний різонер, який наразі не є повним, але достатньо часто використовується в контексті OWL-API, де являється одним зі стандартних різонерів фреймворку. Розроблений на Java, використовує алгоритми прямого та зворотного виведення.

Таблиця 1.1 – Порівняльний аналіз різонерів

Різонер / Критерій		Pellet	RACER	FACT++	Snorocket	SWRLIQ	Hermit	CEL	TrOWL	ELK
Алгоритм		Семан-тичних таблиць	Семан-тичних таблиць	Семан-тичних таблиць	Правил виве-дення	SWRL правид	Гіпер-табличного числення	Правил виве-дення	Правил виведен-ня	Правил виведен-ня
Стійкість		Так	Так	Так	Так	Так	Так	Так	Так	Так
Повність		Так	Так	Так	Так	Ні	Так	Так	Так	Так
Виразність		SROIQ (D)	SHIQ	SROIQ (D)	EL+	-	SROIQ (D)	EL+	SHIQ	EL
Нативний профіль		DL, EL	DL	DL	EL	-	DL	EL	DL, EL	EL
Інкремен-тальна класифі-кація	Дода-вання	Так	Ні	Ні	Так	Ні	Ні	Так	Ні	Так
	Вида-лення	Так	Ні	Ні	Ні	Ні	Ні	Ні	Ні	Так
Підтримка правил		Так (SWRL)	Так (SWRL)	Ні	Ні	Так (SWRL)	Так (SWRL)	Ні	Ні	Так
Платформи		Всі	Всі	Всі	Всі	Всі	Всі	Linux	Всі	Всі
Обґрунтування		Так	Так	Ні	Ні	Так	Ні	Так	Ні	Ні
Виведення логічних висновків щодо індивідів		Так	Так	Так	Ні	Так	Так	Так	Так	Ні
Підтримка OWL API		Так	Так	Так	Так	Ні	Так	Так	Так	Так
Підтримка OWL Link API		Так	Так	Так	Ні	Ні	Так	Так	Ні	Ні
Підтримка Protégé		Так	Так	Так	Так	Так	Так	Так	Так	Так
Підтримка NeOn		Так	Ні	Ні	Ні	Ні	Так	Ні	Ні	Ні
Ліцензія		DULI: AGPL	Власна	GLGPL	Власна	Так/Ні	GLGPL	Apache License 2.0	DULI: AGPL	Apache License 2.0
Підтримка Jena		Так	Ні	Ні	Ні	Ні	Ні	Ні	Так	Ні
Мова написання		Java	Lisp	C++	Java	Prolog	Java	Lisp	Java	Java
Доступ		Відкри-тий	Комер-ційний	Відкри-тий	Комер-ційний	Комер-ційний	Відкри-тий	Відкри-тий	Комер-ційний	Відкри-тий

Висновки до розділу 1

У даному розділі було проаналізовано теоретичні відомості по вибраній темі. Визначено поняття інформаційної системи, її функції, а також класифікації. Також було виділено мультиагенету систему як приклад інформаційної системи. Проведено роз'яснення мультиагентного підходу, визначено поняття агент та його функції. Проаналізовано процеси, що протікають між агентами в мультиагнетній системі, іх спілкування та адаптивність. Було проведено пошук областей, в яких вже використовуються мультиагентні системи, і які ще є перспективні але не зайняті області. Результатом став обраний напрямок – мультиагнета система з онтологіями в якості баз знань для керування групою інтелектуальних роботів.

Проведено аналіз концепцій особливості визначення та структури онтологій у контексті комп'ютерних наук, їх класифікацій.

Було проаналізовано мову OWL для створення та редагування онтологій

семантичної павутини, яка дозволяє описувати класи і відношення між ними, властиві для веб-документів і додатків. Охарактеризовано компоненти, з яких зазвичай складаються онтології, описані за допомогою OWL, їхні особливості та суть застосування.

Розглянуто основні різновиди мови OWL (OWL Lite , OWL DL, OWL Full), проаналізовано можливості кожного з них.

Визначено поняття семантичного різонера. Було виділено головні задачі різонерів (проведення класифікації і виведення ієрархію класів, перевірка консистентності онтології, визначення типу індивіда – належнісості його до певного класу, визначення класів, що не перетинаються з заданим класом, визначення підкласів вибраних класів).

Проаналізовано найчастіше використовувані при проектуванні різонерів алгоритми(прямого та зворотного виведення, побудови семантичних таблиць, побудови гіпертаблиць)

Було здійснено огляд існуючих різонерів, особливостей їх реалізації.

2 МАС ДЛЯ УПРАВЛІННЯ ГРУПАМИ ІНТЕЛЕКТУАЛЬНИХ РОБОТІВ

Гарним прикладом використання інтелектуальних агентів служать системи управління роботами. Роботи можуть мати широкий асортимент штучних органів почуттів (сенсорні датчики) і штучних ефекторів (маніпулятори, педіпулятори). Йдеться про робототехнічних пристроях, що виконують завдання, пов'язані з переміщеннями в просторі. Їх мобільність досягається завдяки колісним, гусеничним, крокуючим і іншим системам переміщення. Активність і автономність роботів тісно пов'язані з наявністю коштів мети й планування дій, систем підтримки вирішення завдань, а інтелектуалізація, крім володіння системою обробки знань, передбачає розвинені засоби комунікації різних рівнів, аж до засобів природньомовного спілкування.

Невід'ємним атрибутом інтелектуальних роботів є наявність спеціальної підсистеми планування, складовою програму дій робота в реальних умовах навколишнього середовища, які визначаються рецепторами робота. Для планування діяльності робот повинен мати знання про властивості навколишнього середовища і шляхи досягнення цілей в цьому середовищі.
Розглянемо два підходи до управління групою інтелектуальних роботів.

Перший заснований на ринкових відносинах в групі роботів. Його суть полягає в наступному. Спочатку кожен робот генерує список цілей. Цілі, що відповідають відомим областям середовища, виключаються зі списків, а що залишилися розміщуються в порядку проходження. Кожен робот, по черзі, намагається продати кожну з цільових завдань всім іншим роботам групи, з якими можливий зв'язок, виставляючи їх на аукціон. Кожен з решти роботів пропонує ціну, яка визначається приблизною витратами і доходами. Робот-аукціоніст, що пропонує цільову завдання, чекає, поки всі роботи не запропонують ціну. Причому на це відводиться певний час. Якщо будь-які

роботи пропонують ціну, велику ніж мінімум, запропонований роботом-аукціоністом, то цільова завдання передається тому з них, який пропонує найбільшу ціну.

Після того, як всі аукціони проведені, роботи починають рух до першої своєї мети. За її досягненні кожен робот генерує нові цілі і починає рух до наступної мети, що міститься в завданні, пропонуючи все решта в списку цільові завдання іншим роботам групи через аукціон. Однак в цьому підході використовуються дуже громіздкі механізми аукціонів і регулювання цін, що істотно обмежує його застосування.

Інший підхід до організації мультиагентної взаємодії в групах інтелектуальних роботів заснований на принципах колективного управління, характерних для колективів людей.

Колективом називають групу роботів, які вирішують загальну цільову завдання і взаємодіючих між собою для вирішення цього завдання найкращим для групи чином. Колективне взаємодія - це взаємодія, що охоплює велику кількість елементів деякої системи і виявляється в їх узгоджені дії.

Метод колективного управління полягає в тому, що кожен робот групи, по-перше, самостійно керує процесом свого функціонування, т. Е. Визначає свої дії, а по-друге, погоджує ці дії з діями інших роботів групи, для того щоб найбільш ефективно, т. е. з мінімальними витратами і максимальною вигодою для групи, вирішити цільове завдання.

Основні принципи колективного управління:

- кожен член колективу групи самостійно формує своє управління (визначає свої дії) в поточній ситуації;
- формування управлінь (вибір дій) кожним членом колективу здійснюється тільки на основі інформації про колективну мети, що стоїть перед групою, ситуації в середовищі в попередній відрізок і в поточний момент часу, свій поточний стан і поточних діях інших членів колективу;
- в якості оптимального управління (дії) кожного члена колективу в поточній ситуації розуміється таке управління (дія), яка вносить

максимально можливий внесок у досягнення спільної (колективної) мети або, іншими словами, дає максимально можливе збільшення цільового функціоналу при переході системи "колектив-середовище "з поточного стану в кінцеве;

- оптимальне управління реалізується членами колективу протягом найближчого відрізка часу в майбутньому, а потім визначається нове управління;
- допускається прийняття компромісних рішень, які відповідають всіх членів колективу, тобто кожен член колективу може відмовлятися від дій, що приносять йому максимальну вигоду, якщо ці дії приносять малі вигоди або навіть шкоду колективу в цілому.

На відміну від групового управління, яке може бути як централізованим, так і децентралізованим, колективне управління групою роботів завжди децентралізоване за своєю суттю. Тому описаний метод колективного управління роботами найбільш ефективний при реалізації в розподілених мультиагентних системах.

Основна перевага - відносно низька обчислювальна складність алгоритмів, що дозволяє швидко приймати якщо не оптимальні, то близькі до них рішення в умовах динамічно змінюється. Цей підхід застосовується не тільки для вирішення завдань колективного управління інтелектуальними роботами, але і для вирішення проблеми організації відмовостійкості обчислювального процесу в розподілених багатопроцесорних інформаційно-керуючих системах складних динамічних об'єктів.

Ще одна важлива особливість мультиагентного підходу - наявність агентной платформи. Агентна платформа (АП) - це middleware, яке реалізує основні механізми, що забезпечують роботу МАС і, таким чином, полегшує побудову агентних систем. МАС працює "поверх" агентной платформи і використовує її сервіси. Основні функції АП:

- Взаємодії агентів;

- Передача повідомлень між агентами всередині платформи (на різних рівнях: на рівні мережевих пакетів, повідомлень будь-якої мови спілкування, протоколів спілкування);
- Передача повідомлень між агентами різних платформ;
- Підтримка онтологій;
- Управління агентами;
- Пошук агентів і даних про них усередині системи;
- Управління життєвими циклами агентів;
- Безпека.

В рамках дипломного проекту була поставлена задача побудови МАС для інтелектуальних роботів і вживлення до їх архітектури онтології для адаптивної взаємодії один з одним.

2.1 Мультиагентна платформа JADE

The Java Agent Development Framework (JADE) є відкритим вихідним кодом, повністю сумісним с FIPA проміжним програмним забезпечення для мультиагентних систем, написаних на Java. Розробка почалася в кінці 1998 року Telecom Italia. JADE був з відкритим вихідним кодом і розповсюджувався Telecom Italia під Library Gnu Public License (LGPL) в 2000 р. Інші організації, такі як Motorola і France Telecom R & D, стали членами правління [Bellifemine et all, 2007, с. 29-30].

JADE не є доменом або додатком безпосередньо. Він пропонує основні функціональні можливості проміжного програмного забезпечення для розподілених додатків, заснованих на мультиагентній парадигмі. Середовище JADE володіє цікавими властивостями [Bellifemine et all, 2007, с. 30-31]:

- Агенти є автономними і проактивними: у них є свій власний потік виконання і вони не передають посилання на їх об'єкт іншим агентам. Вони контролюють свій життєвий цикл і самостійно вирішують, коли виконувати свою наступну дію.

- Агенти можуть відмовитися від виконання дії і слабо пов'язані: JADE агенти спілкуються асинхронно та за допомогою передачі повідомлень. Відправник звертається до приймача, використовуючи його унікальне ім'я, а не посилання на його об'єкт. Повідомлення навіть можуть бути відправлені без вказування відправника, відправивши його до групи агентів або до проксі агента, який буде займатися доставкою відповідним агентам. Крім того, приймач може самостійно вирішити, які повідомлення він хоче обробляти і які повідомлення він хоче відхилити на свій розсуд.
- Система є з'єднанням рівноправних вузлів: Агенти мають унікальні ідентифікатори (The AgentIdentifier (AID)), щоб безпосередньо звертатися один до одного, вони можуть приєднатися чи залишити приймаючу платформу на свій вибір. Агенти можуть шукати інших агентів в жовтих сторінках (надається Agent Management System (AMS) і Directory Facilitator (DF) агентами).

2.1.1 Архітектура JADE

Агентна платформа JADE складається з одного або більше контейнерів, розподілених по мережі. JADE контейнери є Java-процесами, які забезпечують JADE середовищем виконання і сервісами для розміщення та виконання агентів JADE. Основний контейнер має особливе значення. Він являє собою точку початкового завантаження агентної платформи і запускає інші контейнери. Контейнери ідентифікуються по їх логічним ім'ям (за умовчанням головний контейнер називається "Головний Контейнер", інші "Container-1" і т.д.) [Bellifemine et all, 2007, p. 32]. Головний Контейнер має наступні додаткові функціональні можливості [Bellifemine et all, 2007, с. 32-33]:

- Управління Container Table (CT): CT містить посилання на об'єкт і транспортну адресу кожного контейнера, який є частиною агентної платформи.
- Управління Global Agent Descriptor Table (GADT): GADT реєструє кожен

- агент в платформі з його місцезнаходженням і станом.
- Хостинг агентів DF і AMS, що забезпечує управління агентами, сервіси білі і жовті сторінки.

Кожен контейнер має Local Agent Descriptor Table (LADT) і (за винятком основного контейнера) локальний кеш GADT, щоб не було проблем з GADT. Коли контейнер повинен доставити повідомлення, він дивиться на адресу агента в його LADT. Тільки якщо пошук невдалий, він буде намагатися знайти його в GADT і зберегти результат в своєму кеші GADT. Щоб вберерти основний контейнер від того, що він буде єдиною точкою відмови, JADE пропонує механізми реплікації, що дозволяють агентній платформі функціонувати, коли основний контейнер недоступний [Bellifemine et all, 2007, стор. 33].

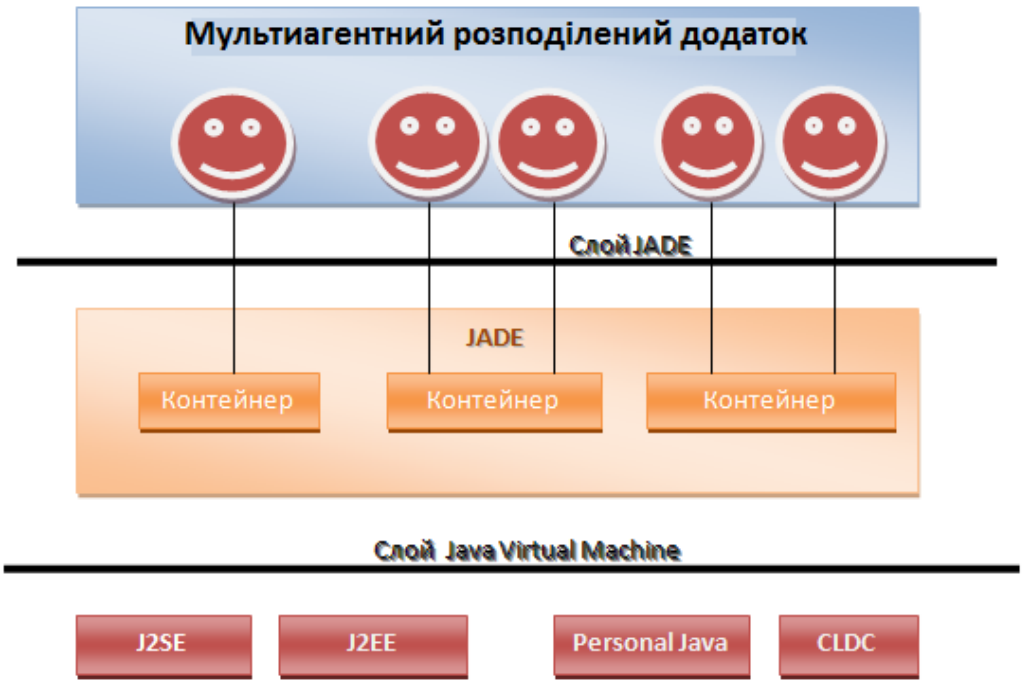

Рисунок 2.1 – Архітектура додатку на JADE

2.1.2 Використання онтологій в мультиагентній платформі JADE

В системі Protégé онтології можуть бути збережені в різних форматах (RDF(S), XML, OWL). JADE агенти, в свою чергу, використовують онтологію в термінах Java об’єктів.

Цей розділ відображає як перетворення з сумісної з Protégé онтології може відбутися в зрозумілий для JADE вигляд за допомогою плагіна BeanGenerator.

2.1.2.1 Content Reference Model

JADE пропонує Content Reference Model (CRM), яка представляє собою класифікацію всіх можливих елементів, які відбуваються в області дискурсу.

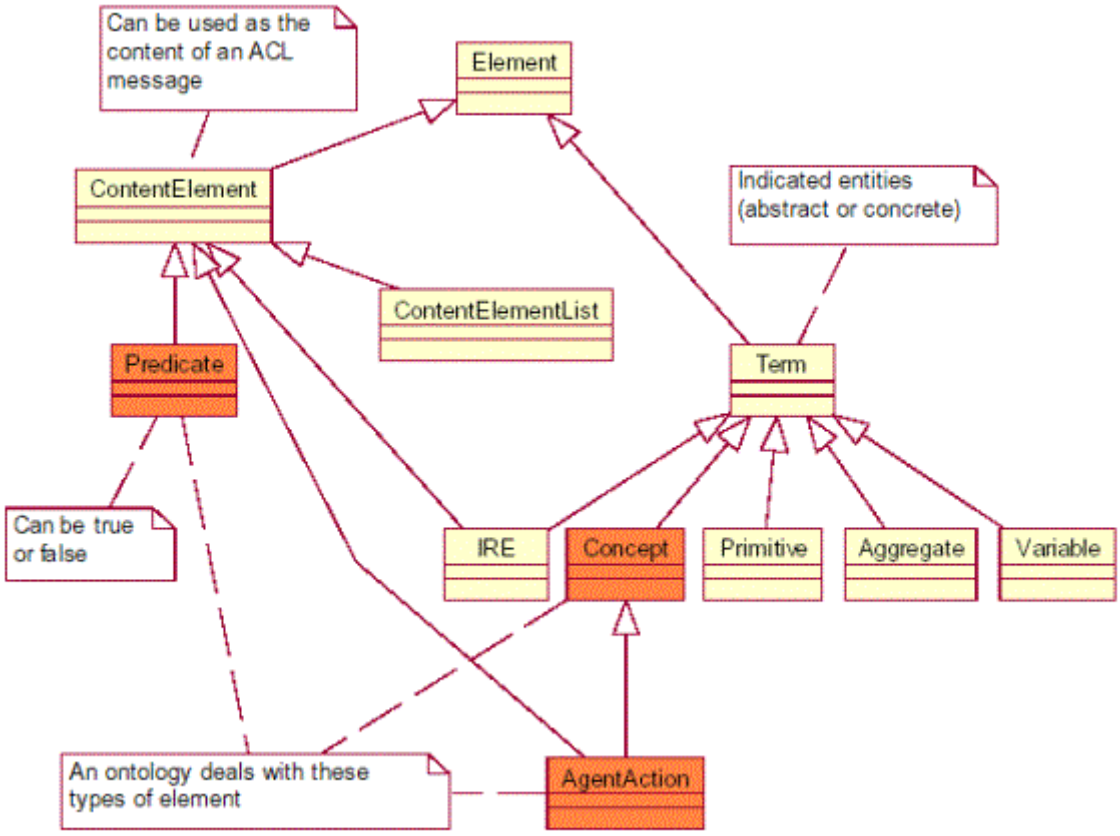

Рисунок 2.2 - JADE CRM [Ontology support in JADE 2007, с.1]

Більш детально, типи елементів, з якими зв'язана онтологя, визначаються наступним чином:

- Concepts - вирази, які вказують на об'єкти, які «існують» і про які спілкуються агенти.
- Predicates - вирази, які говорять щось про стан світу і, як правило, оцінюють до істинни або хиби.

- AgentActions - вирази, які вказують на те, що може бути виконане деяким агентом.

Як вище зазначено, представлення онтологій в JADE і Protégé є несумістними. Для того, щоб можна було перетворити онтологію в сумістну з JADE необхідно включити до онтології *Concept, Predicate, AgentAction.*

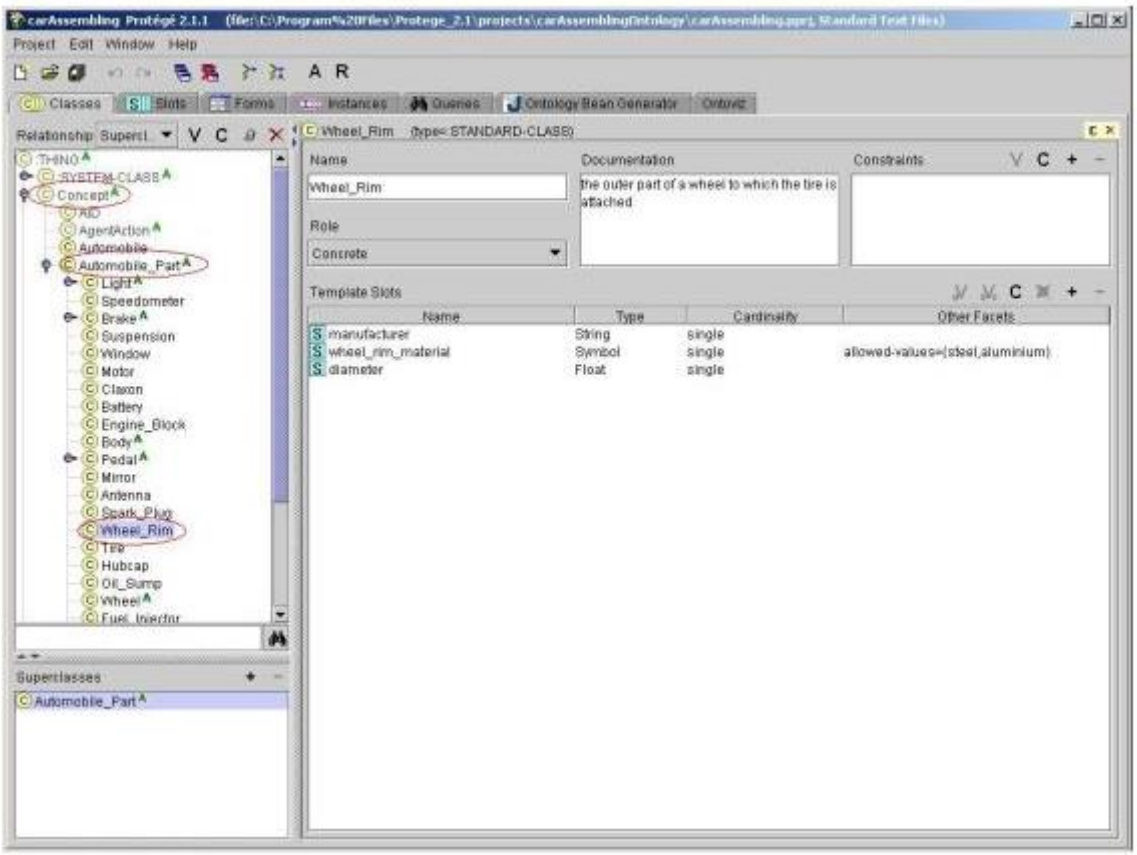

Рисунок 2.3 - визначення підкласів класу *Concept* в Protégé

2.1.2.2 Плагін BeanGenerator

Якщо в процесі визначення онотологій використовувалась CRM, класи для об'єктно-орієнтованих мов програмування можуть бути побудовані на фундаментальній онтології і безпосередньо використані для обміну інформацією. Плагін *BeanGenerator* був спеціально для цього створений для виконання цього завдання, оскільки він може автоматично мігрувати антологію

до прийнятного формату. Це досить зручно, бо інакше кожен клас необхідно було б програмувати руками без використання редактору онтологій.

BeanGenerator пов'язує кожну схему в онтології з класом Java або інтерфейсом. Сукупність всіх створюваних об'єктів, що реалізують інтерфейси *jade.content.Concept, jade.content.Predicate* або *jade.content.AgentAction* представляють собою онтологію. Крім того, "загальний" клас визначає словник усіх класів.

2.2 Принципи конструкції агента

У конструкції агента можна виділити три основні рівні (рис. 2.4):

Рівень комунікації. На цьому рівні виділяються компоненти і функціональні можливості підтримки комунікації між агентами. Класичний спосіб взаємодії агентів - обмін повідомленнями. Забезпечує асинхронний / синхронний обмін повідомленнями між агентами і обмін повідомленнями по протоколу (тобто підтримку контексту переговорів)

Рівень даних. На цьому рівні ізолюються всі властивості і дані агентів: пам'ять, досвід, значення онтологічних атрибутів і т.д. Рівень даних підтримує управління історією змін: скасування змін, збереження змін як окремої сутності, застосування змін.

Рівень поведінки (управління): рівень поведінки є активною складовою агента. На цьому рівні агента визначаються ролі агента. Роль агента - незалежна активна сутність, яка може:

- Ініціювати відсилання повідомлення
- Чекати (знаходиться в стані очікування) і реагувати на повідомлення або події зміни часу
- Отримувати інформацію, правила прийняття рішень і інші складні структури (цілі, стратегії) з рівня даних
- Змінювати інформацію в рівні даних
- Реагувати на зміну інформації в рівні даних

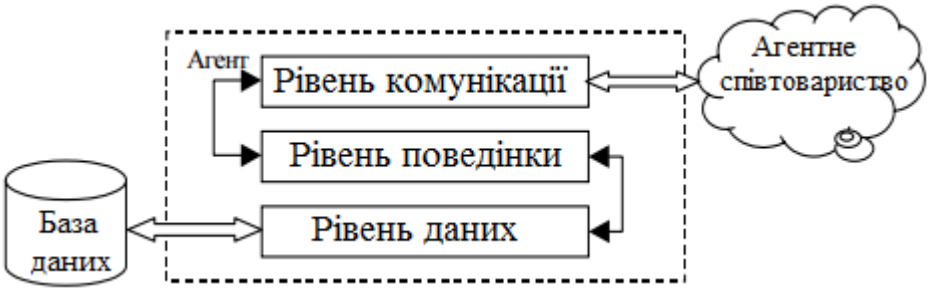

Рисунок 2.4– Рівні побудови

Кожен агент повинен виконувати конкретну дію сприаючись на конкретні дані. Конструкція агента (рис 2.5) вказую на процес який відбувається в середені кожного.

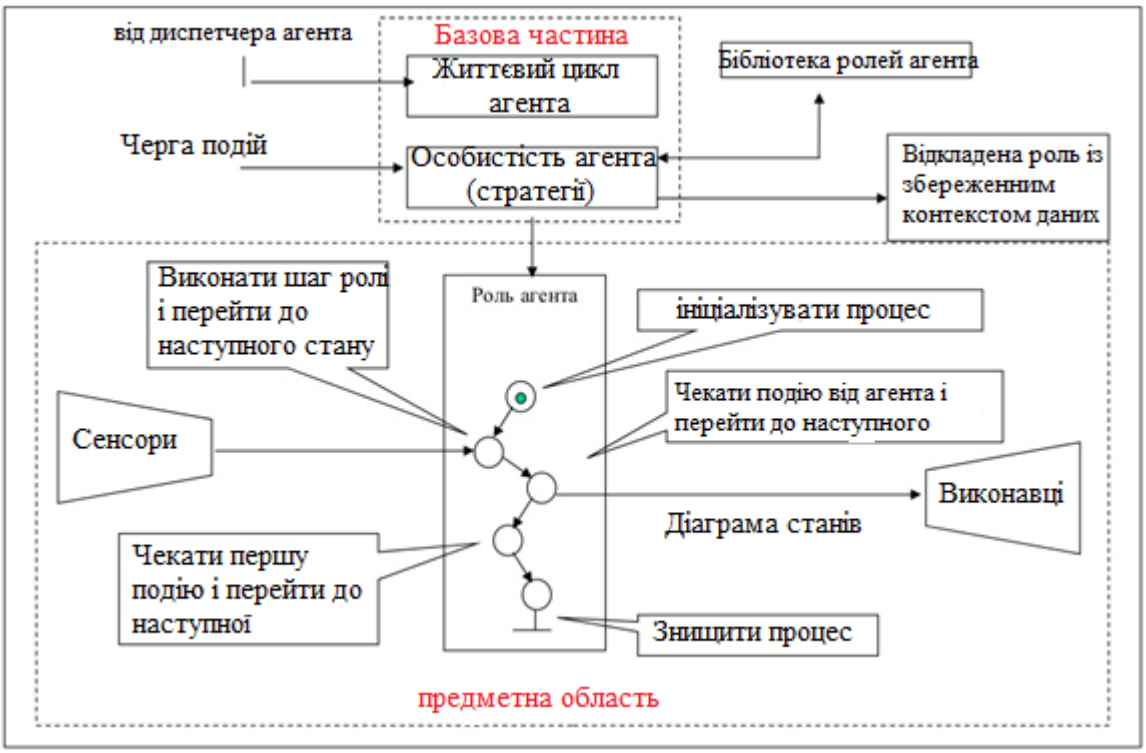

Рисунок 2.5 – Конструкція агента

Життєвий цикл агентів досить простий. Спочатку вони сприймають інформацію із зовнішнього світу. Потім її потрібно обробити, тобто запланувати якісь дії. Ну а дії вже потрібно виконати - віддавши відповідні команди в реальний світ (рис. 2.6)

Рисунок 2.6 – Основний цикл управління ресурсами агента

2.3 Архітектура мультиагентної системи

В архітектурі системи, що розробляється основну частину складає предметно-незалежне Мультиагентне ядро, в складі якого виділяються наступні базові компоненти (рис. 2.7):

1. Служба прямого доступу забезпечує прямий доступ візуальної частини до атрибутів агентів. При цьому, візуальна частина може спілкуватися з агентами за допомогою повідомлень, але використання даної підсистема по-перше, швидше, а по-друге, має ширші можливості для оперування агентами.
2. Служба повідомлень відповідає за передачу повідомлень між самим агентами, а так само між агентами і додатковими системами ядра.
3. Бібліотека класів агентів це частина бази знань, яка містить інформацію, про те, яких типів бувають агенти. Для підвищення гнучкості системи,

інформація в цьому довіднику може бути доповнена з зовнішніх розширень через інтерфейс.

4. Спільнота агентів - місце, де розміщуються агенти. Цей блок, крім життєдіяльності агентів, ще забезпечує функції по завантаження / запису агентів і їх властивостей і за оптимізацію роботи з ресурсами.
5. Онтологія - предметна база знань, яка містить конкретні знання про предмет (логістики), що подаються у вигляді семантичної мережі.

Рисунок 2.7 – архітектура ядра мультиагентної системи

Коли розробляється агент, він (а саме розробник агента) повинен визначити свою область технологічної діяльності та методи, які надаються в рамках цієї діяльності. Дані визначення повинні бути описані в єдиному для системи і єдино можливий вигляді, що представляє собою «онтологічне опис»: [Галузь знань] / [Область технологічної діяльності] / [Розділ технологічної діяльності] / [Дія] / [Параметр]

Щоб розробник міг єдино правильно описати діяльність агента, розділи в описі він вибирає з тих, які надаються батьківським класом. Онтологічні опису в агента задається під кожен метод, який агент надає.

Онтологічний батьківський клас повинен включати в себе максимально можливе опис «світу», щоб розробник мав можливість вибрати те опис до методу агенту, яке є єдино правильним. Також батьківський клас повинен надавати дані для кожного розділу «онтологічного опису», щоб не було

плутанини в порядку складання. Коли агент підключається до системи, він передає серверу свої онтологічні описи.

На діаграмі розглянемо основні дії агентів і серверного агента (далі сервер) пов'язані з онтологією. Перш за все, взаємодія агентів, як видно, починається з «онтологічного запиту» до сервера. Онтологічний запит являє собою «онтологічні опис», тільки спрямоване до сервера. Після, сервер по своїй базі знань звіряє наявні в ньому «онтологічні опису» з прийшли «онтологічним запитом» і, якщо знаходить збіги, видає агенту параметри і адреса агента / методу для запиту. Після, агент який направив запит надсилає запит до агента, з даними у вигляді структури містить параметри. Відповідальний агент передає дані також у вигляді спеціальної структури для відповіді.

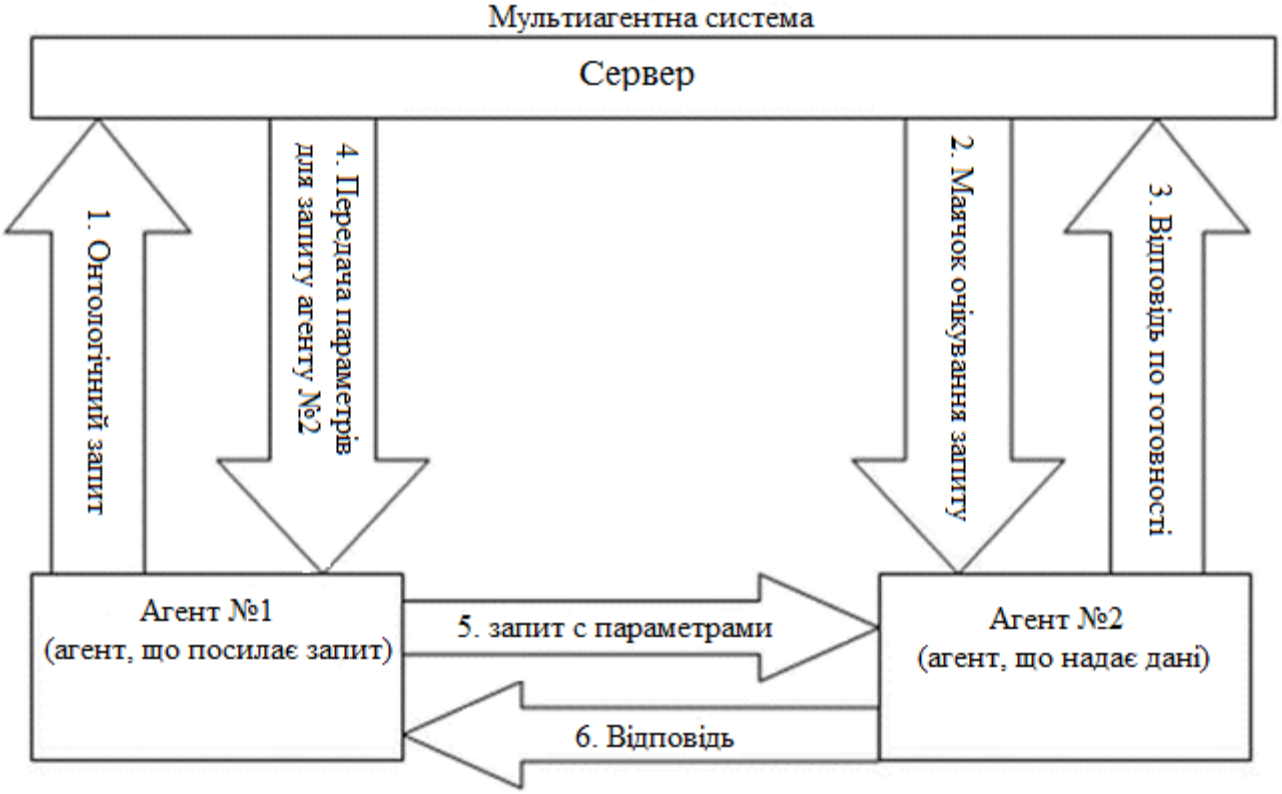

Рисунок 2.8 – Діаграма взаємодії агентів в мультиагентной системі

2.4 Архітектура програмного забезпечення

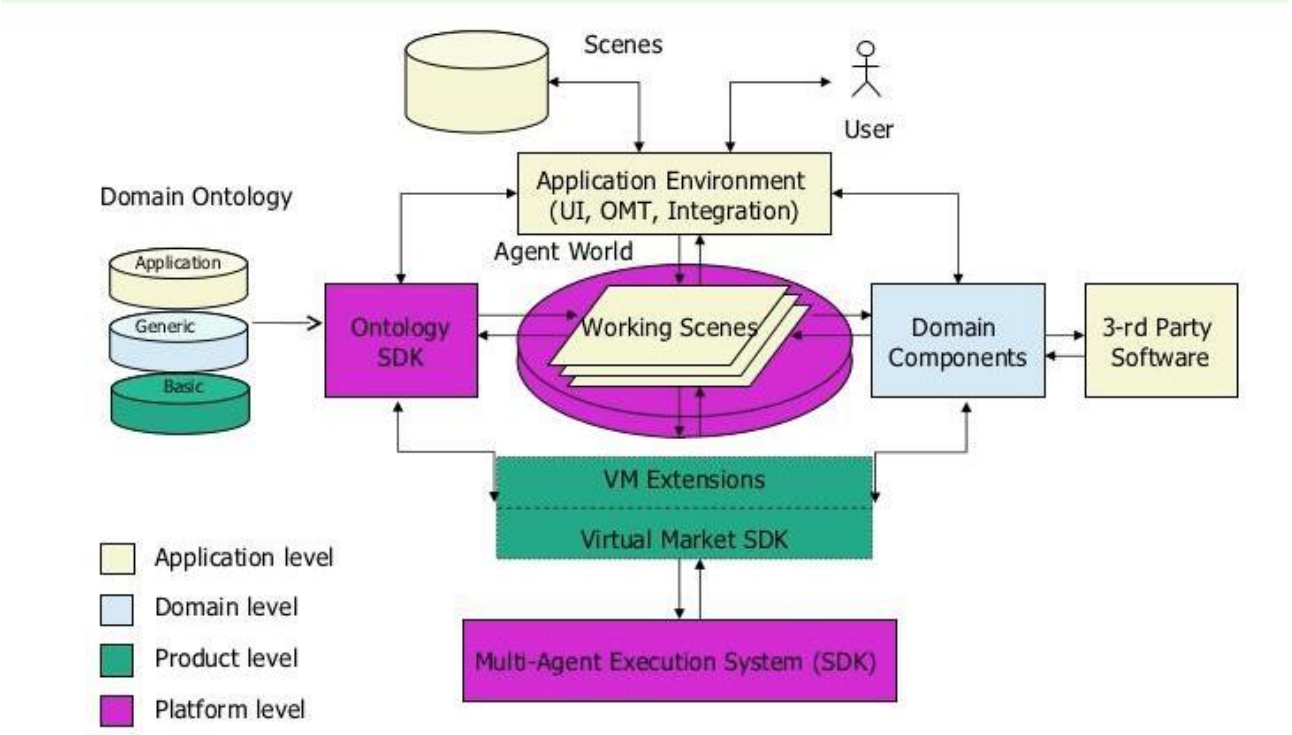

Рисунок 2.9 – структура модулю адаптивного планування

Виконуюча система (Run Time Multi-Agent Execution System) - підсистема, що забезпечує асинхронне виконання програм агентів при переході з одного стану в інший (диспетчер агентів) і передачу повідомлень між агентами, при якому агент отримує «квант» часу на обробку подій і далі повертає управління диспетчеру для просування наступних агентів, тобто агенти працюють як співпрограми. Частиною цієї системи є Інспектор агентів (Agent Inspector) і Журнал переговорів агентів (Agent Log), що показує всі повідомлення між ними.

Черга подій (Event Queue) - підсистема, що забезпечує накопичення подій, що приходять із зовнішнього світу, і їх послідовну обробку. Оскільки система є керованою подіями, при кожній події зберігається мітка часу його надходження і є можливість регулювати порядок надходження подій в систему

на обробку, коли така подія надходить після завершення обробки попереднього або, не чекаючи цього сигналу, в першу чергу вибираються пріоритетні події і т. д.

Світ агентів ПМ-мережі / Віртуальний ринок (Virtual World of RDN - Virtual Market) - місце роботи агентів ПМ-мережі, в якому запускаються і виконуються екземпляри класів агентів. Агенти можуть під управлінням виконуючої системи створюватися і знищуватися в світі, існувати в світі, приймати і передавати повідомлення, звертатися в сцену для читання інформації, записувати інформацію в сцену, підписуватися на події та отримувати повідомлення і т.д.

Сцена світу (Scene of the World) - основна структура даних, яка містить формалізовану модель ситуації в зовнішньому світі, яка може уточнюватися через онтологію. Сцена світу коригується подіями (в тому числі, користувачем), щоб забезпечити адекватність системи в сприйнятті ситуації в навколишньому світі. Сцена містить початкове опис ситуації, яка далі поступово трансформується в рішення проблеми з урахуванням вступників подій. В результаті, сцена містить новий план дій для користувача (водія вантажівки, майстри і робочого і т.д.).

Конструктор онтологій (Ontology Editor) - дозволяє вручну коригувати початкову сцену або вносити в неї зміни в ході роботи.

Онтології (Ontology) - структури даних, що представляють собою моделі знань предметної області, які використовуються для побудови моделей початкових ситуацій або їх коригування. Є базові онтології, які можуть доповнюватися спеціалізованими для предметної області поняттями і відносинами, і далі - спеціальними розширеннями для кожного окремого підприємства.

Бібліотеки планування (Basic Virtual Market & Domain-Specific Extensions) - містять базові та спеціалізовані компоненти, що забезпечують роботу класів агентів ПМ-мережі і їх переговори на віртуальному ринку (наприклад, виявлення конфліктів, визначення зон перекриття, розрахунок

зрушень і т.п.) , доступ до сцени, що містить формалізовану модель ситуації, а також еластичну обробку критеріїв, уподобань і обмежень агентів, розрахунки мікроекономіки і підтримки рахунків агентів і інші функції

База даних (Data base) - дозволяє зберігати вихідні і проміжні сцени, а також сцени з результатом вирішення проблеми.

Спеціалізовані компоненти і інтеграція з третіми системами (3rd Party & Integration Components) - компоненти, які дозволяють виконувати додаткові функції для предметної області.

На рис. 2.10 зображена діаграма класів, що використовуєтья у створеній мультиагентій системі

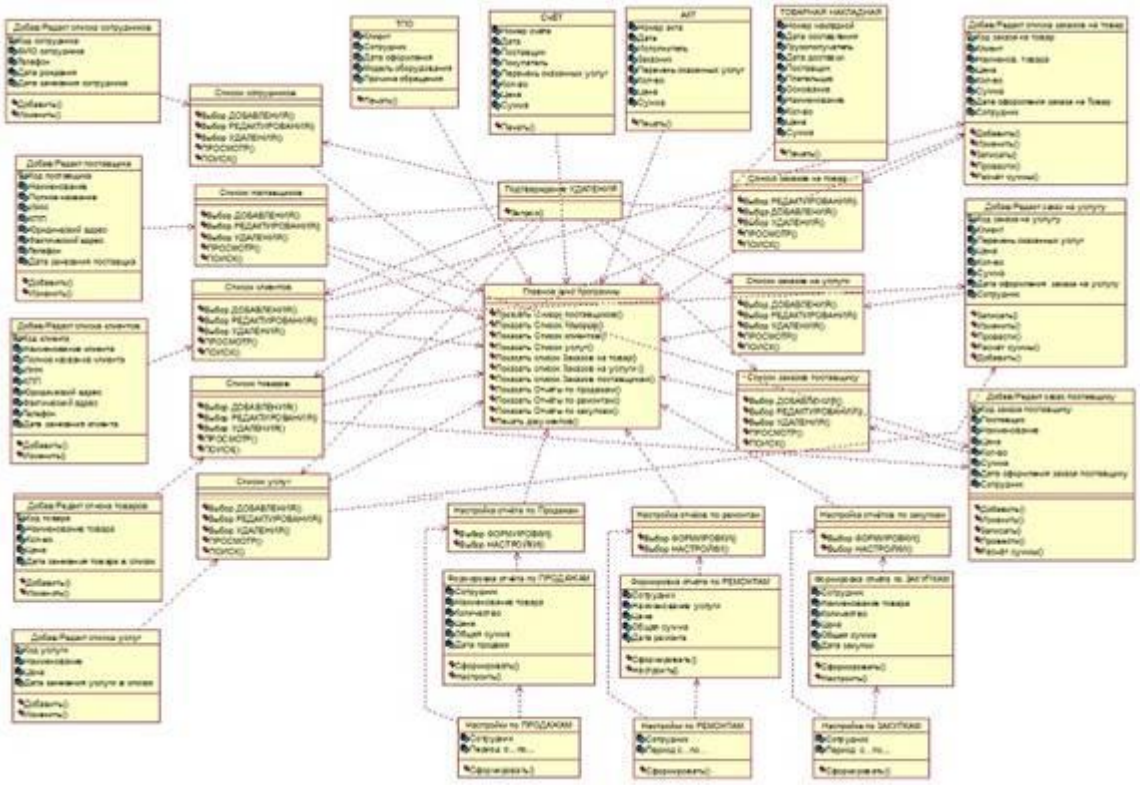

Рисунок 2.10 – UML-діаграма класів

Висновки до розділу 2

В даному розділі були наведені теоретичні дані щодо розробки мультиагентної системи, в компоненти якої входить серевер, три агенти

(організації агентів), а в якості бази знань було створено онтології. Для зручності керування кожним агентом онтологій створено декілька, а саме: окрема для кожного агента – для автономної та адаптивної роботи; та основна на сервері, що вміщує в себе абсолютно всі дані. У даній мультиагентій технологічній системі агенти можуть «спілкуватися» один з одним без втручання зі сторони технолога.

Також для подальшого тестування було розроблено експерементальний мобільний додаток. Поки що він покриває лише ті сценарії, які обумовлені для початкових тестувань самої мультиагентої системи та показ данних, як на звичайних додатках до кожного зі смарт-елементів. Було проаналізовано архітектуру та функціонал даного додатку, наведено UML-діаграму класів, спроектованих для написання програмного коду.

3 ОРГАНІЗАЦІЯ РОБОТИ СМАРТ-ЕЛЕМЕНТІВ МУЛЬТИАГЕНТНОЇ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»

Як було сказано раніше для прикладу було взято три смарт-девайси: пилосос (далі агент А), холодильник (далі агент В) та телевізор (далі агент С). Оскільки кожен з вибраних обьектів має декілька функцій, то можна сказати, що вони є не просто агентами, а організаціями агентів. Схематично таку мультиагентну систему зображено на рисунках 3.1 – 3.3.

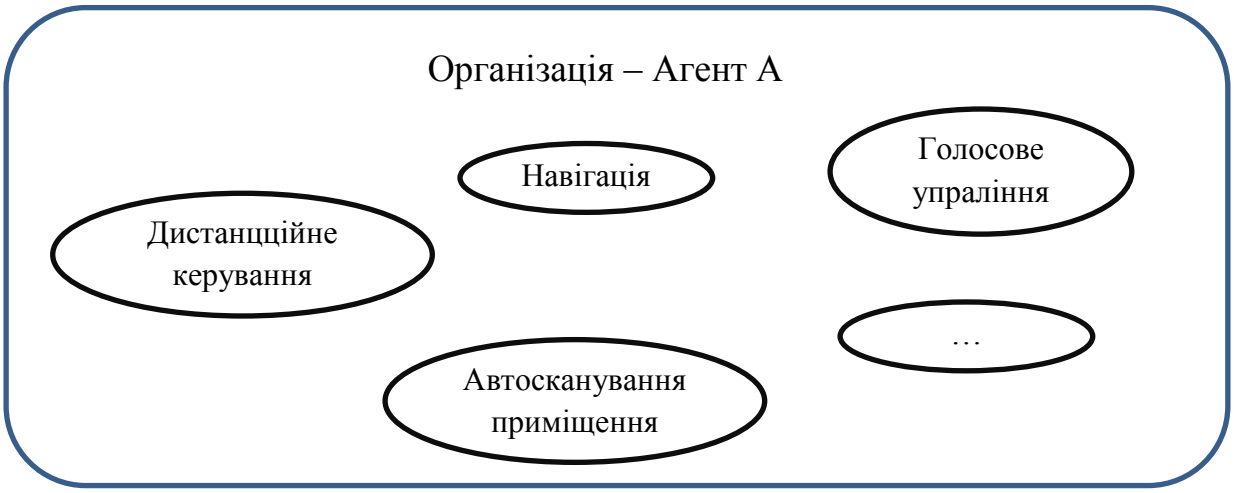

Рисунок 3.1 – Агент А, організація внутрішніх агентів-функцій

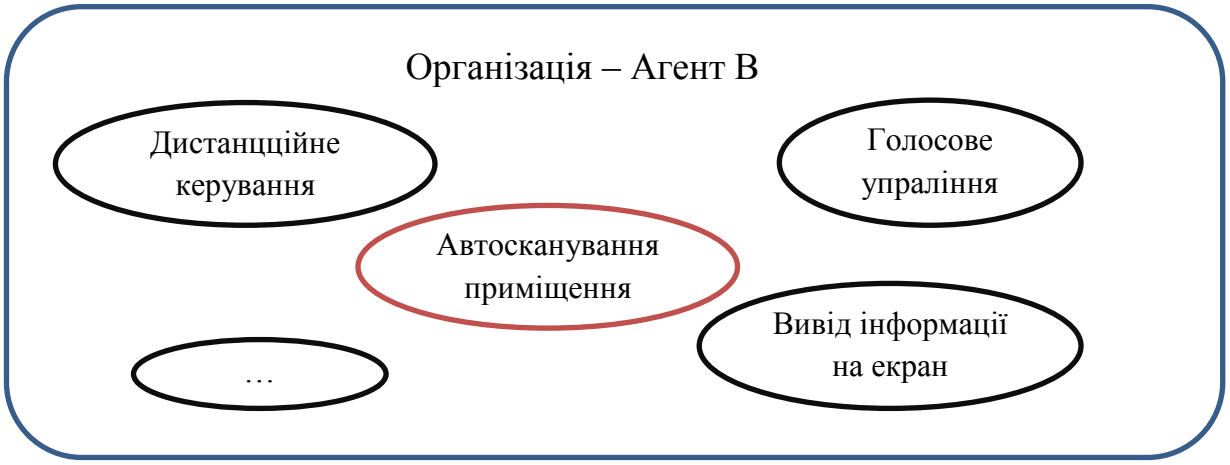

Рисунок 3.2 – Агент В, організація внутрішніх агентів-функцій

Рисунок 3.3 – Агент С, організація внутрішніх агентів-функцій

Для тестування впровадження онтології в якості бази знань в мультиагнетну систему смарт-роботів було розроблено декілька сценарії.

3.1 Сценарій №1

Зараз дуже важливою проблемою смарт-пилососів є те, що вони намагаються прибрати об’єкти, що неможливо прибрати з їх допомогою, адже ситуація стає тільки гіршою, і потрібно вже застосовувати вологе прибирання як для поверхні підлоги так і для самого пилососа. Для таких ситуацій було вирішено додати до функціоналу можливість навчатися. Отже припустимо, що пилососу прийшов запит на прибирання об’єкту незрозумілої консистенції. Послідовність дій буде наступною:

1. Отримання агентом А запиту на прибирання від іншого агента
2. Відповідь агента А про готовність роботи
3. Предача координат агенту А
4. Переміщення агента А до заданого місця.
5. Агент А сканує об’єкт, що необхідно прибрати
6. Порівнює з об’єктами із бази знань і отримує негативний результат
7. Надсилає користувачу запит на подальшу дію (запит містить фото та варіанти вибору подальших дій)
8. Користувач обирає необхідний варіант

9. Агент А починає прибирання при позитивній відповіді користувача, а при негативній – повертається на своє місце.
10. В той же час відповідь користувача і фото об’єкту заноситься до бази знань, для того щоб в подальшому при виникнені такої ситуації, не потрібно було витрачати час на непотрібні дії.

Блок схему такої поведінки агента А на роботу з новим і невідомим об’єктом зображено на рисуку 3.4.

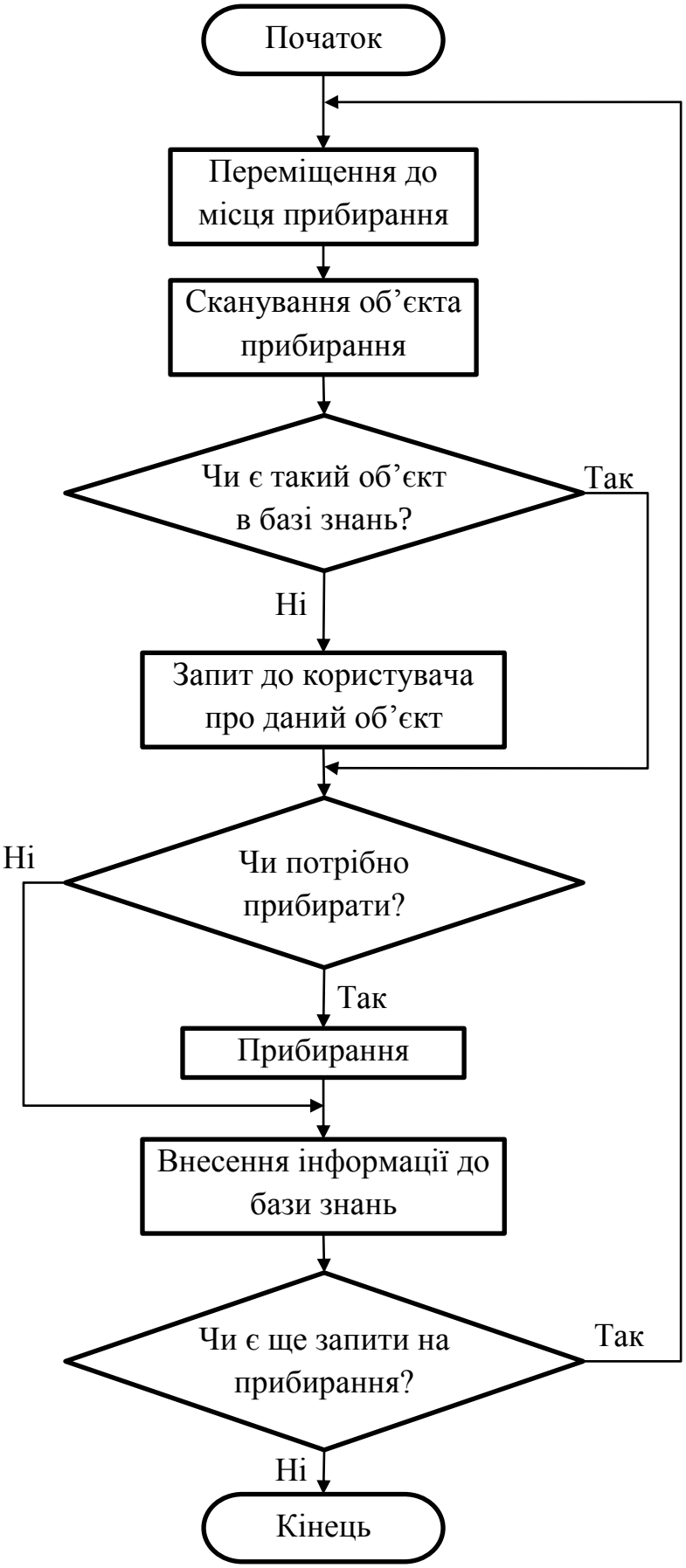

Рисунок 3.4 – блок-схема поведінки агента А з невідомим об’єктом

3.2 Сценарій №2

Розміщення смарт-елементів відрізняється, адже холодильник зазвичай розміщується на кухні, телевізор – у вітальні, а пилосос – у спеціально оснащеному місці. Оскільки кожен з агентів зберігає в собі карту приміщення припускаємо, що на кухні щось розсипалось і потрібно, щоб пилосос все прибрав. Послідовність дій буде наступною:

1. Агент В сканує приміщення
2. Агент В ідентифікує зміну поверхності підлоги
3. Агент В звертаючись до бази знань вирішує, що треба прибрати
4. Агент В знову звертаючись до бази ідентифікує об’єкт(агент А), який цим займається
5. Агент В подає запит на агента А
6. Агент В отримує відповідь про готовність роботи агента А
7. Агент В посилає необхідні параметри (координати) агенту А
8. Агент А направляється до місця призначення
9. Сценарій №1
10. Агент А виконує функцію прибирання
11. В архів подій заноситься процедура, що була виконана.

3.3 Сценарій №3

Всі три смарт-елементи мають функцію голосового управління, тобто вони виконують ту чи іншу роботу після озвучування певної команди. Але якщо необхідно надати таку команду, то користувач повинен бути поруч з агентом, а це не завжди можливо. Але маючи на сервері всі необхідні словники, що входять до онтології, це можна зробити через інший смарт-елемент.

Припустимо, що знаходячись на кухні, необхідно вимкнути телевізор. Потрібна команда озвучується холодильнику, той в свою чергу її прослуховує, після чого звертається до бази знань. Далі якщо така команда присутня у будь-

якому із словників, то шукається відповідність агента, що має виконати цю команду і посилається із сервера відповідна команда.

У випадку, коли необхідно знизити темепературу в холодильнику а поряд занходиться пилосос, то процес точно такий же як і у попередньо описаному варіанті.

3.4 Сценарій №4

Агенти B та C мають екран і функцію відтворення фотографій, відео чи просто відображати будь-яку сторінку у веб-браузері. Але у випадку, коли необхідно переміщуватися з вітальні до кухні і навпаки, було б комфортно не пропускати можливо щось дуже важливе. Тож наступний сценарій це включення тієї ж самої сторінки веб-браузера, що вже була завантажена на іншому девайсі.

Цей сценарій дуже схожий на попередній, але додатково включає в себе передачу від одного до іншого агента необхідних параметрів, таких як адресу веб-сторінки, якщо це було відтворення відео, то параметри перегляду відео (якість, швидкість, гучність, час, на якому було зупинено, тощо)

Для порівняння вибраної стратегії організування роботи групи інтелектуальних роботів за допомогою впровадження мульиагентної системи було використано чотири моделі керування смарт-елементами. Модель 1 – ручне керування. Модель 2 – дистанційне за допомогою пультів. Модель 3 – дистанційне за допомогою мобільних додатків. Модель 4 – адаптивне керування. Результат можна побачити на рисунку 3.5.

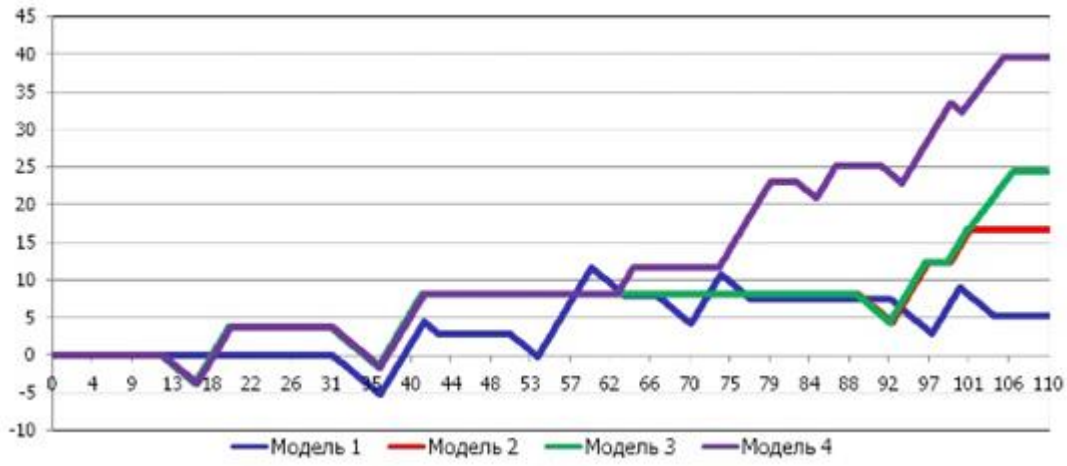

Рисунок 3.5– графік оцінки економії часу використання смарт-роботів

Висновки до розділу 3

Експерементальна частина, тобто тестування створеної системи показало вагоме право на існування, а також значну перевагу перед іншими. Дана система моделює адаптивну, дистанційну та практично самостійну роботу смарт-елементів «розумного будинку».

Сценарії, що були розроблені надали не тільки спосіб вирішення деяких проблем, а також показали шляхи майбутнього розвитку системи та застосування її на масштабніших прикладах.

На графіку оцінки економії часу використання смарт-роботів видно, що за допомогою моделі 4 економія часу виросла на 40-60%, а в подальшому може складати ще більший відсоток. Це доводить значну перевагу моделі адаптивного керування перед іншими.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

В межах підпункту послідовно проаналізовано та подано у вигляді таблиць:

- зміст ідеї;
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- чим відрізняється від існуючих аналогів та замінників;

Перші три пункти подано у вигляді таблиці (табл. 4.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Ідея полягає у тому, щоб створити систему, завдяки якій було б полегшено керування елементами «розумного будинку», а також з’явилась би можливість автономної та адаптивної роботи цих елементів.	1. Дистанційного керування	Полегшення керування, контроль даних у будь-який час, а також вирішення нових або ж спірних задач для смарт-елементів на відстані
	2. Налаштування самостійної та адаптивної роботи пристроїв	Користувачу не постійно контролювати роботу, смарт –пристрої самостійно вирішують вже знайомі проблеми спілкуючись між собою.

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї
- визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;
- проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні).

Таблиця 4.2. –Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	*Техніко-економічні характеристики ідеї*	*(потенційні) товари/концепції конкурентів*				*W (слабка сторо-на)*	*N (нейтральна сторона)*	*S (сильна сторона)*
		Мій проект	*Конкурент1*	*Конкурент2*	*Конкурент3*			
1.	Форма виконання	Програма	Веб додаток	Веб додаток	Програма			+
2.	Собівартість	Низька	Низька	Низька	Висока		+	
3.	Наявність адміністратора	Не треба, дистанційно	Треба	Треба	Треба			+
4.	Наявність інтернету	Необхідно	Необхідно	Необхідно	Необхідно	+		
5.	Кросплатформенність	Так	Так	Ні	Ні			+
6.	Складність використання/автономність	Так	Ні	Ні	Ні			+

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу проведено аудит технології, за допомогою якої реалізувано ідею проекту.

Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл. 4.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту?
- чи існують такі технології, чи їх потрібно розробити/доробити?
- чи доступні такі технології авторам проекту?

Таблиця 4.3. – Технологічна здійсненність ідеї проекту

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
1.	Створення програмного забезпечення	AnyLgic	Наявна	Безкоштовна, доступна
		New Java API	Наявна	Безкоштовна, доступна
		NEW C++ API	Наявна	Безкоштовна, доступна
		JADE	Наявна	Безкоштовна, доступна
Обрана технологія реалізації ідеї проекту: для створення мобільного додатку обрана технологія JADE, яка є безкоштовною та якою володіють розробники.				

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із

урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн/ум.од	20000 грн./ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	R = (3000000 * 100) / (1000000 * 12) = 25%

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. За умови, що останній є вищим, можливо, має сенс вкласти кошти в інший проект.

За результатами аналізу таблиці зроблено висновок щодо того, чи є ринок привабливим для входження за попереднім оцінюванням. Так, є.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5. –Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
1.	Необхідно програмне забезпечення	Потенційними цільовими групами є	Цільова група використовує робототехнвку	Рішення має бути кросплатформним, необхідно, щоб

	для комфортного керування адаптивних смарт-елементів	власники смарт-техніки, компанії, які використовують роботів	для прискорення і полегшення повсякденного життя або ж виробництва.	користувач абстрагувався від налаштування кожного з роботів, а також міг впливати на їх роботу.

Після визначення потенційних груп клієнтів проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. №4.6-4.7). Фактори в таблиці подані в порядку зменшення значущості.

Таблиця4. 4. – Фактори загроз

№ п/п	*Фактор*	*Зміст загрози*	*Можлива реакція компанії*
1.	Конкуренція	Вихід на ринок іншої великої компанії швидше за нашу	1) Вихід з ринку 2) Запропонувати іншій компанії обьєднатися 3) Передбачити додаткові переваги власного ПЗ для того, щоб повідомити про них саме після виходу міжнародної компанії на ринок
2.	Зміна потреб користувачів	Користувачам необхідне програмне забезпечення з іншим функціоналом	1) Передбачити можливість додавання нового функціоналу до створюваного ПЗ

Таблиця 4.7. – Фактори можливостей

№ п/п	*Фактор*	*Зміст можливості*	*Можлива реакція компанії*
1	Зростання можливостей потенційних покупців	Зростанням держфінансування досліджень у галузі ЦОС	Запропонувати свої послуги спеціалізованим компаніям, включити як акцію для відповідної смарт-техніки
2	Зниження довіри до конкурента 1	У ПЗ конкурента №1 нещодавно була знайдена помилка, завдяки якій дані	При виході на ринок звертати увагу покупців на безпеку нашого ПЗ та

		досліджень усіх клієнтів стали доступні в інтернеті для всіх користувачів	авторитетність компанії

Далі проведено аналіз пропозиції: визначено загальні риси конкуренції на ринку (табл. 4.8).

Таблиця 4.5. – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Вказати тип конкуренції - досконала	Існує 3 компанії-конкурентки на ринку	Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами)
2. За рівнем конкурентної боротьби - міжнародний	Всі компанії з інших країн	Додати можливість вибору мови ПЗ, щоб легше було у майбутньому вийти на міжнародний ринок
3. За галузевою ознакою - внутрішньогалузева	Конкуренти мають ПЗ, яке використовується лише всередині даної галузі	Створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях
4. Конкуренція за видами товарів: - товарно-видова	Види товарів є однаковими, а саме – програмне забезпечення (мобільний додаток)	Створити ПЗ, враховуючи недоліки конкурентів
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення ПЗ, щоб собівартість була нижчою	Використання менш дорогих технологій для розробки, ніж використовують конкуренти
6. За інтенсивністю - марочна	Бренди присутні	-

Після аналізу конкуренції проведено більш детальний аналіз умов конкуренції в галузі (табл. 4.9).

Таблиця 4.6. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	*Прямі конкуренти в галузі*	*Потенційні конкуренти*	*Постачальники*	*Клієнти*	*Товари-замінники*
	Навести перелік прямих конкурентів	*Визначити бар'єри входження в ринок*	*Визначити фактори сили постачальників*	*Визначити фактори сили споживачів*	*Фактори загроз з боку замінників*
Висновки:	Існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 3, так як його рішення також представлене у вигляді ПЗ.	Так, можливості для входу на ринок є, бо наше рішення покращує та пришвидшує роботу спеціаліста.	Постачальни-ки відсутні.	Важливим для користувача є кросплатформе-ність ПЗ та якість його роботи.	Товари-замінники можуть використати більш дешеву технологію створення ПЗ та зменшити собівартість товару.

За результатами аналізу таблиці зроблено висновок щодо принципової можливості роботи на ринку з огляду на конкурентну ситуацію. Також зроблено висновок щодо характеристик (сильних сторін), які повинен мати проект, щоб бути конкурентоспроможним на ринку. Другий висновок враховується при формулюванні переліку факторів конкурентоспроможності у п. 3.6.

На основі аналізу конкуренції, проведеного в п. 3.5 (табл. 4.9), а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл.№4.6-4.7) визначено та обґрунтовано перелік факторів конкурентоспроможності. Аналіз оформлюється за табл. 4.10

Таблиця 4.7. – Обґрунтування факторів конкурентоспроможності

№ п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
1.	Виконання ПЗ у вигляді мобільного додатку	Це рішення дозволяє використовувати ПЗ з будь-якого мобільного пристрою, яке підключене до інтернету та для з'єднані зі смарт-елементами
2.	Простота інтерфейсу користувача	Інтерфейс користувача зроблений таким чином, що користувачу необхідно лише провести з'єднання з роботами, обрати за необхідністю роботу, що має виконати смарт-елемент та у спірних або невідомих для робота задач повідомити план дій, просто вибравши правильний пункт із запропонованих самим роботом.

За визначеними факторами конкурентоспроможності (табл. 4.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.8 – Порівняльний аналіз сильних та слабких сторін «назва проекту»

№ п/п	*Фактор конкурентоспроможності*	*Бали 1-20*	*Рейтинг товарів-конкурентів у порівнянні з нашим підприємством*						
			–3	*–2*	*–1*	*0*	*+1*	*+2*	*+3*
1	Виконання ПЗ у вигляді мобільного додатку	15			+				
2	Простота інтерфейсу користувача	20	+						

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.11).

Перелік ринкових загроз та ринкових можливостей складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на

ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 4.9 – SWOT- аналіз стартап-проекту

Сильні сторони: простий інтерфейс користувача, виконання ПЗ у вигляді мобільного додатку	Слабкі сторони: необхідний доступ до інтернету для роботи з ПЗ
Можливості: у конкурента 1 виявлена проблема із безпекою ПЗ, додаткове держфінансування для досліджень на підприємствах, які є потенційними покупцями	Загрози: конкуренція, зміна потреб користувачів

На основі SWOT-аналізу розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. табл. 9, аналіз потенційних конкурентів).

Визначені альтернативи проаналізовано з точки зору строків та ймовірності отримання ресурсів (табл. 4.13).

Таблиця 4.10 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
1.	Створення мобільного додатку за технологією JADE, з використанням онтології в основі, яка дозволить забезпечити автономність і адаптивність роботи смарт-роботів	80%	6 місяців
2.	Створення мобільного додатку за допомогою технології JADE, яка не має в основі семантичний зв’язок, що є більш швидким рішенням	60%	6 місяці

Обираємо альтернативу 1

З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими.

Враховуючи, що впровадження онтології є головним завданням поліпшення спільної роботи смар-елементів розумного будинку, а час виконання однаковий, то обираємо перший варіант

4.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.11 – Вибір цільових груп потенційних споживачів

№ п/п	*Опис профілю цільової групи потенційних клієнтів*	*Готовність споживачів сприйняти продукт*	*Орієнтовний попит в межах цільової групи (сегменту)*	*Інтенсивність конкуренції в сегменті*	*Простота входу у сегмент*
1.	Користувачі смарт-роботів	Час виконання не є критичним у даному випадку, а однією із найголовніших переваг є якість та можливість навчати роботів	Передача хатніх обовязків роботам-помічникам вже давно частина життя сучасних користувачів	Існує 3 конкуренти, які надають схожі, але менш швидкі та якісні рішення.	У сегмент увійти непросто, бо велика конкуренція, також відсутня дуже впливова перевага ПЗ
2.	Виробництва, що використовують роботів	Пришвидшення часу виконання додає їм можливості збільшити прибуток	Виробництва використовують роботів для підвищення швидкості та часу, а також їх заміна не		Маючи перевагу у зручності інтерфесу користувача, що полегшує

			потребує часу на налаштування		керування роботи, вийти на ринок нескладно
3.	Профільні компанії	Пришвидшення часу виконання та адаптивне налаштування допомагає не витрачати багато робочого часу, до того ж в цьому році з держбюджету виділені додаткові кошти на цей напрям досліджень			Маючи перевагу у тому, що рішення є зручним для користування і крос-платформним, вийти на ринок не є складно
Які цільові групи обрано користувачів смарт-роботів та профільні компанії					

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку:

- якщо компанія зосереджується на одному сегменті – вона обирає стратегію концентрованого маркетингу;
- якщо працює із кількома сегментами, розробляючи для них окремо програми ринкового впливу – вона використовує стратегію диференційованого маркетингу;
- якщо компанія працює із всім ринком, пропонуючи стандартизовану програму (включно із характеристиками товару/послуги) – вона використовує масовий маркетинг.

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (табл. 4.15).

Таблиця 4.12 – Визначення базової стратегії розвитку

№ п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку **
1.	Створення мобільного додатку на Tizen, з використанням онтології в основі, яка дозволить адаптивно та віддалено навчати смарт-роботів	Ринкове позіціювання	Простота інтерфейсу, пришвидшення роботи, навчаємість, крос-платформність	Диференціації

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16).

Таблиця 4.13 – Визначення базової стратегії конкурентної поведінки

№ п/п	*Чи є проект «першопрохідцем» на ринку?*	*Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?*	*Чи буде компанія копіювати основні характеристики товару конкурента, і які?*	*Стратегія конкурентної поведінки**
1.	Ні	Так	Буде, а саме: основною задачею ПЗ є дистанційне керування смарь-роботами (конкуренти 1, 2, 3), крос-платформність (як у конкурента 1)	Зайняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.15) та стратегії конкурентної поведінки (табл. 4.16) розробляється стратегія позиціонування (табл. 4.17). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.14 – Визначення стратегії позиціонування

№ п/п	*Вимоги до товару цільової аудиторії*	*Базова стратегія розвитку*	*Ключові конкурентоспроможні позиції власного стартап-проекту*	*Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)*
1.	Простота інтерфейсу, дистанційність керування, навчаємість, крос-платформність	Диференціації	Простота користувацького інтерфейсу, навчаємість, дистанційність керування що дозволяє пришвидшити, та спростити роботу користувачів та покращити роботу смарт-роботів	Швидкість, якість, дистанційність

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії на ринку.

4.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування *маркетингової концепції товару*, який отримає споживач. Для цього у табл. 4.18 підсумовано результати попереднього аналізу конкурентоспроможності товару.

Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 4.15 – Визначення ключових переваг концепції потенційного товару

№ п/п	*Потреба*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити*
1.	Крос-	Використання	Рішення є крос-платформним

	платформність	будь-якої операційної системи	
2.	Спрощення інтерфейсу користувача	Пришвидшення роботи з ПЗ	Користувачам не потрібно замислюватись над тим, як налаштувати нового робота, всю інформацію він отримує від інших смарт-елементів «розумного будинку». Також швидше починається робота та навчаємість смарт-роботів

Розроблена трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.19).

1-й рівень - При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень - Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень - Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін)

Таблиця 4.16 – Опис трьох рівнів моделі товару

Рівні товару	*Сутність та складові*
І. Товар за задумом	Об’єкт допомогає користувачам смарт-роботів відмовитись від власноручного налаштування, покращити навчаємість їх помічників, а також постійно бути вкурсі всіх даних. Вони лише з’єднуються з кожним смарт-елементом, та вібирають роботу, що повинен виконати робот. Для того, щоб постійно покращувати роботу роботів, користувач має змогу навчати смарт-роботів тоді коли, робот запитує правильність вибору варіанту роботи з тим чи іншим об’єктом..

II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Крос-платформне 2. Наявність інтернету необхідне 3. Дистанційність керування 4. Простота у використанні 5. можливість навчання	-	-
	Якість: згідно до стандарту ISO 4444 буде проведено тестування		
	Маркування відсутнє.		
	Компанія-виробник. «Татлекс», назва товару – «Smart pith»		
III. Товар із підкріпленням	1-місячна пробна безкоштовна версія		
	Постійна підтримка для користувачів		
За рахунок чого потенційний товар буде захищено від копіювання: ноу-хау.			

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та третьому рівнях товару.

Наступним кроком визначено цінові межі, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20). Аналіз проводився експертним методом.

Таблиця 4.17 – Визначення меж встановлення ціни

№ п/п	*Рівень цін на товари-замінники*	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни на товар/послугу*
1.	25000	30000	200000	20000

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);

- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 4.18 – Формування системи збуту

№ п/п	*Специфіка закупівельної поведінки цільових клієнтів*	*Функції збуту, які має виконувати постачальник товару*	*Глибина каналу збуту*	*Оптимальна система збуту*
1.	Купують ПЗ та роблять щорічні внески для подовження ліцензії	Продаж	0(напряму), 1(через посередника)	Власна та через посередників

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.19 – Концепція маркетингових комунікацій

№ п/п	*Специфіка поведінки цільових клієнтів*	*Канали комунікацій, якими користуються цільові клієнти*	*Ключові позиції, обрані для позиціонування*	*Завдання рекламного повідомлення*	*Концепція рекламного звернення*
1.	Купівля ПЗ через інтернет, робота з ПЗ на смартфонах з різними ОС	Інтернет	Швидкодія, простота використання, крос-платформ-ність	Показати переваги ПЗ, у тому числі і перед конкурентами	Демо-ролик із використання

Результатом пункту 5 має стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

Висновки до розділу 4

Згідно до проведених досліджень:

- існує можливість ринкової комерціалізації проекту;
- існують перспективи впровадження з огляду на потенційні групи клієнтів, бар’єри входження не є високими, проект має дві значні переваги перед конкурентами;
- необхідно реалізувати мобільний додаток із використанням технології JADE;
- подальша імплементація є доцільною.

ВИСНОВКИ

Основним завданням даної дисертації було дослідження використання онлотогії в якості бази знань в інформаційних системах. За приклад такої системи було взято мультиагентну систему.

В ході розробки дисертації дану задачу було розкрито повною мірою на прикладі проектування та реалізації мультиагентої системи керування смарт-елементами «Розумного будинку» та проведено сценарійні експерименти над ними. Було встановлено, що мультиагентний підхід з використанням онтологій до обраної задачі значно перевершує традиційні методи в ефективності, адже дозволяє врахувати значно більшу кількість факторів, збільшує адаптивність такої системи та зменшує час роботи з нею користувача.

Експерементальна частина показала вагоме право на існування такої системи, а також значну перевагу перед іншими. Дана система моделює адаптивну, дистанційну та практично самостійну роботу інтелектуальних об’єктів «Розумного будинку». Сценарії, що були розроблені надали не тільки спосіб вирішення деяких проблем, а також показали шляхи майбутнього розвитку системи та застосування її на масштабніших прикладах.

Порівняння з іншими видами керування видно на графіку оцінки економії часу використання смарт-роботів і показало, що за допомогою моделі 4 економія часу виросла на 40-60%, а в подальшому може складати ще більший відсоток. Це доводить значну перевагу моделі адаптивного керування перед іншими.

Перед реалізацією практичної частини було проаналізовано різні парадигми реалізації мультиагентних систем та онтологій. Для детального ознайомлення з предметною областю дослідження було проведено огляд особливостей визначення та структури онтологій у контексті комп’ютерних наук.

Було проаналізовано мову OWL для опису онтологій семантичної павутини, охарактеризовано компоненти, з яких зазвичай складаються

онтології, описані за допомогою OWL, їхні особливості та суть застосування. Розглянуто основні різновиди мови OWL (OWL Lite , OWL DL, OWL Full), проаналізовано можливості кодного з них.

Було проаналізовано поняття семантичного різонера, основну концепцію його побудови. Було проаналізовано принципи роботи таких алгоритмів: прямого виведення,зворотного виведення, побудови семантичних таблиць, побудови гіпертаблиць. Було здійснено огляд існуючих різонерів, особливостей їх реалізації.

Практична частина довела користь та перевагу застосування мультиагентної системи з використанням онтологій. Звичайно, одним із перших кроків буде робота над вирішенням питання безпечного спілкування агентів. Це дає поштовх для подальших розробок. Адже система будувалася на прикладі лише трьох інтелектуальних роботів, а система «розумного дому» розвинена настільки, що вміщує в себе більше 10 різних смарт-елементів. Також неможна забувати про те, що «розумний будинок» - це тільки один елемент такого великого поняття як інтернет-речей. Туди також входять: медицина, торгівля, промисловість, сільське господарство, екологія, траспорт, безпека тощо. І для всього цього переліку областей є доцільним використання саме таких систем для покращення та оптимізації їх роботи.

Ще одним важливим висновком є те, що голосове управління передбачає, конкретні фрази і на кокретній мові, що закладені у кожний об’єкт свої. Це не завжди є зручним, тож доцільним буде вподальшому розвивати систему у напрямі застосування актів мови. Це однозначно полегшить з перших же кроків користувачу роботу з такою системою, а система стане більш гнучкою та ще біль адаптивною.

.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, M. Sintek, and K. DFKI. Toward a technology for organizational memories. Intelligent Systems and Their Applications, IEEE 13(3):40–48, 1998.
2. Baumgartner P. Hyper Tableaux / Baumgartner P., Furbach U. - Niemela Universitat Koblenz Institut für Informatik Rheinau 1, 56075 Koblenz, Germany, 2013. – 18 c.
3. Bock J. Benchmarking OWL Reasoners / Bock J., Haase P., Ji Q., Volz R. - ARea2008 - Workshop on Advancing Reasoning on the Web: Scalability and Commonsense, 2008. – 104 c
4. British Computer Society's Specialist Group on Expert Systems. Internetbased Decision Support for Evidence-based Medicine. Expert Systems '98, 1998.
5. Cornet R. Non-standard reasoning services for the debugging of description logic terminologies. / Cornet R., Schlobach S. - Gottlob, G., Walsh, T., eds.: IJCAI, Morgan Kaufmann , 2003. - 529 c.
6. Deshendran Moodley Ontology driven multi-agent system: an architecture for sensor web applications, University of KwaZulu-Natal, Durban, South Africa, December 2009
7. Dignum, V., Dignum, F. Modelling agent societies: co-ordination frameworks and institutions. In: Brazdil, P., Jorge, A. (eds.) Progress in Artificial Intelligence: Knowledge Extraction, Multi-agent Systems, Logic Programming, and Constraint Solving, LNAI 2258, Springer, pp. 191-204, 2001.
8. Eivazzadeh S, Anderberg P, Larsson TC, Fricker SA, Berglund J Evaluating Health Information Systems Using Ontologies, JMIR Med Inform, 2016
9. Emmanuel Solidakis, Nikolaos Konstantinou, Emily-Sirin Pashou, Anthi Papa-konstantinou and Nikolas Mitrou. A Decentralized Multi-Agent Ontology-Based System for Information Retrieval. 2005.
10. F. Bellifemine, G. Caire, T. Trucco, and G. Rimassa. JADE Programmer's Guide. Italy: CSELT SpA, 2:120–122, 2000.

11. Fabio Bellifemine, Giovanni Caire, and Dominic Greenwood. Developing Multi-Agent Systems with JADE. John Wiley & Sons Ltd., The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2007.
12. Fensel, D. Ontologies and Electronic Commerce. IEEE Intelligent Systems, January/February, 8, 2001.
13. Gardiner Tom. Automated Benchmarking of Description Logic Reasoners / Gardiner Tom, Ian Horrocks, Dmitry Tsarkov. - Description Logics Workshop 2006. – 8 c.
14. Hähnle, R. Tableaux and Related Methods. Handbook of Automated Reasoning / Hähnle, R. – Volume I. Elsevier science, 2001. – 277 c.
15. Hayes, P. OWL Web Ontology Language Semantics and Abstract Syntax / Hayes, P., Horrocks, I., Patel-Schneider, P.F. - W3C Recommendation, 10 February 2004. – 136 c.
16. Hayes-Roth F. - Building Expert Systems / Hayes-Roth F., Waterman D., Lenat D.- Addison-Wesley, 1983. – 254 c.
17. J. Becker, M. Kugeler, and M. Rosemann. Process Management: A Guide for the Design of Business Processes. Springer, 2003.
18. Kaczor K. Overview of Expert System Shells / Kaczor K., Szymon B., Grzegorz J. - Krakow, Poland: Institute of Automatics: AGH University of Science and Technology, Poland, 5 December 2010. – 334 c
19. M. Wooldridge and N.R. Jennings. Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, 10(2):115–152, 1995.
20. M. Wooldridge. An Introduction to MultiAgent Systems, 1-14, 2002.
21. Malucelli, A., Cardoso, H. L. Oliveira, E. Enriching a MAS Environment with Institutional Services. In: Weyns, D., Parunak, V., Michel, F. (eds.) Environments for Multiagent Systems II (E4MAS), LNCS 3830, Springer-Verlag, Berlin, 2005. Bibliography 153
22. Malucelli, A., Palzer, D., Oliveira, E. Combining Ontologies and Agents to Help in Solving the Heterogeneity Problem in E-Commerce Negotiations. In

Proceedings of the International Workshop on Data Engineering Issues in E-Commerce (DEEC 2005), IEEE Computer Society, Tokyo, Japan, pp. 26-35, 2005.

23. Naumenko T.O. The analysis of fields of using ontologies in the construction of information systems / 19-th International conference on System Analysis and Information Technology SAIT 2017, May 22 – 25, 2017 Institute for Applied System Analysis at the Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine (2017.05.15), p.234-235
24. Schalkoff R. Intelligent Systems: Principles, Paradigms and Pragmatics / Schalkoff R. - Jones & Bartlett Learning, 2009. – 762 с
25. Shoham Y., Leyton-Brown K. Multiagents systems: Algorithmic, Game-Theoretic and Logical Foundations, London: Cambridge University Press, 2009. – P. 14–37.
26. Soares A. Building Ontologies for Information Systems: What we have, what we need / A. Soares, F. Fonseca, Pennsylvania State University, 2010, 312 p.
27. Todd N. R. Religious networking organizations and social justice: An ethnographic case study //American journal of community psychology. 2012. Т. 50. №. 1-2. С. 229-245.
28. Tsvetkov V.Ya. Worldview Model as the Result of Education // World Applied Sciences Journal. 2014. № 31 (2). p. 211-215.
29. Tsvetkov V.Ya. Incremental Solution of the Second Kind Problem on the Example of Living System, Biosciences biotechnology research Asia, November 2014. Vol. 11(Spl. Edn.), pp. 177-180. doi: http://dx.doi.org/10.13005/bbra/1458.
30. Uschold, M. Barriers to Effective Agent Communication, Position Statement, Workshop on Ontologies in Agent Systems, Montreal, Canada, 2001.
31. Wooldridge M. Introduction to MultiAgent Systems / Wooldridge M. John Wiley and Sons, 2002. – 214 с
32. Амелин К.С., Баклановский М.В., Граничин О.Н. и др. Адаптивная мультиагентная операционная система реального времени // Стохастическая оптимизация в информатике. 2013. Т. 9. Вып. 1. С. 3-16.

33. Безгубова Ю.О. Модели программных агентов в задачах информационного поиска // Славянский форум. 2015. № 2(8). С. 41-49.

34. Безгубова Ю.О. Мультиагентное управление распределенными информационными потоками // Образовательные ресурсы и технологии. 2015. № 1(9). С. 113-119.

35. Бет Э. Математическая теория логического вывода / Бет Э. – М.: Наука, 1967. – 523 с.

36. Бланк, С. Стартап. Настольная книга основателя / С. Бланк, Б. Дорф ; пер. с англ. Т. Гутман, И. Окунькова, Е. Бакушева. – 2-е изд. – Москва : Альпина Паблишер, 2014. – 614 с.

37. Бочаров В. А., Маркин В. И. Основы логики: Учебник. — М.: ИНФРА-М, 2001. — 296 с

38. Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений. 2009. № 2. С. 45-57.

39. Васильева Т.Н., Мамонова Т.Е. Применение методов искусственного интеллекта. // XII Международная научно-практическая конференция студентов, аспирантов и молодых ученых «Молодежь и современные информационные технологии». Томск, 2014. С. 402-403.

40. Гайдамакин Н. А. Автоматизированные системы, базы и банки данных. Вводный курс : Учебное пособие. — М.: Гелиос АРВ, 2002. — 368 с.

41. Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы (обзор) // Новости искусственного интеллекта. – 1998. – №2. – С. 64–116.

42. Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. — Х.: ХНАМГ, 2010. — 222 с.

43. Гуревич Л. А., Вахитов А. Н. Мультиагентные системы [Текст] / Л. А. Гуревич, А. Н. Вахитов // Введение в Computer Science. – 2005. –с.116- 139

44. Д.Г. Досин Розробка онтології матеріалознавства засобами Protégé-OWL / Д.Г. Досин, Р.Р. Даревич, Н.В. Шкутяк, Фізико-механічний інститут імені Г.В. Карпенка НАН України, м. Львів, 2008
45. Драгалин А.Г. Введение в математическую логику / Драгалин А.Г., Колмогоров А.Н. – М.: МГУ, 1982. - 120 с
46. Дрейпер, У. Стартапы : профессиональные игры Кремниевой долины / У. Дрейпер ; предисл. Э. Шмидта ; пер. с англ. В. Егорова. – Москва : Эксмо, 2012. – 378 с.
47. Захарова И. В Способы автоматического построения онтологии для задач анализа текстов, Институт математики им. С. Л. Соболева СО РАН, 2011
48. Иващенко А. В. Мультиагентные системы для управления производством в реальном времени, Научно-производственная компания «Разумные решения», 2011
49. Избачков Ю. С. Информационные системы: учебник : — 2-е изд. — СПб: Питер, 2008. — 656 с.
50. Информационные системы в экономике. Под ред. Г.А. Титоренко. – 2-е изд., перераб. и доп. – М.: ЮНИТИ-ДАНА, 2008. – 463с.
51. Копайгородский А.Н. Применение онтологий в семантических информационных системах, «Ontology of Designing» scientific journal, 2014, 78-89с.
52. Коэн, Д. Стартап в Сети : мастер-классы успешных предпринимателей / Д. Коэн, Б Фелд ; пер. с англ. М. Иутина. – 2-е изд. – Москва : Альпина Паблишер, 2013. – 337 с.
53. Маллинс, Дж. Поиск бизнес-модели : как спасти стартап, вовремя сменив план / Дж. Маллинс, Р. Комисар ; пер. с англ. М. Пуксант и Е. Бакушевой. – Москва : Манн, Иванов и Фербер, 2012. – 329 с.
54. Маркелов В.М. Применение мультиагентных систем для управления логистическими системами // Славянский форум. 2014. № 2 (6). С. 82-87.

55. Маслов В.П. Інформаційні системи і технології в економіці : Посібник для студ. вузів/ В.П. Маслов; М-во освіти і науки України. -К.: Слово, 2005. -263 с.
56. Методи та засоби мультимедійних інформаційних систем : навч. посіб. / Т. М. Басюк, П. І. Жежнич; Нац. ун-т "Львів. політехніка". - Львів : Вид-во Львів. політехніки, 2015. - 426 с. - Бібліогр.: с. 413-416.
57. Найданов Д.Г., Шеин Р.Е. Онтологии в мультиагентных системах / XII Всероссийское совещание по проблемам управления/ ВСПУ-2014. – с.9044-9049)
58. Науменко Т.О. Використання онтологій для зберігання та обміну занннями в мультиагентних системах / Міжнародний науковий журнал «Інтернаука» // № 6 (28), 2017. – с.50-52
59. Основи інформаційних систем : Навч. посіб./ Віктор Ситник, Тамара Писаревсь-ка, Ніна Єрьоміна та ін.; За ред. В.Ф.Ситника; М-во освіти України. Київський нац. еко-ном. ун-т. -2-е вид., перероб. і доп.. -К.: КНЕУ, 2001. -420 с.
60. П.І. Андон Проблеми і можливості програмування в середовищі SEMANTIC WEB / П.І. Андон, Л.П. Бабенко, Інститут программних систем НАН України, 2012
61. Парасюк И.Н., Ершов С.В. Моделе-ориентированная архитектура нечетких мультиагентных систем // Компьютерная математика. 2010. № 2. С. 62-74.
62. Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ. - М.: Высшая школа, 1989
63. Проектування інформаційних систем : Навч. посібник/ Ред. Володимир Пономаренко,. -К.: Академія, 2002. -486 с.
64. Робемед, Н. Самые интересные стартапы 2013 года [Электронный ресурс]. – Режим доступа: http://www.forbes.ru/svoi-biznes-photogallery/startapy/248976-samye-interesnye-startapy-2013-goda/photo/1
65. Скобелев П.О. Приминение онтологии в интеллектуальной системе распределенного управления группировкой малоразмерных космических

аппаратов, Самарский государственный аэрокосмический университет имени академика С.П. Королева, 2015

66. Статистика смертности и советы по безопасности для стартапов [Электронный ресурс]. – Режим доступа: https://vc.ru/p/startup-eset
67. Статистика указала на условия для появления стартапов, успешных как Google и Facebook [Электронный ресурс]. – Режим доступа: https://naked-science.ru/article/sci/statistika-ukazala-na-usloviya
68. Тарасов В.Б. Агенты, многоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте // Новости искусственного интеллекта. 1998. № 2. С. 5-63
69. Тиль, П. От нуля к единице : как создать стартап, который изменит будущее / П. Тиль, Б. Мастерс; перевод с англ. – Москва : Альпина паблишер, 2015. – 188 с.
70. Управление на базе мультиагентных систем / Национальный открытый институт «Интуит» / [электронный ресурс] – Режим доступа: http://www.intuit.ru/studies/courses/13833/1230/lecture/24081
71. Харниш, В. Правила прибыльных стартапов : как расти и зарабатывать деньги / В. Харниш ; пер. с англ. В. Хозинского. – Москва : Манн, Иванов и Фербер, 2012. – 279 с.
72. Цветков В.Я. Информационная модель как основа обработки информации в ГИС // Геодезия и аэрофотосъемка. 2005. № 2. С. 118-122.
73. Цветков В.Я. Применение принципа субсидиарности в информационной экономике // Финансовый бизнес. 2012. № 6. С. 40-43.
74. Экланд С. Ангелы, драконы и стервятники : как привлечь правильных инвесторов в свой стартап и сохранить бизнес / С. Экланд ; пер. с англ. О. Терентьевой. – Москва : Манн, Иванов и Фербер, 2011. – 275 с.