

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

15.03 — КМР. 1940–“С” 2022.12.30. 012 ПЗ

ХАРЧЕНКА ЮРІЯ БОРИСОВИЧА

2023 р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК 004.89:378.14

«ПОГОДЖЕНО»

Декан факультету

Глазунова О. Г., д.п.н., професор

_____ 2023 р.

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Завідувач кафедри комп'ютерних наук
інформаційних технологій

Голуб Б. Л., к.т.н., доцент

_____ 2023 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему **ІНТЕЛЕКТУАЛЬНИЙ АГЕНТ НА ОСНОВІ ЗНАНЬ**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Інформаційні управляючі системи і технології»**

Орієнтація освітньої програми **освітньо-професійна**

Гарант освітньої програми

к.т.н., доцент

(підпис)

Голуб Белла Львівна

Керівник магістерської кваліфікаційної роботи

к.т.н., доцент

(підпис)

Дудник Алла Олексіївна

Виконав

(підпис)

Харченко Юрій Борисович

КИЇВ-2023

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет (ННІ) Інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

к.т.н., доцент _____ Голуб Б. Л.

“ ____ ” _____ 2023 року

**ЗАВДАННЯ
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ**

Харченко Юрію Борисовичу

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Інформаційні управляючі системи і технології»**

Орієнтація освітньої програми **освітньо-професійна**

Тема магістерської кваліфікаційної роботи **Інтелектуальний агент на основі знань**

затверджена наказом ректора НУБіП України від **“30” грудня 2022 р. № 1940–“С”**

Термін подання завершеної роботи на кафедру **“05” листопада 2023 р.**

Вихідні дані до магістерської кваліфікаційної роботи

Аналіз сучасного стану в галузі інтелектуальних агентів, вивчення методів та інструментів агентно-орієнтованого дизайну та представлення знань дозволить побудувати архітектуру програмного модуля як системи взаємодіючих агентів, реалізація якого дозволить інтегрувати системи підтримки дистанційного електронного навчання з компонентами структурованих знань та процесів.

Перелік питань, що підлягають дослідженню:

1. Можливість та межі застосування підходів, методів, інструментів моделювання інтелектуальних агентів та представлення структурованих знань.
2. Загальна архітектура системи, структури даних та алгоритми процедур інтелектуальної підсистеми системи дистанційного електронного навчання.
3. Рішення по проектуванню, розробці та розгортанню запропонованої системи.

Дата видачі завдання “ ____ ” _____ 20__ р.

Керівник магістерської кваліфікаційної роботи _____ Дудник А.О. к.т.н., доцент

Завдання прийняв до виконання _____ Харченко Ю.Б.

ЗМІСТ

| | |
|--|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ..... | 4 |
| ВСТУП..... | 6 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ..... | 10 |
| 1.1 KR, когнітивні процеси та міркування..... | 10 |
| 1.2 Адаптивність, навчання та вирішення задач..... | 17 |
| 1.3 Класифікація ІА..... | 26 |
| 1.4 КВІА в EL..... | 28 |
| 2 МОДЕЛЮВАННЯ СИСТЕМИ..... | 31 |
| 2.1 КВІА як система агентів..... | 31 |
| 2.2 Моделі агентів..... | 35 |
| 2.3 Моделі представлення знань..... | 41 |
| 2.4 Моделі планування та прийняття рішень..... | 48 |
| 3 РОЗРОБКА СИСТЕМИ..... | 55 |
| 3.1 Предметна область та архітектура рішення..... | 55 |
| 3.2 Компоненти системи та їхня взаємодія..... | 58 |
| 3.3 Реалізація KB..... | 60 |
| 3.4 Обробка текстів природної мови..... | 62 |
| 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ..... | 64 |
| ВИСНОВКИ..... | 68 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 70 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AI - artificial intelligence (штучний інтелект)

AOD - agent-oriented design (агентно-орієнтований дизайн)

API - application programming interface (прикладний програмний інтерфейс)

BDI - belief-desire-intention (переконавання-бажання-намір)

BN - Bayesian networks (Байєсівські мережі)

CBE - competency-based education (навчання на основі компетентностей)

CS - cognitive science (когнітивна наука)

DAG - directed acyclic graph (орієнтований ациклічний граф)

EL - E-Learning (електронна дистанційна освіта)

IA - intelligent agent (інтелектуальний агент)

ITS - intelligent tutoring system (інтелектуальна система навчання)

KB - knowledge base (база знань)

KBIA - knowledge-based IA (IA на основі знань)

KR - knowledge representation (представлення знань)

LLM - large language model (велика лінгвістична модель)

LMS - learning management system (система управління навчанням)

MAS - multi-agent system (багатоагентна система)

MDP - Markov decision processes (Марковський процес прийняття рішень)

ML - machine learning (машинне навчання)

NLP - natural language processing (обробка природної мови)

PLE - personal learning environment (особисте навчальне середовище)

POMDP - partially observable Markov decision processes (частково спостережуваний MDP)

RL - reinforcement learning (навчання з підкріпленням)

ВСТУП

Сучасний стан розвитку інформаційних технологій та комп'ютерних комунікацій призвів до того, що нас оточують петабайти неструктурованої та слабоструктурованої цифрової інформації. Цифрова інфраструктура через сучасні засоби комунікації: комп'ютери, мобільні пристрої, інтернет речей (IoT), надає онлайн та оффлайн доступ до широкого кола джерел, таких як індексовані сховища даних, створені пошуковими системами (Google, Bing, Yahoo!, Baidu, DuckDuckGo); цифрові енциклопедії (Wikipedia, Encyclopædia Britannica); галузеві онтології на базі Semantic Web; відкриті дані урядових та неурядових організацій з доступом через прикладні програмні інтерфейси (API). Зростає потужність персональних обчислювальних та комунікаційних пристроїв, розвиваються теорії та методи в сфері представлення знань, з'являються нові мови програмування та інструментальні засоби сфери штучного інтелекту (AI) придатні для вирішення задач представлення знань (KR), логічного виводу, оптимізації, пошуку.

Нагальним завданням стає поєднання доступу до практично необмеженої інформації з сучасним рівнем розвитку теорій, методів, технологій сфери AI для вирішення поточних задач пересічного користувача з пошуку, отримання та організації інформації шляхом перетворення її на знання. Результатом такої інтеграції має стати інтелектуальний агент (IA), який буде в автоматизованому режимі згідно визначених користувачем напрямків створювати базу знань (KB), вести двосторонній обмін запитами з користувачем на формалізованій підмножині природної мови, відповідати на питання та ставити їх сам. Такий IA в якості додатку на комп'ютері чи гаджеті може стати особистим секретарем-референтом, консультантом чи навіть експертом з визначених користувачем питань.

В магістерській роботі пропонується дослідити підходи до застосування ІА на основі знань (КВІА) в галузі електронної дистанційної освіти - E-Learning (EL). EL розглядається як індивідуальний і налаштовуваний сервіс, який дозволяє кожному користувачеві мати простий доступ через мережу до інструментів, послуг і цифрових засобів, необхідних для навчання [1]; як “поєднання навчальних послуг і комп'ютерних технологій для забезпечення високої результативності (high value) інтегрованого навчання: будь-коли (anytime), будь-де (anyplace)” [2]. “Це не те, щоб пройти курс і розмістити його на робочому столі, це нове поєднання ресурсів, інтерактивності, підтримки ефективності та структурованої навчальної діяльності” [2].

Об'єктом дослідження є КВІА, які діють в змінюваному віртуальному середовищі. Предмет дослідження - аналіз підходів агентно-орієнтованого дизайну (AOD) та методів KR для застосування їх в системах підтримки EL, таких як LMS (Learning Management System), ITS (Intelligent Tutoring System), CBE (Competency-Based Education), CAI (Computer-Assisted Instruction), PLE (Personal Learning Environment).

Мета дослідження полягає в тому, щоб шляхом аналізу сучасного стану в галузі ІА, вивчення методів та інструментів AOD та KR, визначення можливостей різних методів адаптації та навчання побудувати прототип програмного модуля як системи взаємодіючих КВІА, реалізація якого дозволила б інтегрувати системи підтримки EL з компонентами структурованих знань та процесів, які представляють таку предметну область як вища освіта з її стандартами, ресурсами, програмами та акторами.

Виходячи з предмета дослідження визначено послідовність завдань, виконання яких дозволяє досягти поставлених в роботі мети та цілей.

На стадії аналізу визначити можливість і межі застосування імовірнісної, нечіткої, динамічної, темпоральної логіки, числення подій; методів автоматичного доведення теорем та резолюцій; алгоритмів теорії графів,

динамічного програмування, навчання з підкріпленням (RL), байєсівського аналізу; методів KR: декларативних, процедурних, гібридних; порівняти та вибрати для застосування мови програмування та інструменти проектування і реалізації.

На стадії проектування розробити загальну архітектуру системи; визначити структури даних та алгоритми процедур.

На стадії реалізації згідно проєкту створити прототип програмного модуля, як багатоагентної системи (MAS) взаємодіючих KBIA.

Основними методами виконання роботи є пошук та аналіз джерел релевантних до об'єкта, предмета, мети та завдань дослідження; практична апробація вибраних підходів, методів та інструментів на тестових і реальних даних; порівняння та висновки стосовно можливого застосування випробуваних артефактів. Джерела включають сайти, книги, наукові статті, обговорення, стандарти, патенти, документація на програмні засоби, вихідні коди програм. Практична апробація здійснюється шляхом комп'ютерного моделювання, програмування прототипів, підготовки та аналізу даних, виконання тестових прикладів. Порівняння результатів апробації дозволяє прийняти рішення щодо подальшого використання досліджуваних підходів, методів, структур, інструментів, мов, пакетів, бібліотек для виконання поставлених в дослідженні завдань.

Запропоновані підходи до застосування KBIA в галузі EL матимуть певне інноваційне значення та актуальну практичну цінність в разі використання результатів цієї магістерської роботи для створення реального програмного проєкту та продукту для розширення існуючої системи EL в бік більшої інтерактивності, адаптивності та інтелектуальності.

Апробація результатів дослідження провадилася шляхом публікації тез та виступів на конференціях, таких як "XIII Міжнародна науково-практична

конференція студентів, аспірантів та молодих вчених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ЕКОНОМІКА, ТЕХНІКА, ОСВІТА»» 26-27 жовтня 2022 р, «XIV Міжнародна науково-практична конференція студентів, аспірантів та молодих вчених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ЕКОНОМІКА, ТЕХНІКА, ОСВІТА»», 26-27 жовтня 2023 р.

Структура роботи:

сторінок - 71,

використаних джерел - 20,

додатків - 0,

розділів - 6.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

КВІА представляють системи AI, які використовують акумульовані знання для прийняття обґрунтованих рішень і виконання складних завдань. Ці агенти розширюють межі систем, заснованих на правилах (rule-based), застосовуючи такі підходи та методи як семантичне розуміння (semantic understanding), контекстно-орієнтоване міркування (context-driven reasoning) та адаптивна поведінка (adaptive behavior). Вивчення та застосування КВІА обіцяє революцію в усіх галузях людської діяльності, зміну взаємодії між людиною та комп'ютером, переосмислення підходів до вирішення інтелектуальних задач.

Розвиток КВІА ґрунтується на поєднанні різноманітних галузей знань та технологій, включаючи машинне навчання (ML), KR, обробку природної мови (NLP), когнітивну науку (CS) та робототехніку. Таке поєднання дозволяє створювати агентів, здатних обробляти невизначену (uncertain) та неповну (incomplete) інформацію, імітуючи людські процеси міркування (reasoning) та динамічно адаптуючись до змінних контекстів.

На рис. 1 представлена загальна структура об'єкта дослідження. В наступних підрозділах буде більш детально розглянуто кожен з структурних елементів ієрархії.

1.1 KR, когнітивні процеси та міркування

1.1.1 Декларативне представлення знань.

Завдання KR полягає не лише в тому, щоб зберігати й упорядковувати інформацію, а й у тому, щоб наповнити її змістом та контекстом. Завдяки символічним представленням (symbolic representations), онтологіям (ontologies), семантичним мережам (semantic networks) і імовірнісним структурам (probabilistic frameworks) КВІА інтегрують предметно-специфічні знання, що

дозволяє їм орієнтуватися в складних просторах рішень (decision spaces) із врахуванням основних концепцій предметної області (domain).

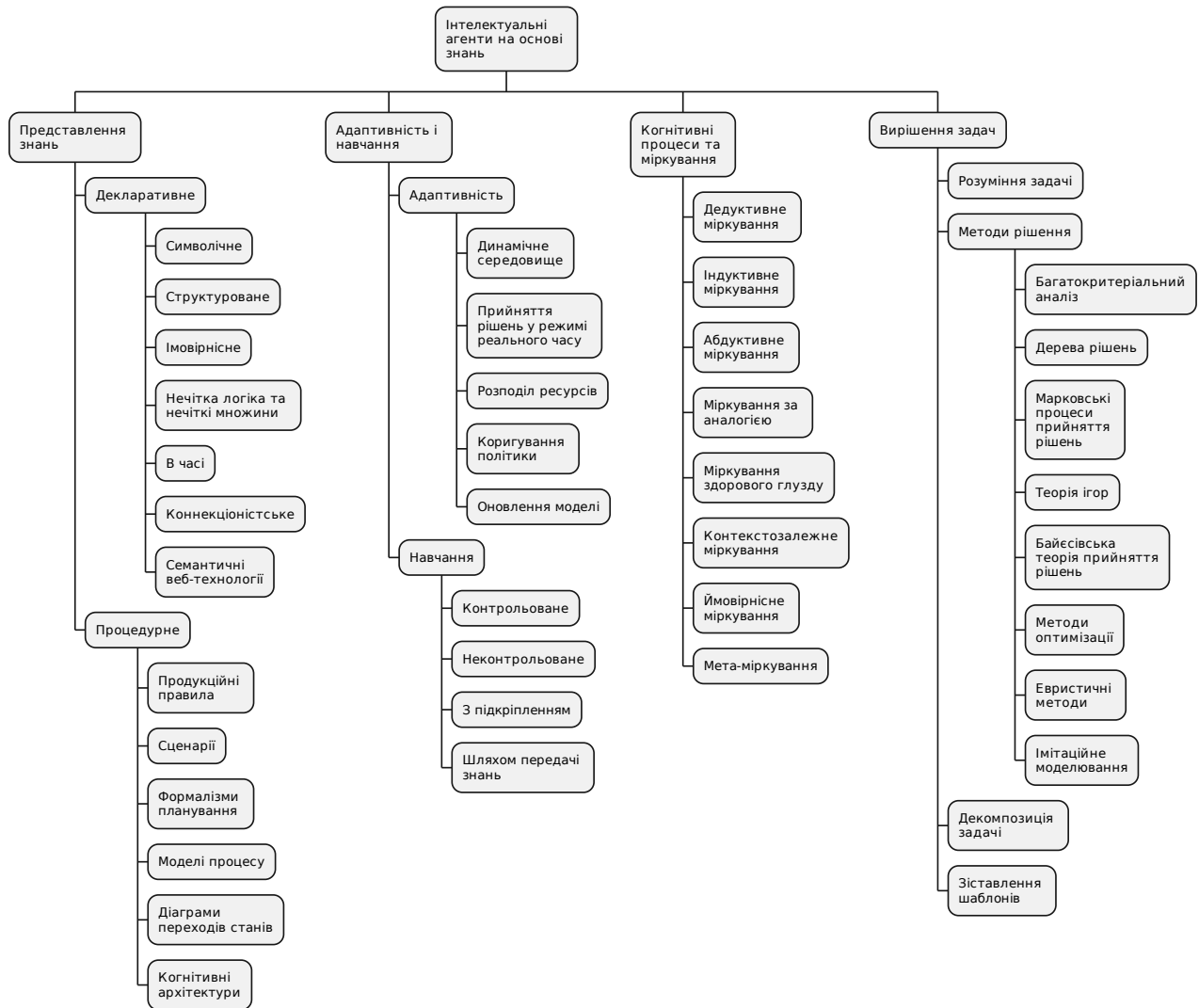


Рис. 1 Структура об'єкта дослідження

Символічне представлення знань. Логіка предикатів представляє знання за допомогою предикатів, змінних і логічних операторів. Прикладами є логіка першого порядку та логіки вищого порядку. Добре підходить для представлення структурованих і формальних знань. Представлення на основі фреймів організовує знання у вигляді фреймів або об'єктів із слотами та атрибутами. Використовується для представлення структурованої інформації та зв'язків успадкування. Семантичні мережі представляють знання у вигляді графів - вузлів (концептів), з'єднаних зв'язками (відношеннями). Корисно для фіксації зв'язків і асоціацій між поняттями.

Онтології, графи знань (knowledge graphs), дескриптивна логіка пропонують мови та моделі для відображення структурованих знань. Онтологія - формальне представлення понять, зв'язків і атрибутів у межах предметної області. Онтології визначають спільний словниковий запас і семантику, що дозволяє міркувати та взаємодіяти. Графи знань використовують графові структури для представлення понять і зв'язків, уможливлючи семантичні висновки та міркування. Дескриптивна логіка це підмножина формальної логіки, яка використовується в онтологіях для вираження зв'язків, обмежень і аксіом.

Для імовірнісного представлення знань використовуються мережі Байєса та марківські моделі. Мережі Байєса представляють невизначені знання за допомогою імовірнісних залежностей між змінними та використовуються для моделювання ймовірнісних залежностей. Моделі Маркова відображають послідовні процеси з імовірнісними переходами. Використовуються для таких завдань, як розпізнавання мовлення, NLP та машинний переклад тощо.

Нечітка логіка та нечіткі множини. Нечітка логіка представляє невизначеність у знаннях за допомогою лінгвістичних термінів і функцій належності. Нечіткі множини відображають ступені приналежності до понять, дозволяючи поступовий перехід між категоріями.

Представлення знань в часі. Модальна (modal logic) та часова логіка (temporal logic) розширюють логіку предикатів на міркування про час і часові зв'язки. Корисна для моделювання подій, причинності та часових обмежень. Часові бази даних зберігають дані з мітками часу та дають змогу робити запити та міркування щодо часової інформації.

Коннекціоністське представлення знань це нейронні мережі, які пропонують підходи відображення знань та залежностей в даних за допомогою взаємопов'язаних вузлів (нейронів). Використовуються для завдань ML та розпізнавання образів.

Семантичні веб-технології та такі мови представлення знань як Resource Description Framework (RDF) і Web Ontology Language (OWL) використовуються для створення структурованих, машиночитаних веб-даних. Поєднують пов'язаність даних та семантичну сумісність.

Кожен метод декларативного представлення знань має свої переваги та обмеження. Вибір залежить від характеру представлених знань, необхідних можливостей міркування та конкретної області застосування. На практиці часто використовуються гібридні підходи, які поєднують різні методи, для подолання обмежень одних і використання переваг інших.

1.1.2 Процедурне представлення знань.

Процедурне представлення знань в KBIA відображає інформацію про процеси, процедури, дії та послідовності кроків, необхідних для досягнення конкретних цілей. Процедурні знання зазвичай представлені у спосіб, який дозволяє агентам виконувати дії та слідувати встановленим процедурам для виконання завдань.

Продукційні правила складаються з набору умовних операторів (правил «якщо-тоді»), що визначають дії, які необхідно виконати, коли виконуються певні умови. Добре підходять для представлення експертних знань і процедурної інформації. Вони використовуються в експертних системах на основі правил і системах підтримки прийняття рішень.

Сценарії це структуровані представлення типових послідовностей дій. Вони описують кроки, необхідні для виконання завдань або досягнення цілей. Сценарії використовуються для представлення процедурних знань у випадках, де завдання мають передбачувану послідовність дій, таких як NLP або планування.

Формалізми планування (planning formalisms) представляють формальні знання про дії та їхні зв'язки. Вони включають такі представлення, як STRIPS,

ADL і PDDL. Формалізми планування використовуються для визначення дій, їхніх передумов, наслідків і зв'язків між діями. Вони забезпечують автоматизоване планування та міркування.

Моделі процесу використовують графічні позначення (наприклад, блок-схеми, мережі Петрі) для представлення потоку дій, рішень і взаємодій у процесі. Моделі процесів широко використовуються в управлінні бізнес-процесами, системах документообігу та промисловій автоматизації.

Діаграми переходів станів представляють стани системи та переходи між станами на основі конкретних подій або дій. Вони зазвичай використовуються для моделювання поведінки систем, які мають дискретні стани та переходи.

Гібридні підходи поєднують різні методи для представлення процедурних знань. Наприклад, поєднання продукційних правил з діаграмами переходів станів. Вони пропонують гнучкість у представленні складних процедур, які включають як логічне міркування, так і послідовні дії.

Когнітивні архітектури, такі як Soar або ACT-R, використовують продукційні правила та інші елементи для представлення процедурних знань разом з іншими типами знань.

Вибір методів представлення процедурних знань залежить від складності завдань, характеру процедур, необхідності формального обґрунтування та конкретних вимог застосування КВІА. У багатьох випадках для ефективного охоплення та представлення процедурних знань використовується комбінація методів.

1.1.3 Когнітивні процеси та міркування.

Процеси міркування КВІА охоплюють дедукцію (deduction), індукцію (induction), абдукцію (abduction) та міркування за аналогією (analogical reasoning). Агенти обробляють сховища даних, встановлюють кореляції та залежності, роблять висновки про причинно-наслідкові зв'язки та усувають

прогалини в інформації, що дозволяє їм отримати логічні висновки про нюанси взаємозв'язків між сутностями та процесами предметної області.

Дедуктивне міркування забезпечує конкретні висновки із загальних принципів або посилок. КВІА використовує дедуктивне міркування для застосування правил, фактів і логічних зв'язків для отримання точних висновків. Наприклад, система медичної діагностики може використовувати дедуктивні міркування, щоб зробити висновок про конкретне захворювання на основі спостережуваних симптомів і медичних знань.

Індуктивне міркування передбачає отримання узагальнених висновків із конкретних спостережень або випадків, виявляє закономірності, тенденції і зв'язки у даних. Наприклад, система фінансового прогнозування може використовувати індуктивне міркування для прогнозування ринкових тенденцій на основі історичних даних.

Абдуктивне міркування виводить висновок, який дає найкраще пояснення спостережуваних явищ, навіть якщо пояснення не є гарантовано істинним. КВІА використовує абдуктивні міркування для створення гіпотез і обґрунтованих припущень щодо основних причин. Наприклад, діагностична система може використовувати абдуктивні міркування, щоб запропонувати потенційні причини набору симптомів. Часто використовується для створення обґрунтованих припущень або гіпотез щодо основних причин або механізмів, які могли призвести до спостережуваної ситуації. Абдуктивне міркування є ключовим аспектом вирішення проблем, створення гіпотез і наукових відкриттів.

Міркування за аналогією передбачає визнання подібності між різними ситуаціями чи випадками та використовує знання з однієї області для розуміння іншої. КВІА використовує аналогічні міркування для передачі знань і рішень з одного контексту в інший. Наприклад, КВІА, що надає допомогу в інженерному проектуванні, може спиратися на минулі успішні проекти для створення рішень

для нових завдань. Міркування за аналогією покращує здатність КВІА переносити ідеї, рішення та знання зі знайомих контекстів у нові та незнайомі сценарії.

Міркування здорового глузду (*common sense reasoning*) узагальнює знання, які люди сприймають як належне. Докладаються зусилля для розробки КВІА, здатних робити виведення знань здорового глузду, щоб покращити розуміння природної мови та сценаріїв реального світу.

Контекстозалежне міркування враховує контекстну інформацію під час прийняття рішень. Воно дозволяє аналізувати навколишній контекст, включаючи історію користувача, уподобання та поточну ситуацію, щоб створити відповіді, які відповідають створеному контексту. Контекстозалежне обґрунтування підвищує релевантність та ефективність взаємодії КВІА та людини.

Ймовірнісне міркування забезпечує обробку невизначеності шляхом призначення ймовірностей різним гіпотезам або висновкам, використовує теорію ймовірностей для моделювання та кількісної оцінки невизначеної інформації. КВІА використовують ймовірнісні міркування для управління неповною або суперечливою інформацією. Вони зважують докази та генерують ймовірнісні оцінки для прийняття рішень.

Мета-міркування, також відоме як «мислити про мислення», це міркування вищого рівня, яке допомагає КВІА вибрати найефективніші стратегії, евристики та алгоритми для досягнення своїх цілей. Передбачає рефлексію агента над власними процесами міркування. Це допомагає визначити найбільш ефективний підхід до вирішення конкретної проблеми, враховуючи власні сильні сторони та обмеження.

КВІА поєднує ці різні форми міркувань, щоб орієнтуватися в складних просторах рішень, адаптуватися до мінливих контекстів і доходити до

обґрунтованих висновків. Інтегруючи ці когнітивні процеси у свою діяльність, агент симулює людський інтелект і сприяє вирішенню складних проблем у таких областях, як охорона здоров'я, фінанси, інженерія тощо.

1.2 Адаптивність, навчання та вирішення задач

Можливості КВІА для вирішення задач зростають завдяки навчанню та адаптації. Методи ML, що охоплюють RL, глибоке навчання та навчання шляхом передачі знань (transfer learning), дають змогу агентам уточнювати свої стратегії вирішення задач на основі нового досвіду та нової інформації.

1.2.1 Адаптивність та навчання

Адаптивність означає здатність КВІА коригувати свою поведінку відповідно до обставин, що змінюються, тоді як навчання передбачає отримання нових знань і навичок на основі даних і досвіду. Такі властивості агентів забезпечують можливість реагування на зміну середовища, дозволяючи їм підтримувати максимальну продуктивність у різноманітних областях та сценаріях.

КВІА працюють в динамічних середовищах, які можуть змінюватися непередбачувано. Здатність до адаптації дозволяє їм ефективно реагувати на нові дані, ситуації та виклики. У сценаріях, де потрібно приймати рішення у режимі реального часу, адаптивність гарантує, що агенти можуть коригувати свої стратегії у відповідь на зміну інформації. Адаптивність допомагає оптимально розподіляти ресурси (обчислювальні ресурси, час, увагу тощо) відповідно до поточного стану та вимог задачі.

КВІА, які використовують RL, можуть адаптувати та коригувати політики прийняття рішень на основі відгуків про свої дії. Це призводить до покращення ефективності політик, оскільки вони наближуються до оптимальних. У ML адаптивність передбачає оновлення моделей у міру появи нових даних. Це запобігає старінню моделей і допомагає КВІА залишатися актуальним.

Контрольоване навчання (supervised learning) полягає в навчанні на позначених даних, де вхідні дані поєднуються з правильними вихідними. Це дає агентам змогу робити узагальнення на основі прикладів і робити прогнози на основі нових невідомих даних. Неконтрольоване навчання (unsupervised learning) використовується для виявлення закономірностей і зв'язків у даних без явних позначок. Поширеними застосуваннями є кластеризація та зменшення розмірності.

RL відбувається через взаємодію з навколишнім середовищем через отримання нагород та покарань за результатами своїх дій. КВІА вдосконалюють свою політику прийняття рішень, максимізуючи сукупні винагороди.

Навчання шляхом передачі знань (transfer learning) - це можливість використання знань, отриманих під час виконання одного завдання, щоб покращити ефективність виконання пов'язаних завдань. Це прискорює навчання в нових сферах і зменшує потребу в тривалому навчанні.

1.2.2 Вирішення задач.

КВІА відіграють важливу роль у підходах до вирішення задач (problem solving), надаючи передові інструменти та методи для вирішення складних проблем у різноманітних предметних областях. Вони інтегрують представлення знань, міркування, навчання та адаптацію, щоб запропонувати ефективні можливості вирішення задач:

- зберігають та організовують предметно-специфічні знання в структурованих форматах, таких як онтології, семантичні мережі або графи знань; ці знання формують основу для розуміння та вирішення задач у певній предметній області;
- аналізують формулювання задач і вхідні дані, щоб зрозуміти контекст та вимоги проблеми; використовують свою КВ, щоб отримати відповідну інформацію та визначити ключові елементи проблеми;

- виконують міркування (дедуктивні, індуктивні, абдуктивні, аналогічні), щоб робити висновки, генерувати гіпотези та приймати обґрунтовані рішення на основі логічних правил та зв'язків зі своєї KB;
- обробляють складні простори прийняття рішень, враховуючи численні фактори, невизначеності та компроміси, використовуючи імовірнісне міркування, методи оптимізації та адаптивні стратегії для прийняття рішень, які відповідають бажаним результатам;
- використовують навчання шляхом передачі знань та аналогічне міркування для розуміння проблеми з іншої предметної області; це прискорює вирішення задач у нових контекстах, шляхом застосування результатів, отриманих в минулому досвіді;
- розбивають складні проблеми на менші підзадачі для простішого аналізу та вирішення; використовують свої знання предметної області, щоб визначити зв'язки між підзадачами та координувати їх рішення;
- вивчають нові дані та досвід, адаптуючи свої стратегії та знання до мінливого середовища; адаптивність допомагає їм залишатися ефективними;
- можуть брати участь у творчому вирішенні проблем, досліджуючи нестандартні рішення або поєднуючи різноманітні знання для створення інноваційних підходів;
- можуть діяти як персоналізовані системи рекомендацій, допомагаючи користувачам знаходити оптимальні рішення, продукти або контент на основі їхніх уподобань і минулих взаємодій.

KBIA роблять внесок у теорію вирішення задач, забезпечуючи систематичну структуру для підходу до складних завдань у широкому діапазоні областей. Інтегруючи знання, міркування, навчання та здатність до адаптації,

вони пропонують ефективні рішення для задач, які були б складними для традиційних алгоритмів або людського досвіду.

Розуміння проблеми є критично важливим кроком у процесі вирішення задачі. Це передбачає розуміння контексту, вимог, обмежень та відповідної інформації щодо даної проблеми. Агенти використовують свої можливості представлення знань та міркування, щоб проаналізувати постановку задачі, виділити важливі деталі та сформулювати чітке розуміння того, що потрібно вирішити.

Для цього KBIA здійснюють такі кроки:

- розпізнають та тлумачать терміни, поняття та сутності, згадані у постановці задачі, використовуючи свої знання предметної області;
- виконують семантичний аналіз, щоб виділити значення слів і фраз у постановці задачі, визначають ключові терміни, атрибути та зв'язки, які відіграють ролі у проблемі;
- визначають контекст, у якому існує задача, аналізують передумови, цілі та наслідки проблеми, щоб визначити їхнє значення та релевантність;
- ідентифікують обмеження та вимоги, пов'язані з задачею, що дозволяє звужити потенційний простір рішення;
- інтерпретують та попередньо обробляють вхідні задачі дані для визначення значущих шаблонів і зв'язків;
- розбивають складні задачі на менші підзадачі або завдання, що спрощує процес розв'язання проблеми і дозволяє розглядати окремі компоненти рішення окремо;
- виконують вивід та абдуктивне міркування для створення гіпотез і пояснень для неоднозначної або неповної інформації в постановці задачі, допомагають заповнити прогалини та отримати більш повне розуміння;

- аналізують завдання та цілі задачі, щоб визначити, що потрібно досягти та можливо оптимізувати;
- розглядають задачу в межах ширшого контексту предметної області, пов'язують проблему з наявними знаннями та досвідом, що дозволяє їм провести паралелі та використати аналогічні рішення.

Здатність КВІА інтерпретувати й аналізувати постановки задач, використовуючи свої знання та міркування, дозволяє їм генерувати точні ідеї, визначати відповідні стратегії та приймати обґрунтовані рішення для досягнення бажаних результатів.

КВІА використовують широке коло методів для прийняття рішень, які включають врахування багатьох факторів, невизначеності та компроміси. Ці методи дозволяють приймати обґрунтовані й оптимальні рішення в складних і динамічних середовищах.

Багатокритеріальний аналіз рішень (multi-criteria decision analysis) передбачає оцінку альтернатив на основі кількох критеріїв або цілей. КВІА можуть використовувати цей метод для кількісної оцінки та порівняння впливу різних варіантів на результати рішення, враховуючи як кількісні, так і якісні фактори.

Дерева рішень (decision trees) застосовуються для моделювання послідовних рішень і результатів. Вони допомагають оцінити потенційні результати та пов'язані з ними ймовірності, щоб вибрати найкращий напрям дій у кожній точці прийняття рішення.

Марковські процеси прийняття рішень (Markov decision processes - MDP) моделюють складні задачі прийняття рішень, які включають стани, дії, ймовірності переходу та винагороди. Використовуються такі алгоритми, як Q-навчання або ітерація по політиках, щоб знайти оптимальну політику. Теорія ігор використовується у сценаріях із залученням кількох агентів або

зацікавлених сторін для моделювання взаємодії, стратегій і виграшів, аналізуються рівноваги Неша для прийняття стратегій рішення задачі. Байєсівська теорія прийняття рішень (Bayesian decision theory) дозволяє приймати рішення на основі попередніх знань і нових свідчень, оновлюючи ймовірності та роблячи вибір, який максимізує очікувану корисність.

RL визначає оптимальну політику прийняття рішень через взаємодію з навколишнім середовищем досліджуючи різні варіанти дій, щоб максимізувати сукупні винагороди.

Методи оптимізації шукають рішення, які задовольняють певним обмеженням, одночасно оптимізуючи цілі. Лінійне, цілочисельне та нелінійне програмування є найбільш поширеними методами. Евристичні методи допомагають швидкому пошуку приблизних рішень складних задач, використовуючи емпіричні правила або стратегії, засновані на досвіді, щоб керувати процесом прийняття рішень. Імітаційне моделювання сценаріїв і оцінки результатів рішень допомагає визначити можливі наслідки результатів рішення задачі.

КВІА використовують ці методи для навігації в складному просторі прийняття рішень, врахування невизначеностей і збалансування суперечливих цілей. Використовуючи комбінацію математичних моделей, алгоритмів машинного навчання та предметно-спеціальних знань, агенти здатні приймати добре обґрунтовані рішення, які відповідають цілям та бажаним результатам.

Декомпозиція — це процес поділу складної задачі на менші більш керовані підзадачі або завдання. КВІА використовують декомпозицію, щоб спростити процес вирішення задач, підвищити ефективність і полегшити координацію рішень.

Можливі кроки декомпозиції складної задачі:

- виявити підзадачі, пов'язані з різними аспектами або компонентами більшої проблеми, які можна вирішити окремо;
- створити ієрархічну структуру, організовуючи підзадачі у вигляді дерева, де кожен рівень ієрархії представляє різний рівень деталізації з вихідною задачею на вершині;
- призначити підзадачі спеціалізованим модулям або підсистемам агента, використовуючи їхній досвід та можливості в конкретних сферах;
- розглянути кілька можливостей рішення підзадач паралельно, розподіляючи навантаження між різними процесами або модулями;
- використовувати базу знань для обміну інформацією між підзадачами, що полегшить передачу знань та забезпечить послідовність міркувань і прийняття рішень;
- визначити порядок, у якому слід вирішувати підзадачі, щоб забезпечити логічне й ефективне просування рішення;
- при потребі розкласти складні підзадачі на ще менші підзадачі, використовуючи рекурсивний підхід, поки задачі не стануть достатньо простими, щоб їх було легко розв'язати;
- інтегрувати рішення підзадач у загальне рішення вихідної задачі, щоб інтегроване рішення було узгодженим і відповідало цілям проблеми;
- виявити сфери, де знань недостатньо для вирішення конкретних підзадач для керування стратегіями навчання та отримання знань.

Декомпозиція задач дозволяє КВІА керувати складністю масштабних задач, ефективно використовувати свій досвід у прикладній області та раціонально розподіляти ресурси, розбиваючи проблеми на керовані компоненти, вирішувати кожен підзадачу окремо, а потім інтегрувати рішення для комплексного вирішення загальної задачі.

Зіставлення шаблонів (pattern matching) — це техніка, яку КВІА можуть використовувати для пошуку рішень задач шляхом виявлення подібності між відомими шаблонами в їхній КВ і наявною проблемою. Зіставлення шаблонів передбачає порівняння характеристик проблеми зі збереженими шаблоном або шаблонами, що дозволяє агенту розпізнати знайомі ситуації та застосовувати відповідні рішення.

Для застосування зіставлення шаблонів КВІА виконує такі дії:

- зберігає шаблони в структурованих форматах, таких як правила, онтології тощо, фіксуючи в шаблонах інформацію про атрибути та відношення, які стосуються вирішення задачі;
- визначає шаблони, які представляють типові ситуації, сценарії або стратегії вирішення проблем у своїй базі знань, відображаючи основні властивості різних типів задач;
- аналізує задачу, вихідні дані та контекст, щоб отримати відповідну інформацію про компоненти, атрибути та зв'язки;
- порівнює отриману інформацію про задачу з попередньо визначеними шаблонами в своїй базі знань, шукаючи відповідності між властивостями проблеми та характеристиками, описаними в шаблонах;
- використовує алгоритми, які вимірюють подібність між проблемою та збереженими шаблонами, враховуючи атрибути, зв'язки та, можливо, варіанти, щоб визначити ступінь відповідності;
- після виявлення відповідного шаблону отримує рішення, стратегії чи рекомендації зі своєї бази знань, які були попередньо представлені цим шаблоном;
- адаптує або налаштовує отримані результати відповідно до особливостей поточної задачі, що може передбачати зміну параметрів, коригування стратегій або включення додаткового контексту;

- вивчає результати, оцінюючи ефективність застосованих рішень; якщо рішення мають успішні результати, вони стають частиною репертуару для майбутнього зіставлення шаблонів;
- враховує варіативність задач, визначаючи шаблони, які охоплюють низку можливих ситуацій, використовуючи нечітку логіку або ймовірнісне міркування для обробки випадків, які не відповідають шаблонам;
- оновлює та вдосконалює шаблони на основі нових даних та досвіду, гарантуючи, що шаблони залишаються актуальними та ефективними у вирішенні релевантних проблем.

Зіставлення шаблонів дозволяє КВІА швидко розпізнавати та вирішувати знайомі проблеми без необхідності отримувати рішення з нуля. Використовуючи існуючі шаблони, агент може дати ефективне та послідовне вирішення проблем у різних сценаріях і сферах.

1.2.3 Великі лінгвістичні моделі.

Великі лінгвістичні моделі (LLM), займають чільне місце в поточному ландшафті КВІА. Вони являють собою дивовижне явище перетину різних методів AI, включаючи KR і NLP.

LLM розуміють і генерують текст, забезпечуючи більш природну та інтуїтивно зрозумілу взаємодію між користувачами та системами AI. Ці моделі використовують величезні обсяги текстових даних для розуміння контексту, семантики та нюансів мови, що робить їх важливими компонентами КВІА, які мають ефективно спілкуватися з людьми.

Хоча LLM не мають явних представлень знань у традиційному розумінні, вони можуть використовувати свої навчальні дані для зберігання інформації з широкого кола тем. Вони можуть надавати контекстно-релевантні відповіді, маючи доступ до знань, вбудованих у їхні навчальні дані, що фактично діє як неявна форма включення знань.

LLM сприяють розробці інтелектуальних агентів, які можуть брати участь у змістовних розмовах з користувачами. Ці взаємодії виходять за рамки простих відповідей на запитання та включають міркування, розуміння контексту та керування діалогом.

КВІА часто вимагає можливості отримувати інформацію з величезних сховищ знань. LLM можуть діяти як інтелектуальні помічники, отримуючи релевантну інформацію зі своїх навчальних даних або зовнішніх джерел, допомагаючи користувачам швидко знаходити відповіді та рішення.

LLM допомагають ефективному прийняттю рішень, надаючи ідеї, рекомендації та пояснення природною мовою, покращуючи зрозумілість рішень КВІА.

Хоча LLM можуть робити значний внесок у створення КВІА, вони не еквівалентні повноцінним агентам, які втілюють структуровані знання предметної області та складні можливості міркування. Натомість вони утворюють важливий компонент ширшої екосистеми АІ, яка покращує наше розуміння того, як системи АІ можуть взаємодіяти, міркувати та допомагати людям у різних сферах.

1.3 Класифікація ІА

Пропонується класифікація ІА, яка дозволить підкреслити особливості КВІА. Це дасть більш чітке розуміння їхніх ключових характеристик.

Агенти на основі правил (rule-based agents) дотримуються заздалегідь визначених правил для прийняття рішень і виконання дій. Їм не вистачає здатності вчитися або адаптуватися поза межами початкового набору правил. КВІА виходять за рамки традиційних систем, заснованих на правилах, використовуючи накопичені знання, що дозволяє їм міркувати за межами явних правил і адаптуватися до нових ситуацій.

Агенти реактивного навчання (reactive learning agents) навчаються на прямому досвіді та оптимізують свої дії на основі минулих взаємодій. КВІА інтегрують навчання з накопиченням знань, використовують історичні дані та досвід, щоб покращити процес навчання та прийняття рішень.

Символічні когнітивні агенти (symbolic cognitive agents) використовують символічне міркування, щоб маніпулювати та виводити зв'язки між абстрактними поняттями. КВІА розширюють символічне міркування, використовуючи методи представлення знань, що дозволяє їм розуміти контекст, семантику та складні зв'язки в процесах міркування.

Контекстно-орієнтовні агенти (context-sensitive agents) адаптують свою поведінку на основі контекстних сигналів, покращуючи свою реакцію на різні ситуації. КВІА вирізняються контекстною адаптацією шляхом інтеграції з предметно-специфічними знаннями, що дозволяє глибше зрозуміти контекст і полегшує прийняття обґрунтованих рішень.

Агенти логічного міркування (logical reasoning agents) використовують формальну логіку для виведення висновків із передумов. КВІА доповнюють логічні міркування ймовірнісними та семантичними міркуваннями, що дозволяє їм ефективно керувати невизначеною та неповною інформацією.

Агенти RL (reinforcement learning agents) визначають оптимальні стратегії шляхом взаємодії зі своїм середовищем методом проб і помилок. КВІА поєднують RL зі знаннями предметної області, що дозволяє їм здобувати досвід більш ефективно та раціонально адаптуватися до змін.

Інтегровані навчальні агенти (integrated learning agents) поєднують різні методи штучного інтелекту, такі як нейронні мережі, символічне міркування та RL, щоб досягти різносторонніх підходів до вирішення проблем. КВІА часто використовує гібридні підходи, щоб використовувати сильні сторони відповідних задачам парадигм.

Ця класифікація підкреслює, що KBIA, відрізняються від звичайних IA своєю здатністю акумулювати та використовувати знання. Ця унікальна функція надає їм розширені можливості вирішення проблем, адаптивність і глибшого відображення тонкощів своєї предметної області.

1.4 KBIA в EL

Структуру предмета наведену на рис. 2 можна розглядати як дорожню карту дослідження: три нитки дослідження - KR, IA, EL вивчаються від загального аналізу через проєктування до пропозицій з реалізації.

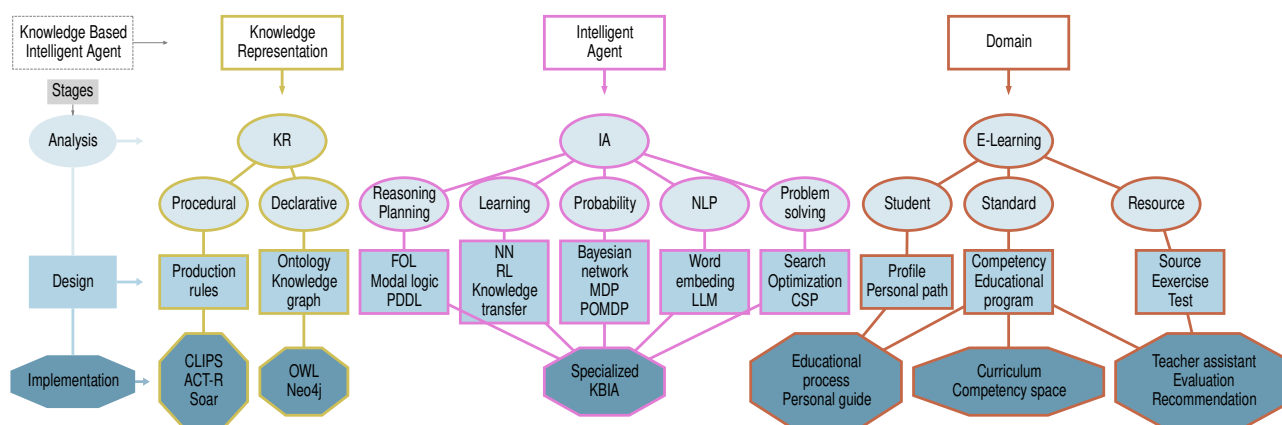


Рис. 2 Структура предмета дослідження

Типовий курс в системі EL університету складається з таких типів навчальних ресурсів: лекції оформлені як текстові документи або веб-сторінки; методичні матеріали до виконання лабораторних, практичних, самостійних, курсових робіт; завдання на виконання робіт з можливою параметризацією умов за варіантом; модульні та підсумковий тести; можливі додаткові посилання на корисні навчальні ресурси в мережі Інтернет. Процес навчання засобами EL зводиться до: завантаження ресурсів з системи EL; вивчення текстів лекцій, методичних матеріалів, завдань; виконання завдань та оформлення текстових звітів про результати; завантаження звітів в систему EL; проходження модульних та підсумкового тестів.

Недоліки: статичність та пласкість (недостатня гіпертекстовість) текстових навчальних ресурсів; лінійність процесу переходу від завдання до завдання; недостатня пов'язаність результатів виконання одних завдань для виконання інших; оцінювання та відгуки про результати несинхронізовані з переходами між завданнями; відсутність можливості самооцінювання для визначення досягнутого стану компетентностей; недостатня увага колективній роботі та кооперації.

Науково-методична рада Міністерства освіти і науки України розробила стандарти вищої освіти відповідно до галузей знань [3]. Ці стандарти представляють сукупність характеристик та вимог до навчання з конкретної спеціальності для досягнення певного освітнього рівня. Характеристики дають опис галузі та спеціальності: об'єкт; ціль навчання; теоретичний зміст предметної області; методи, методики та технології тощо. Вимоги складаються з детального опису компетентностей (інтегральних, загальних, спеціальних), які здобувач рівня освіти має набути.

Стандарти освіти оформлені у вигляді текстових документів, призначених для читання та використання людиною. Структурованої форми представлення стандартів освіти зручної для обробки комп'ютером не передбачено.

Пропонується на основі AOD розробити прототип програмного модуля, який буде можливо інтегрувати в існуючу систему EL для виправлення перелічених вище недоліків, що зробить процес навчання більш адекватним сучасному рівню розвитку інформаційних технологій. Програмний модуль має бути побудований як MAS, що складається із спеціалізованих KBIA. За відправну точку може бути взято методологію Prometheus AUML [1, 4] та архітектуру MAS, запропоновану в [5]. Важливим компонентом інтеграції програмних модулів KBIA в систему EL може бути підсистема NLP, яка дозволила б автоматизувати обробку текстових навчальних матеріалів та перетворення їх в KB [6].

На прикладі стандартів освіти [3] за спеціальностями факультету інформаційних технологій освітніх рівнів бакалавр та магістр розробити онтології відповідних предметних областей та необхідних компетентностей. Провести семантичне міждисциплінарне узгодження отриманих онтологій. Побудувати онтології наявних навчальних робочих програм, пов'язаних з вказаними спеціальностями, та узгодити їх з вже розробленими онтологіями. На основі отриманих результатів сформулювати пропозиції до проєктування та реалізації автоматизованої системи підтримки онтологій навчальних робочих програм в актуальному узгодженому стані з відповідними стандартами освіти та між собою. Вивчити можливість інтеграції такої автоматизованої системи в існуючу платформу EL та розповсюдження запропонованої технології на інші галузі та спеціальності.

2 МОДЕЛЮВАННЯ СИСТЕМИ

КВІА можна представити як систему, що складається з інших ІА. Для моделювання складних систем АІ ми будемо використовувати модульний підхід, коли різні спеціалізовані агенти або компоненти працюють разом для досягнення загальних цілей системи. В цій роботі КВІА буде розглядатися як система, де агенти вищого рівня координують та оркеструють діяльність інших ІА для досягнення цілей, вирішення задач та прийняття рішень.

2.1 КВІА як система агентів

На рис. 3 представлено КВІА як систему взаємодіючих ІА.

2.1.1 Характерні властивості моделі КВІА як системи агентів.

- Ієрархічна структура. КВІА можна змоделювати як систему агентів, де ІА вищого рівня контролюють, оркеструють і координують діяльність ІА нижчого рівня. Ця ієрархічна структура допускає модульність і спеціалізацію, де кожен ІА зосереджується на конкретних підзадачах.
- Спеціалізовані агенти. У системі КВІА окремі ІА можуть бути розроблені для виконання спеціальних завдань, таких як попередня обробка даних, вилучення ознак (feature extraction), міркування, навчання або взаємодія із зовнішніми системами.
- Співпраця. ІА співпрацюють з іншими агентами системи для об'єднання їхнього досвіду і знань. Спеціалізовані ІА можуть застосувати свої спеціальні знання щодо конкретної предметної області для вирішення складних задач.
- Мультиагентні системи. Системи КВІА можна будувати як багатоагентні системи, де різні ІА взаємодіють один з одним для досягнення спільних цілей. Кожен ІА може мати свої власні цілі та можливості.

- Розподілена обробка. Робоче навантаження системи можна розподілити між різними ІА, що забезпечує паралельну обробку та ефективне використання ресурсів.
- Динамічне розміщення. Залежно від вимог задачі, КВІА може динамічно розподіляти завдання між різними ІА, адаптуючись до мінливих умов і доступності ресурсів.
- Комунікація та координація. Агенти в системі КВІА спілкуються один з одним для обміну інформацією, діляться успіхами та невдачами для координації своїх дій.

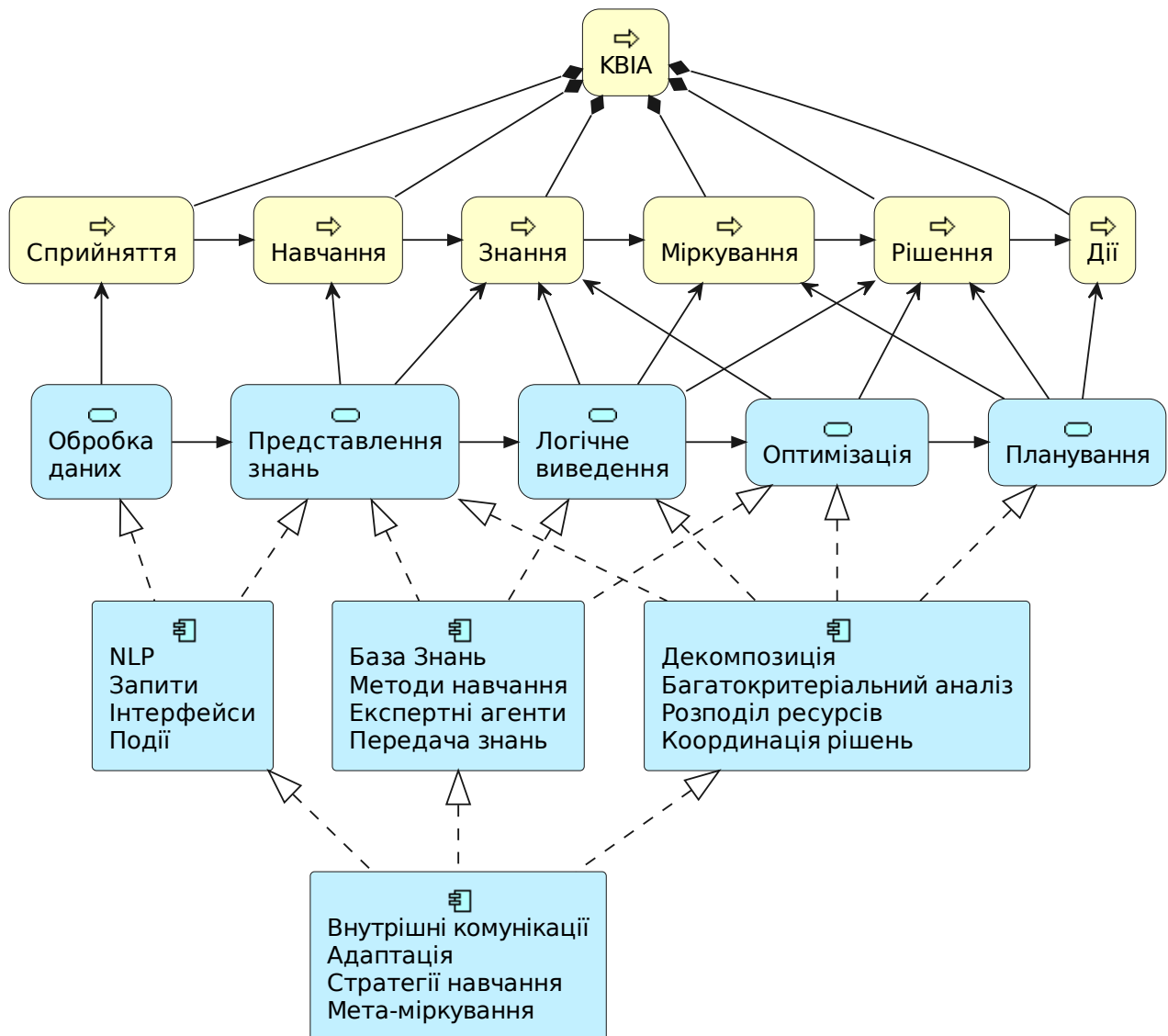


Рис 3. Система КВІА

- Гнучкість і масштабованість. Система KBIA може бути гнучкою та масштабованою, дозволяючи додавати або видаляти спеціалізовані ІА за потреби.
- Гібридні підходи. KBIA як система може поєднувати різні типи ІА, щоб використовувати їх взаємодоповнюючі переваги.

Таким чином, модель KBIA може бути представлена як композиція різних ІА, кожен з яких робить внесок у загальний процес вирішення задач і прийняття рішень. Цей модульний підхід забезпечує гнучкість, спеціалізацію, співпрацю та ефективне використання ресурсів, що робить системи KBIA здатними вирішувати складні задачі в різноманітних предметних областях.

2.1.2 Спеціалізація агентів.

У системі KBIA спеціалізовані агенти призначені для виконання конкретних завдань, компонентів або аспектів процесу вирішення проблем. Ці агенти співпрацюють у системі, щоб внести свій вклад і колективно досягти цілей системи.

Агент попередньої обробки даних відповідає за очищення, перетворення та нормалізацію вихідних даних перед їх використанням для аналізу. Він виконує такі завдання, як очищення даних, вилучення ознак і зв'язків.

KR агент зосереджується на представленні знань предметної області в структурованих форматах, таких як онтології, семантичні мережі або графи знань, забезпечує організацію та доступність бази знань.

Агент міркування виконує різні форми логічних міркувань, включаючи дедуктивні, індуктивні, абдуктивні та аналогічні міркування. Він робить висновки з наявних знань і підтримує прийняття рішень.

Агент навчання відповідає за отримання знань із даних за допомогою контрольованого або неконтрольованого навчання, адаптує нові знання на основі RL.

Агент планування створює послідовності дій та сценарії, які забезпечують досягнення конкретних цілей, розробляє плани з урахуванням наявних ресурсів, обмежень і потенційних результатів.

Комунікаційний агент забезпечує спілкування та співпрацю між різними ІА в системі, здійснює обмін інформацією та підтримує необхідні внутрішні протоколи.

Агент інтерфейсу керує взаємодією між системою КВІА та користувачами або зовнішніми системами, обробляє введення/виведення даних та запити користувачів, підтримує протоколи та інтерфейси зв'язку.

Агент оптимізації спеціалізується на методах оптимізації для пошуку оптимальних рішень у обмежених середовищах або середовищах з обмеженими ресурсами.

Агент стратегії навчання відповідає за мета-навчання, яке передбачає навчання тому, як вчитися, адаптує стратегії навчання інших агентів на основі характеристик нових задач.

Експертний агент втілює досвід у певній предметній області, надаючи спеціалізовані ідеї та знання, пов'язані з цією сферою.

Агент адаптації відстежує продуктивність системи та адаптує поведінку інших агентів у відповідь на зміну умов та вимог.

Агент мета-міркування контролює та координує процеси міркування інших агентів, приймаючи на основі контексту задачі рішення про те, коли та яку стратегію міркування використовувати.

Агент з прийняття рішень спеціалізується на прийнятті складних рішень, враховуючи численні критерії, компроміси та невизначеності для оптимального вибору.

Агент передачі знань здійснює передачу знань між різними задачами та предметними областями, дозволяючи системі використовувати попередній досвід для нових задач.

Наведені класи спеціалізованих агентів можна комбінувати різними способами, щоб створювати моделі систем КВІА, які будуть підходити для вирішення задач, прийняття рішень і управління знаннями конкретної предметної області. Кожен агент буде вносити свої унікальні сильні сторони та можливості для підвищення загальної продуктивності системи.

2.2 Моделі агентів

2.2.1 Модель BDI [7].

Модель “переконання-бажання-намір” (BDI) є когнітивною архітектурою, яка використовується для моделювання процесів прийняття рішень і міркування ІА. BDI забезпечує структуру для розуміння того, як агенти обробляють інформацію, встановлюють цілі, приймають рішення та виконують дії в складних і динамічних середовищах. Вона широко використовується в області AI, зокрема при розробці ІА і MAS.

Переконання відображають сприйняття світу ІА. Вони охоплюють знання агента про поточний стан середовища, включаючи інформацію про себе, інших агентів, об’єкти, події тощо. Переконання зазвичай представляють як набір суджень (propositions) або тверджень (statements).

Бажання представляють цілі або завдання агента. Вони вказують на те, чого агент хоче досягти або зробити в середовищі. Бажання можуть варіюватися від короткострокових цілей, таких як виконати певну дію, до довгострокових цілей, таких як максимізація корисності або досягнення певного результату.

Наміри охоплюють заплановані чи передбачувані дії ІА для досягнення своїх бажань. Агент формує наміри, обираючи напрям дій, який, на його думку,

допоможе досягти його цілей. Наміри керують прийняттям рішень і поведінкою агента, встановлюючи конкретний план дій.

Складові процесу BDI.

- Перегляд переконань (belief revision). ІА постійно оновлюють і переглядають свої переконання на основі нової інформації та спостережень середовища.
- Вибір бажань (desire selection). ІА визначають пріоритети своїх бажань на основі своїх поточних переконань і уподобань. Бажання можуть суперечливими, тому агенти мають вибирати, яким з них слідувати.
- Формування наміру (intention formation). ІА приймають рішення про намір, вибираючи план або курс дій, які, на їхню думку, приведуть до досягнення їхніх бажань.
- Виконання плану (plan execution). ІА виконують обрані ними плани для реалізації своїх намірів і досягнення своїх цілей.
- Моніторинг та перегляд планів (plan monitoring and reconsideration). ІА контролюють своє середовище та виконання своїх планів. Якщо відбуваються несподівані події, агенти можуть переглянути свої наміри та відповідно сформувані нові плани.

BDI забезпечує структурований підхід до моделювання поведінки ІА. Вона підходить для сценаріїв, коли агентам потрібно міркувати про складні та невизначені ситуації, приймати рішення на основі своїх переконань і бажань та адаптуватися до мінливого середовища.

Системи на основі BDI зазвичай застосовуються при моделюванні MAS, роботів, інтелектуальних помічників та систем підтримки прийняття рішень. Архітектура дозволяє розробляти агентів, які проявляють цілеспрямовану поведінку, ставлять пріоритети цілям і приймають відповідні контексту рішення.

BDI черпає натхнення у філософії, зокрема в роботах таких філософів, як Майкл Братман (Michael Bratman), і його теорії практичного міркування (theory of practical reasoning). Ідеї Братмана про те, як люди приймають рішення, формують наміри та переслідують цілі, вплинули на розвиток архітектури BDI [7 p.15]. Рання концептуалізацію архітектури BDI можна прослідкувати у 1980-х роках в роботах таких дослідників, як Кейт Декер (Keith Decker), Ананд Рао (Anand S. Rao) та Майкл Джорджеф (Michael Georgeff). Вони досліджували способи моделювання поведінки ІА у MAS.

Одним із новаторських внесків в архітектуру BDI стала модель «Системи практичного міркування» (Practical reasoning systems - PRS), розроблена Майклом Джорджефом і Емі Ланскі (Amy Lansky) наприкінці 1980-х років. PRS запропонувала структуру BDI для представлення практичних міркувань і поведінки агентів, заснованої на намірах [7 p.22]. BDI набула популярності в 1990-х роках, коли її все частіше почали застосовувати при розробці MAS. Її визнали ефективним підходом для моделювання поведінки агентів у складних середовищах, де агентам необхідно взаємодіяти та співпрацювати.

Для підтримки архітектури BDI були розроблені мови програмування та фреймворки. Одним із яскравих прикладів є сімейство мов AgentSpeak, включаючи AgentSpeak(L) та її варіанти. Ці мови забезпечили спосіб вираження поведінки та міркувань агентів на основі BDI [7 p.235].

Протягом багатьох років архітектура BDI широко вивчалася та застосовувалася в різних областях, включаючи багатоагентні системи, робототехніку, інтелектуальні системи навчання та системи підтримки прийняття рішень. Це сприяло прогресу в моделюванні складної поведінки, співпраці та прийнятті рішень. BDI продовжує розвиватися, враховуючи результати досліджень у таких сферах, як CS, AI та психологія. Досліджуються розширення та варіації моделі BDI, щоб знайти нові можливості застосування.

2.2.2 Модель АСТ-R [8].

АСТ-R (Adaptive Control of Thought - Rational) — це когнітивна архітектура, призначена для моделювання людського пізнання та поведінки. Розроблена Джоном Андерсоном (John Anderson) та його колегами з Університету Карнегі-Мелона (Carnegie Mellon University). АСТ-R пропонує основу для розуміння та моделювання когнітивних процесів, включаючи сприйняття, пам'ять, увагу, навчання, вирішення задач і прийняття рішень. Вона має на меті охопити основні механізми людського пізнання та поведінки, що робить її цінним інструментом для CS, досліджень AI та створення КВІА.

Поняття та особливості АСТ-R.

- Декларативна та процедурна пам'ять (declarative and procedural memory). Розділяє пам'ять на два компоненти: декларативну та процедурну пам'ять. Декларативна пам'ять зберігає факти, знання та концепції, тоді як процедурна пам'ять зберігає правила та процеси для виконання завдань.
- Продукційні правила (production rules). Використовує продукційні правила для представлення когнітивних процесів. Ці правила складаються з умов і дій, які дозволяють моделювати процеси прийняття рішень та поведінку агента. Правила АСТ-R пов'язані з процедурною пам'яттю.
- Розбиття на фрагменти та навчання (chunking and learning). Включає в себе фрагментування, механізм для організації та представлення інформації. Розбиття на фрагменти дозволяє ІА фіксувати та повторно використовувати шаблони інформації, що сприяє більш ефективній когнітивній обробці, містить механізми для навчання та отримання нових фрагментів.
- Буфери (buffers). Використовує буфери як тимчасові області зберігання інформації під час когнітивних задач. Буфери представляють різні аспекти

пізнання, такі як сенсорний вхід (sensory input), робоча пам'ять (working memory) і інформація про цілі (goal information).

- Структура цілей (goal structure). Відображає цілеспрямовану поведінку ІА. Цілі представляють бажані результати когнітивних задач, які агент прагне досягти шляхом виконання відповідних продукційних правил.

Фокус АСТ-R на когнітивних процесах роблять його цінним інструментом для створення інтелектуальних систем, які імітують когнітивні процеси, включаючи сприйняття, пам'ять, увагу, навчання та прийняття рішень.

2.2.3 Модель Soar [9].

Soar — це когнітивна архітектура, яка служить обчислювальною моделлю людських когнітивних процесів, міркувань, вирішення задач і прийняття рішень. Він був розроблений Джоном Лердом (John Laird), Аленом Ньюеллом і Полом Розенблумом (Paul Rosenbloom) з Мічиганського університету (University of Michigan). Протягом багатьох років вона постійно вдосконалювалася та розширювалася. Метою розробки було охоплення механізмів, які лежать в основі інтелектуальної поведінки, що робить його цінним інструментом для моделювання людського пізнання та створення інтелектуальних агентів на основі знань [10].

Характеристики та концепції Soar.

- Уніфікована структура (unified framework). Має на меті забезпечити уніфіковану структуру для моделювання широкого спектру когнітивних процесів, включаючи сприйняття, навчання, пам'ять, міркування та прийняття рішень.
- Символічна та субсимволічна обробка (symbolic and subsymbolic processing). Використовує як символічну, так і субсимволічну обробку для представлення та маніпулювання знаннями: символічні структури (такі як

продукції) для представлення когнітивних процесів високого рівня та субсимволічні коннекціоністські (connectionist) механізми для навчання.

- Продукційні системи (production systems). Базується на концепції продукційні систем, які складаються з продукційних правил, що представляють моделі міркувань, прийняття рішень і поведінки. Продукції мають умови (передумови) і дії (висновки), що дозволяє використовувати їх для моделювання когнітивних процесів.
- Робоча пам'ять (working memory). Зберігає поточний стан ІА, включаючи інформацію про середовище, цілі та відповідні факти. Продукції зіставляються з вмістом робочої пам'яті для запуску дій.
- Пошук у проблемному просторі (problem-space search). Використовує пошук у проблемному просторі, щоб дослідити потенційні послідовності дій і зробити висновок про порядок застосування продукцій, що дозволить досягти бажаних цілей.
- Навчання (learning). Містить механізми для навчання, включаючи фрагментацію (навчання абстракціям вищого рівня) і RL. Механізми навчання дозволяють системі отримувати нові знання та покращувати свою ефективність.
- Підцілі (subgoaling). Підтримує створення підцілей та ієрархічних планів, що дозволяє системі розбивати складні завдання на керовані підзадачі.

Soar забезпечує комплексну структуру для моделювання складних когнітивних процесів, що робить його придатним для створення КВІА, які можуть демонструвати поведінку подібну до людської. Вона долає розрив між символічним міркуванням і субсимволічним навчанням.

2.3 Моделі представлення знань

В попередньому підрозділі ми розглянули моделі, які дають можливість представляти KBIA фокусуючи увагу на їхній поведінці та взаємодії з іншими агентами та зовнішнім середовищем. В цьому підрозділі зосередимо увагу на сучасних підходах до організації KB, які можуть бути використані при проектуванні IA.

2.3.1 Онтології [11].

Онтології забезпечують формальний і структурований спосіб представлення знань предметної області за допомогою класів, властивостей та індивідів. Базуючись на дескриптивній логіці (description logic - DL), вони підтримують складне моделювання знань і можуть виконувати задачі міркування.

Використання онтологій як представлення знань у KBIA має кілька переваг завдяки їх структурованому, семантичному та сумісному характеру. Вони забезпечують стандартизовану структуру для збору та організації знань, дозволяючи агентам міркувати, робити висновки та ефективно спілкуватися.

Детальніше про особливості представлення знань через онтології.

- Забезпечують семантичне представлення знань предметної області шляхом визначення понять, зв'язків, атрибутів і обмежень. Це дозволяє агентам відображати значення та контекст інформації.
- Пропонують структурований спосіб організації та категоризації знань, що дає можливість агентам ефективно орієнтуватися в базі знань і та робити запити до неї. Поняття та зв'язки можуть бути організовані ієрархічно або за допомогою інших формальних структур.
- Сприяють спільному розумінню концепцій предметної області серед агентів, людей та інших систем, орієнтованих на знання. Це зменшує неоднозначність і забезпечує цілісну інтерпретацію знань.

- Надають загальний словник (common vocabulary) взаємодії, який агенти та інші системи можуть використовувати для спілкування та обміну інформацією. Це особливо корисно в багатоагентних системах та гетерогенних середовищах.
- Дозволяють агентам виконувати автоматизовані міркування та робити висновки. Агенти можуть виводити нові знання, перевіряти узгодженість і ідентифікувати неявні зв'язки за допомогою методів логічного міркування.
- Пропонують засоби комплексного моделювання предметної області, охоплюючи різні аспекти, включаючи таксономії, атрибути, ролі та події. Це сприяє цілісному розумінню предмету моделювання.
- Підтримують обмін знаннями та їх повторне використання в різних застосуваннях. Чітко визначена онтологія може служити основою для розробки системи КВІА предметної області.
- Можуть бути модульними та розширюваними, що робить їх масштабованими при отримання нових знань. Зміни або оновлення в предметній області можна відобразити шляхом модифікації або розширення онтології.
- Покращують NLP, надаючи структуровану основу для зв'язку лінгвістичних концепцій з їхніми семантичними відповідниками. Це може розширити можливості в розумінні та генерації артефактів мови.
- Допомагають інтегрувати різноманітні джерела даних, надаючи загальну модель для відображення різних структур даних в уніфіковане семантичне представлення.

Розробка онтології потребує досвіду моделювання предметної області та ретельного розгляду концепцій, зв'язків і потенційних сценаріїв використання. Процес розробки онтології може зайняти багато часу, а підтримка узгодженості

протягом всього часу застосування має вирішальне значення. Вибір відповідної мови онтології (наприклад, RDF, OWL) та відповідних інструментів є важливим для досягнення бажаного рівня виразності та масштабованості. Інтеграція онтологій з іншими методами KR, такими як продукційні правила, може бути використана для охоплення ширшого кола знань.

Одним з шляхів вирішення складнощів створення та підтримки цілісності бази знань на основі онтологій може бути використання підходу формування онтології через навчання.

Процес навчання онтології передбачає отримання агентом відповідної інформації про предметну область та представлення її як знання про концепції, зв'язки, атрибути та обмеження. Це дозволяє агенту створити структуроване відображення предметної області, уможливлуючи краще міркування, виведення висновків та прийняття рішень.

В процесі формування та підтримки онтології шляхом навчання KBIA має виконувати такі дії:

- витяг концепцій (concept extraction) з текстових даних, документів предметної області або взаємодії з користувачем; методи NLP можуть ідентифікувати ключові терміни та фрази, які представляють поняття в моделі;
- виявлення зв'язків (relationship discovery) між поняттями, аналізуючи шаблони в даних або вивчаючи їхню взаємодію; наприклад, якщо певні поняття часто згадуються разом, то можна зробити висновок про зв'язок між ними;
- виведення загальних понять з конкретних фактів за допомогою алгоритмів індуктивного навчання (inductive learning); наприклад, маючи набір екземплярів можна узагальнити спільні атрибути для концепцій вищого рівня;

- групування схожих екземплярів у кластери методами кластеризації та класифікації (clustering and classification) для подальшої ідентифікації понять; алгоритми класифікації можуть призначати екземпляри попередньо визначеним поняттям;
- семантичний аналіз (semantic analysis) текстових або структурованих даних для виявлення семантичної подібності та групування пов'язаних понять;
- активне навчання (active learning) шляхом взаємодії з користувачами або експертами для уточнення відображення предметної області; може включати питання, пошук роз'яснень або запит на оцінку елементів сформованої онтології;
- навчання через вивчення поведінки користувачів (learning from user behavior) шляхом спостереження за тим, як користувачі взаємодіють з агентом; витяг можливих висновків про зв'язки та поняття на основі поведінки користувачів, запитів і вподобань;
- інтеграція зовнішніх джерел даних (integration of external data sources) таких як сховища даних, бази знань тощо, щоб розширити онтологію актуальною та релевантною інформацією;
- вирівнювання та злиття онтологій (ontology alignment and merging) шляхом взаємодії з іншими агентами або системами для узгодження навченої онтології з існуючими онтологіями для забезпечення спільного представлення та інтерпретації.

При формуванні та підтримці онтології шляхом навчання, потрібно враховувати, що якість і точність такої онтології залежать від якості даних, алгоритмів навчання та процесів перевірки. Крім того, може знадобитися людський нагляд і експертний аналіз для підтвердження, що навчена онтологія

узгоджується зі знаннями предметної області та точно представляє семантику моделі.

2.3.2 Ймовірнісні та нечіткі моделі.

Ймовірнісні моделі (probabilistic models) — це клас моделей представлення знань, які використовуються в КВІА для обробки невизначеності, неточності та випадковості в процесах відображення та міркування. Ці моделі особливо корисні в ситуаціях, коли достовірність інформації не є абсолютною і коли необхідно міркувати про ймовірності та вірогідності. Ймовірнісні моделі дозволяють агентам приймати обґрунтовані рішення в невизначених середовищах [12 p.510].

Ймовірнісні моделі передбачають призначення ймовірностей різним подіям або результатам, де їх розподіл (probability distribution) представляє ймовірність кожної можливої події в ситуації, коли агент не має повної інформації, забезпечуючи спосіб кількісного визначення ступеня невизначеності.

Мережі Байєса. Мережі Байєса, також відомі як мережі переконань (belief networks) або моделі ймовірнісних графів (probabilistic graphs models), є поширеною формою ймовірнісних моделей. Вони використовують орієнтовані ациклічні графи (directed acyclic graphs - DAG) для представлення зв'язків між змінними та таблиці умовних ймовірностей для кількісного визначення залежностей [12 p.513].

Моделі Маркова. Моделі Маркова фіксують ймовірності переходів між станами системи в часі. Приховані моделі Маркова (hidden Markov models - НММ) часто використовуються в моделюванні предметної області, коли справжній стан системи неможливо спостерігати безпосередньо [12 p.566].

Точність і надійність ймовірнісних моделей значною мірою залежать від якості даних, які використовуються для оцінки ймовірностей. Розробка та

підтримка великих імовірнісних моделей може бути складною через потребу в точних оцінках ймовірності та ефективних алгоритмах міркування. Обчислення міркувань за допомогою імовірнісних моделей може мати значну інтенсивність, особливо для великомасштабних моделей. Розробка ефективних імовірнісних моделей потребує досвіду предметної області для точної оцінки ймовірностей та моделі залежностей.

Нечіткі моделі (fuzzy models) — це клас моделей представлення знань, які використовуються в КВІА для обробки неточностей і невизначеностей у процесах представлення та міркування. Нечітка логіка надає спосіб представляти та міркувати про концепції, які не мають чітких двійкових значень істинності чи приналежності до множини, а мають ступінь істинності чи приналежності в діапазоні від 0 до 1. Нечіткі моделі особливо корисні в ситуаціях, коли приймаються рішення засновані на розпливчастій або неповній інформації [13].

Ключові поняття нечітких моделей:

- функції приналежності до множини (membership functions) визначають ступені членства значеннями у межах лінгвістичної змінної; функції можуть описувати ступінь приналежності значення до певної категорії;
- нечіткі множини (fuzzy sets) узагальнюють традиційні чіткі множини, дозволяючи елементам частково належати до кількох множин; ступінь приналежності до множини визначає ступінь членства;
- оператори нечіткої логіки, такі як І, АБО та НЕ, розширені для обробки нечітких множин використовують значення приналежності для виконання операцій над нечіткими множинами;
- нечіткі правила виражають зв'язки між нечіткими множинами за допомогою лінгвістичних змінних і пар “умова-дія”; нечітке виведення висновків використовує ці правила для прийняття нечітких рішень;

- дефазифікація — це процес перетворення нечітких множин і ступенів приналежності до них в чіткі значення або дії.

Розробка ефективних функцій приналежності та лінгвістичних змінних потребує значного досвіду предметної області, щоб адекватно відобразити невизначеності у моделі. Керування нечіткими правилами можуть бути складним, особливо для великих систем із багатьма правилами та змінними. Нечіткий логічний висновок може бути вимогливим до обчислень, особливо під час роботи зі складними нечіткими множинами та правилами.

2.3.3 Нейронні мережі.

Нейронні мережі, також відомі як коннекціоністські моделі (connectionist models), — це клас моделей KR, які також можуть бути застосовані в системах KBIA. Ці моделі засновані на структурі та функціях, які імітують роботу людського мозку. Вони складаються з взаємопов'язаних одиниць обробки (нейронів), які можуть адаптуватися до вхідних даних шляхом налаштування параметрів. Коннекціоністські моделі особливо корисні для вирішення завдань, які включають розпізнавання образів, навчання з даних і складні відображення між входами та виходами [14].

Основні поняття нейронних мереж.

- Нейрони та активація (neurons and activation). Штучні нейрони імітують поведінку біологічних нейронів. Кожен нейрон отримує вхідні сигнали, обробляє їх за допомогою функції активації та створює вихідний сигнал.
- Ваги та зв'язки (weights and connections). Нейрони з'єднані один з одним через зважені зв'язки. Ці ваги визначають силу впливу виходу одного нейрона на вхід іншого.
- Шари (layers). Коннекціоністські моделі можуть мати кілька шарів нейронів. Вхідний рівень отримує зовнішні вхідні дані, приховані шари

обробляють проміжні представлення, а вихідний рівень генерує кінцевий вихід.

- Алгоритми навчання (learning algorithms) налаштовують параметри зв'язку між нейронами. Наприклад, алгоритм зворотнього поширення коригує ваги на основі різниці між прогнозованими та фактичними результатами.
- Мережі прямого зв'язку та рекурентні мережі (feed-forward and recurrent networks). Мережі прямого зв'язку обробляють інформацію в одному напрямку, від входу до виходу. Рекурентні мережі мають зворотні зв'язки між шарами, що дозволяє моделювати часову обробку та пам'ять.

Для ефективного навчання нейронні мережі вимагають великих обсягів даних. Недостатні або зміщені дані можуть призвести до неоптимальної продуктивності. Існує проблема надмірного налаштування параметрів, коли модель може запам'ятовувати навчальні дані та погано працювати з невідомими. Нейронні мережі вважаються чорними ящиками, що означає, складність пояснення та обґрунтування прийнятих рішень. Навчання великих нейронних мереж може потребувати інтенсивних обчислень, вимагаючи потужного апаратного забезпечення та ефективних методів оптимізації. Вибір правильної архітектури, функцій активації та гіперпараметрів може потребувати експериментування та евристичного дослідження.

2.4 Моделі планування та прийняття рішень

КВІА використовують моделі планування та прийняття рішень для ефективного вирішення задач. Ці моделі потрібні агентам, щоб формулювати стратегії та політики, робити обґрунтований вибір та досягати бажаних цілей.

2.4.1 Марківські процеси прийняття рішень.

Марківські процеси прийняття рішень (MDP) та частково спостережувані Марківські процеси прийняття рішень (POMDP) - це моделі, які

використовуються в системах KBIA для планування та прийняття рішень у невизначених середовищах та забезпечують структуровану основу для прийняття агентами оптимальних або майже оптимальних рішень, враховуючи невизначеності та ймовірності переходів між станами.

У MDP агент взаємодіє з середовищем протягом послідовних кроків в часі. На кожному кроці агент приймає рішення (вибирає дію) щодо переходу з поточного стану в новий стан. MDP визначаються п'ятьма компонентами [12 р.610].

- Стани (States - S). Скінченна множина можливих станів, у яких може перебувати агент і середовище.
- Дії (Actions - A). Скінченна множина дій, які може виконувати агент.
- Імовірності переходу (Transition probabilities - T). Розподіл ймовірностей переходу з одного стану в інший за певної дії.
- Винагороди (Rewards - R). Негайна винагорода, яку отримує агент за певну дію в певному стані.
- Коефіцієнт знижки (Discount factor - γ). Параметр, який представляє перевагу, яку агент надає негайним винагородам перед майбутніми винагородами.

Агенти прагнуть знайти політику, яка відображає стани в дії (mapping from states to actions) та максимізує очікувану кумулятивну винагороду з врахуванням знижки (expected cumulative discounted reward). Вони можуть використовувати такі алгоритми, як динамічне програмування (dynamic programming), ітерації по значеннях (value iteration) та ітерації політиках (policy iteration).

POMDP розширює MDP на ситуації, коли спостереження агента є невизначеними або неповними. У POMDP агент безпосередньо не спостерігає справжній стан середовища, але отримує спостереження, які ймовірно

пов'язані з основним станом. POMDP складаються з таких компонентів [12 р.645].

- Стани (States - S). Те саме, що й в MDP - представляють можливі базові стани агента та середовища.
- Дії (Actions - A). Те саме, що й в MDP - представляють доступні дії агента.
- Спостереження (Observations - O). Скінченна множина спостережень, які агент може отримати на основі основного стану та дій агента.
- Імовірності переходу (Transition probabilities - T). Те саме, що й в MDP - розподіл ймовірностей переходу з одного стану в інший за певної дії.
- Імовірності спостереження (Observation probabilities - Z). Розподіл ймовірностей отримання спостереження з урахуванням певного основного стану та дії.
- Винагороди (Rewards - R). Те саме, що й в MDP - негайна винагорода, яку отримує агент за певну дію в певному стані.
- Коефіцієнт знижки (Discount factor - γ). Те саме, що й в MDP - параметр, який представляє перевагу, яку агент надає негайним винагородам перед майбутніми винагородами.

POMDP вводить концепцію станів переконань (belief states), які є розподілами ймовірностей основних станів, враховуючи спостереження агента. Рішення полягає в пошуку політики, яка відображає стан переконань в дії, враховуючи невизначеність спостережень, що вимагає виведення висновків шляхом міркування над станами переконань (reason over belief states). Для отримання наближених до оптимальних рішень використовуються такі підходи, як точкова ітерація по значеннях (point-based value iteration) і фільтри часток (particle filters).

2.4.2 Байєсівські мережі

Байєсівські мережі (BN), також відомі як байєсівські мережі переконань (Bayesian belief networks) або моделі імовірнісних графів (probabilistic graphical models), є універсальним і широко використовуваним інструментом представлення знань і міркування для планування та прийняття рішень у системах КВІА. BN дозволяють агентам моделювати та міркувати про невизначеність зв'язків та залежності між змінними ймовірнісним способом [12 р.510].

Поняття BN.

- Орієнтований ациклічний граф (DAG). BN представляється як DAG, де вузли відповідають випадковим змінним, а спрямовані ребра відображають імовірнісні залежності.
- Вузли (випадкові змінні) (nodes). Кожен вузол графа представляє випадкову змінну, яка може бути спостережуваною або ні.
- Таблиці умовної ймовірності (conditional probability tables - CPT). CPT кожного вузла визначає умовні ймовірності, заданого його батьками на графі.
- Теорема Байєса (Bayes' theorem). В BN використовується теорема Байєса для оновлення переконань та виведення висновків на основі спостережених свідчень (observed evidence).
- Алгоритми розповсюдження (propagation algorithms). Для виконання ймовірнісного висновку в BN використовуються такі алгоритми, як усунення змінних (variable elimination) і з'єднання дерев (junction tree), що дозволяє ефективно оновлювати переконання (update beliefs).

- Навчання (learning). BN можна навчатися на даних за допомогою таких методів, як оцінка параметрів (parameter estimation) і вивчення структури (structure learning).

BN надають системам KBIA можливість міркувати про невизначеність і приймати обґрунтовані рішення на основі ймовірностей та моделювати причинно-наслідкові зв'язки, допомагаючи агентам зрозуміти вплив різних факторів на результати. Існують ефективні алгоритми розповсюдження для обчислення ймовірнісного висновку навіть у великих мережах. Моделі BN особливо цінні при роботі в ситуаціях, коли ймовірнісні залежності та невизначеність відіграють значну роль у прийнятті рішень.

2.4.3 RL.

RL - це підхід, який може бути застосований в системах KBIA для планування та прийняття рішень у динамічних і невизначених середовищах. RL дозволяє агентам вивчати оптимальні або близькі до оптимальних політики та стратегії через взаємодію з середовищем, максимізуючи кумулятивні винагороди. Цей підхід є важливим в моделях, для яких явних знань або заздалегідь визначених правил недостатньо, і тому агентам необхідно адаптувати свої політики та стратегії на основі зворотного зв'язку з середовищем [12 p.830, 15].

Концепції RL.

- Агент і середовище (Agent and Environment). Агент взаємодіє з середовищем: він виконує дії, а середовище реагує переходом до нових станів і наданням зворотного зв'язку у вигляді винагороди.
- Стани (States). Представляють умови середовища в момент часу. Вони інкапсулюють усю релевантну інформацію, яка потрібна агенту для прийняття рішень.

- Дії (Actions). Вибір, зроблений агентом для впливу на середовище. Мета агента - сформувані політики та стратегії, які визначатимуть найкращі дії в кожному стані.
- Винагороди (Rewards). Числові значення, які вказують на бажаність дії агента в певному стані. Агенти прагнуть максимізувати сукупні винагороди, які вони отримують в часі.
- Політика (Policy). Стратегія, яка відображає стани в дії. Агент формує політики для прийняття рішень, які оптимізують його довгострокову винагороду.
- Функція цінності (Value Function). Оцінює очікувану кумулятивну винагороду, яку агент може отримати від певного стану, дотримуючись певної політики.
- Q-функція (Q-Function). Відома як функція дія-цінність, оцінює очікувану кумулятивну винагороду, яку агент може отримати від певної пари стан-дія, дотримуючись певної політики.

Методи які можуть бути використані агентом в процесі RL.

- Q-навчання (Q-Learning). Безмодельний алгоритм, який формує оптимальну Q-функцію шляхом дослідження (exploration) та використання (exploitation) дій.
- Методи градієнта політики (Policy Gradient Methods). Безпосередньо формують параметризовану політику, яка максимізує очікувану кумулятивну винагороду.
- Методи актора та критика (Actor-Critic Methods). Поєднують підходи на основі політики (policy-based) та на основі значень цінності (value-based), використовуючи актора для оновлення політики (update policies) та критика для оцінки функцій цінності (estimate value functions).

- Глибоке RL (Deep RL). Поєднує алгоритми RL із глибокими нейронними мережами для роботи з просторами станів великої розмірності та складними середовищами.

RL дозволяє системам KBIA адаптуватися до мінливого середовища та формувати оптимальні стратегії не маючи чітких правил та взаємодіючи з середовищем, що робить його придатним для сценаріїв, де бракує детальних знань. Такий підхід дозволяє вирішувати складні завдання прийняття рішень із великим простором станів та невизначеною динамікою.

При моделюванні агентів, які використовують методи RL, потрібно враховувати такі обставини: знаходження балансу між дослідженням (exploration) і використанням (exploitation); використання великої кількості взаємодій із середовищем для формування ефективної політики; розробка відповідних функцій винагороди може мати вирішальне значення для досягнення бажаної поведінки; неоптимальні або небезпечні політики можуть бути сформовані раніше, ніж будуть знайдені оптимальні або наближені до оптимальних, що може бути загрозою у критично важливих для безпеки програмах.

3 РОЗРОБКА СИСТЕМИ

Предметною областю застосування розглянутих в попередніх розділах підходів та методів будуть інтелектуальна система навчання (ITS) та система управління навчанням (LMS).

ITS поєднує AOD з онтологіями для створення персоналізованого навчального середовища. Онтології використовуються для представлення знань предметної області та для створення та супроводу моделі студента, що дозволяє генерувати персоналізований зміст процесу навчання. Агентно-орієнтовані підходи дозволяють підтримувати зворотний зв'язок, оцінювати знання студентів і адаптувати зміст та темп навчання враховуючи особливості студента.

LMS використовує KBIA для забезпечення кращого управління курсами та персоналізованим навчанням. Агенти можуть допомагати викладачам виконувати адміністративні задачі, формувати персоналізований контент, аналізувати дані про успішність студентів тощо.

Надалі ми будемо розглядати розроблювану систему як органічне поєднання функцій ITS та LMS з метою автоматизації інтелектуальної праці викладачів та адміністраторів навчального процесу.

3.1 Предметна область та архітектура рішення

На рис. 4 зображено джерела, функції та актори навчального процесу з точки зору розроблюваної системи.

Система має забезпечувати інтелектуальну автоматизацію таких функцій.

- Розробка навчальних програм. Моделювання структури та змісту комплексних та структурованих програм. Ієрархічна організація предметів, тем та цілей навчання.

- Виявлення ресурсів та джерел. Класифікація та впорядкування освітніх ресурсів, таких як підручники, статті, аудіо, відео та онлайн-курси. Зручний пошук відповідних навчальних матеріалів.
- Моделювання знань, навичок та вподобань студентів. Оцінка сильних та слабких сторін студента, вподобань у навчанні. Визначення персоналізованих шляхів навчання. Рекомендації конкретних ресурсів та дій на основі індивідуального профілю студента.
- Відображення компетентностей. Представлення компетентностей та навичок, які мають набуті студентами. Оцінка прогресу студентів та визначення областей, де вони потребують вдосконалення.
- Адаптація навчання. Приведення рівня складності та змісту навчальних завдань до поточних знань і здібностей студента.
- Стандартизація та сумісність. Забезпечення безперервної взаємодії та обміну даними між різними навчальними системами та платформами.
- Програми професійного розвитку викладачів. Моделювання навичок та компетентностей, необхідних для ефективного навчання. Профілі викладачів.
- Міждисциплінарна взаємодія. Відображення концепцій і тем, які охоплюють кілька предметних областей та дисциплін.
- Віртуальні лабораторії та середовища. Створення віртуальних навчальних середовищ та симуляцій для проведення безпечних експериментів, практичних та лабораторних робіт. Проведення навчальних ігор.
- Навчання через співпрацю. Підтримка спільного навчання, колективних обговорень, взаємного оцінювання та групових проєктів. Координація діяльності груп.

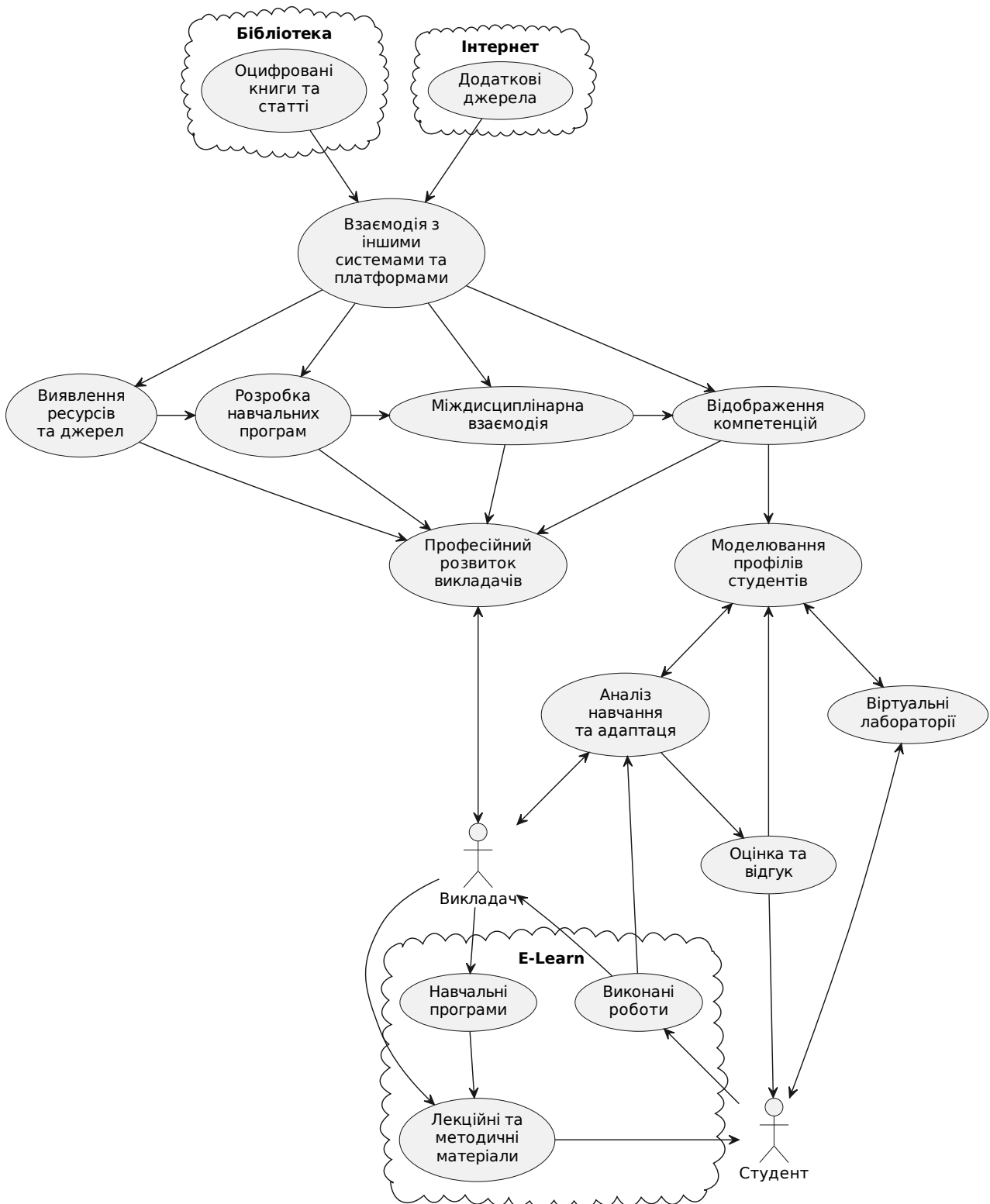


Рис. 4 Сценарії функціонування системи

3.2 Компоненти системи та їхня взаємодія

На рис.5 представлено принципову схему взаємодії компонентів системи.

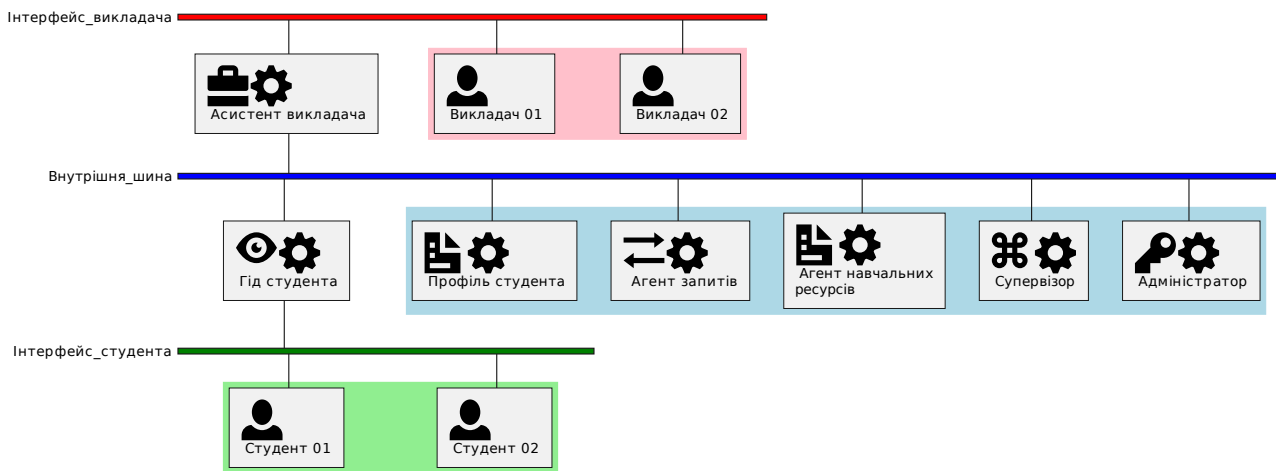


Рис. 5 Агенти та їхня взаємодія

Інтерфейси викладача та студента. Забезпечують роботу викладача та студента з системою через агентів Асистент Викладача або Гід Студента відповідно. Реалізується як web-, mobile-, desktop-додаток. Основні функції: перевірка авторизації, відображення системних сповіщення, ввід та контроль інформації, обмін даними з асистентом або гідом.

Внутрішня Шина - сервіс обміну даними між агентами. Реалізується засобами системи обміну повідомленнями (messaging system), наприклад, таким як NATS [16]. Основні функції: передача даних між агентами в режимі запитів та публікація/підписка (publish/subscribe), підтримка розподіленої обробки (distributed processing) та кластеризації (clustering).

Асистент Викладача - виконує персоналізований доступ викладача до функцій системи. Реалізується як агентно-орієнтований сервіс з зовнішнім API та доступом до Внутрішньої Шини системи. API має підтримувати широке коло платформ та мов програмування. Основні функції: взаємодія з Профілем Студента, Супервізором, Агентом Запитів, Агентом Навчальних Ресурсів та

Адміністратором для організації та забезпечення навчального процесу, спостереження та повідомлення викладача про виконані дії підписаних студентів, передача запитів/відповідей між студентами та викладачем, підтримка в організації курсів, модулів, тестів завдань.

Гід Студента є персональним провідником студента в EL. Реалізується як агентно-орієнтований сервіс з зовнішнім API та доступом до Внутрішньої Шини системи. API також має підтримувати широке коло платформ та мов програмування. Основні функції: взаємодія з Профілем Студента, Супервізором, Агентом Запитів, Агентом Навчальних Ресурсів та Адміністратором для планування та ведення навчального процесу студента, формування індивідуальних рекомендацій, передача запитів/відповідей між студентами та викладачем.

Профіль Студента підтримує індивідуальні дані студента у вигляді KB. Вони включають інформацію про студента, його курси, плани, виконані завдання та тести, оцінки, визначає рух студента в просторі компетентностей. Забезпечує даними Гіда Студента, Супервізора, Асистент Викладача через Агента Запитів. Реалізується як онтологія у графовій базі даних.

Агент Навчальних Ресурсів підтримує в актуальному, інтегрованому та цілісному стані KB про всі ресурси пов'язані з навчальним процесом: стандарти, компетентності, освітні та робочі програми, методичні матеріали, тести, завдання тощо. Через Агента Запитів забезпечує даними всіх інші компоненти системи. Реалізується як онтологія у графовій базі даних.

Агент Запитів реалізується як агентно-орієнтований сервіс, що забезпечує уніфікований доступ до KB Профілю Студента та Агента Навчальних Ресурсів. Основні функції: забезпечує однаковий інтерфейс до різних представлень KB, підтримує кеш для типових запитів.

Супервізор - агентно-орієнтований сервіс, який виконує роль координатора та планувальника. Основні функції: підтримка розкладів, графіків та календарів, повідомлення про настання подій.

Адміністратор - системний агентно-орієнтований сервіс, забезпечує ведення та авторизацію користувачів, налаштування параметрів системи та компонентів, ведення та аналіз системних журналів, формування повідомлень системним адміністраторам.

3.3 Реалізація KB

Для представлення KB як онтологічного графа знань обрано підходи запропоновані в [17] засобами графової системи баз даних Neo4j [18].

На рис. 6 наведено граф, який відображає концепції верхнього рівня та зв'язки між ними.

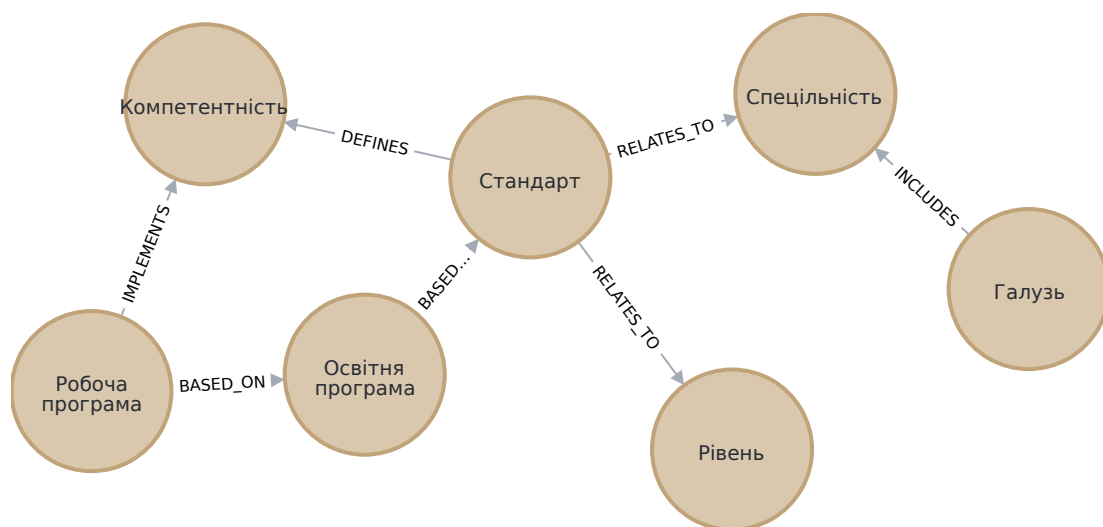


Рис. 6 Концепції верхнього рівня

Затверджені МОН України освітні стандарти [3], які представляють сукупність характеристик та вимог до навчання і мають відношення до Спеціальностей, що включені в конкретну Галузь знань. Наприклад: Галузь знань - 12 Інформаційні технології, Спеціальність - 123 Комп'ютерна інженерія.

Крім того Стандарт визначається своїм відношенням до Рівня (ступеня) освіти. Наприклад: Магістр.

Спеціальності та Галузі знань утворюють ієрархію та класифікують Стандарти. Наприклад: Стандарт вищої освіти України: другий (магістерський) рівень, галузь знань 12 Інформаційні технології, спеціальність 123 Комп'ютерна інженерія.

На рис. 7 наведено приклад такої ієрархії для факультету Інформаційних технологій.



Рис. 7 Ієрархія стандартів для ФІТ

Навчальний заклад розробляючи Освітні програми виходить з вимог, які визначені у відповідному Стандарті (рис. 8). Утворюється замикання, де Освітня Програма імплементує Компетентності та відноситься до Стандарту, якій визначає вимоги до цих Компетентностей. Множина формалізованих у вигляді DAG Компетентностей утворює свого роду простір, через координати

якого надалі мають бути визначені всі елементи навчального процесу: робочі програми, методичні матеріали, завдання, тести, результати оцінювання.

Простір Компетентностей дозволить надалі формувати навчальний шлях студента, як траєкторію його руху в цьому просторі: від початкової точки, отриманої в результаті первинного оцінювання, через виконання завдань та тестів до кінцевої, яка фіксує результати освіти. Включення в цей процес можливості самооцінювання через взаємодію з агентним середовищем системи, може зробити процес навчання ітеративним та безперервним.

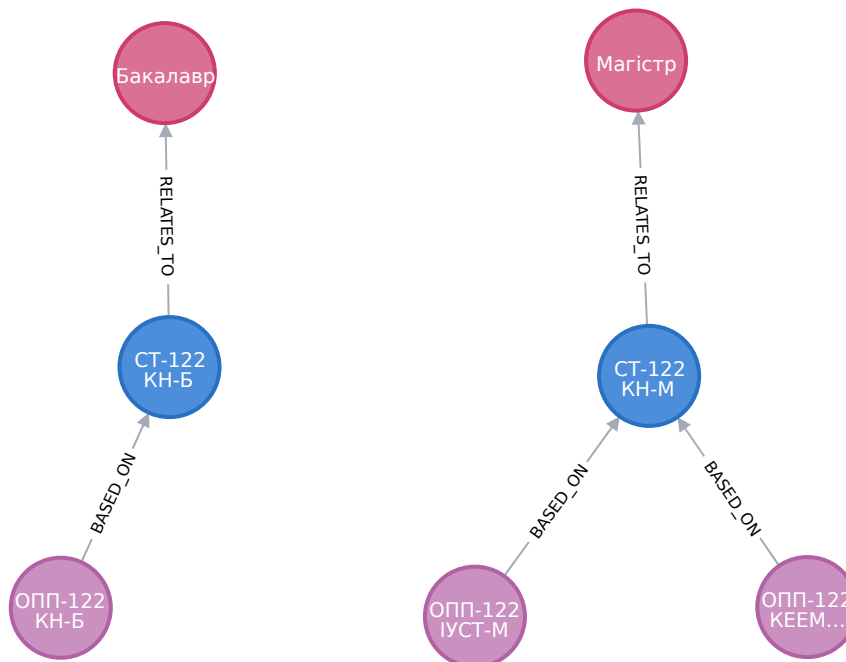


Рис. 8 Освітні програми базуються на відповідних стандартах

3.4 Обробка текстів природної мови

Як правило нормативні документи та ресурси навчального процесу: робочі програми, методичні матеріали, завдання, тести, результати оцінювання представлені у вигляді неструктурованих та слабоструктурованих текстів

природною мовою. Перетворення текстових матеріалів в граф знань може бути непростим та трудомістким завданням, яке потребує залучення висококваліфікованих експертів. Застосування сучасних методів та інструментів NLP може значно спростити та прискорити цей процес, забезпечити його більшу стійкість та цілісність. Для автоматизації завдання трансформації текстових документів з визначеним контекстом предметної області в граф знань можуть використані підходи та інструменти запропоновані в [19].

Процес трансформації складається з таких основних елементів: токенизація (tokenization) - поділ тексту на значущі сегменти, які називаються токенами; позначення частин мови (part-of-speech tagging) - визначення для кожного токена частини мови та позначення її; морфологічний аналіз (morphology) - аналіз структури слів, виділення кореня; лематизація (lemmatization) - групування відмінюваних форм слова так, щоб їх можна було проаналізувати як єдиний елемент; розпізнавання іменованих сутностей (named entity recognition) - ідентифікація та позначення іменованих та числових сутностей; розбір залежностей (dependency parsing) - розпізнавання меж речень і перебирання базових фраз іменників; зв'язування сутностей (entity linking) - зв'язування сутностей з вузлом графа KB.

В результаті трансформації створюється прототип підграфа, який має бути розглянутий експертами та в разі прийняття інтегрований в граф KB. Процес трансформації та розгляду є ітеративним, поки зміни не будуть прийняті. В разі відхилення варіанту наступна ітерація виконується з новими значеннями параметрів, які мають наблизити результат наступної ітерації до цілі. В разі прийняття KB набуває нових знань про предметну область, отриманих з обробленого джерела.

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

В ході дослідження було опрацьовано більше 100 джерел: серед них книги, підручники, наукові статті, тематичні та проєктні сайти, обговорення, стандарти, патенти. Розглянуто вихідні коди проєктів з відкритим кодом (open source) дотичних до досліджуваної предметної області на таких мовах програмування як Java, C/C++, R, Python, Julia, Go, Prolog, Common Lisp. Шляхом розробки прототипів, підготовки та аналізу даних, реалізації тестових та навчальних завдань виконано апробацію моделей, пакетів, фреймворків та бібліотек, порівняння їх властивостей та меж застосування. З результатами та артефактами випробування досліджуваних засобів та методів можна ознайомитися у відкритому репозиторії магістерської роботи [20].

В результаті дослідження визначено шляхи інтелектуалізації існуючих систем EL через застосування онтологічних KB та AOD. Інтеграція EL з такими підсистемами як ITS та CBE.

ITS підтримує детальну модель навичок та історії навчання студента, яка представлена онтологією або графом знань. Ця модель допомагає системі відслідковувати поточний рівень знань і траєкторію навчання. ITS також включає онтології або графи знань, специфічні для дисципліни, яка викладається. Такі представлення охоплюють структуру предметної області та зв'язки між поняттями, дозволяючи системі надавати контекстуально релевантні рекомендації. ITS може адаптувати зміст і навчальні стратегії відповідно до індивідуальних потреб і прогресу студента. Вона може регулювати складність питань, надавати додаткові пояснення та за потреби пропонувати додаткові матеріали. ITS забезпечує миттєвий і конструктивний зворотний зв'язок зі студентами. Зворотній зв'язок може включати пояснення правильних і неправильних відповідей, підказки щодо вирішення проблеми та

пропозиції щодо покращення. Вона надає можливість самооцінювання, успіхи студентів відстежуються для визначення сфер, які потребують покращення.

СВЕ надає карти компетентностей або графи знань, які забезпечують представлення зв'язків і залежностей між компетентностями. Вона підтримує індивідуальні навчальні маршрути для кожного студента на основі його поточних компетентностей і бажаних результатів навчання. Це включає в себе рекомендації щодо послідовності компетентностей, які необхідно опанувати, і навчальних ресурсів необхідних для використання. СВЕ може автоматизувати оцінювання компетентностей, як початкове, так і підсумкове, щоб визначити рівень володіння студентом конкретними компетентностями. Ці оцінки можуть адаптуватися до продемонстрованих студентом знань та навичок. СВЕ аналізує дані про компетентності, щоб визначити тенденції продуктивності навчання. Така інформація може допомогти викладачам і адміністраторам приймати рішення щодо коригування навчальних програми та підтримки підтримки студентів. СВЕ може вести систему цифрових відзнак або рейтингів на основі досягнутих студентами компетентностей, що може сприяти мотивації в навчанні.

Підходи AOD використовують ІА для створення більш персоналізованого, інтерактивного та адаптивного освітнього середовища. ІА слугують персональним гідом студента, контролюють індивідуальні сильні та слабкі сторони та вподобання у навчанні. На основі цієї інформації ІА можуть рекомендувати персоналізовані навчальні шляхи, ресурси та заходи, адаптовані до потреб кожного учня та темпу навчання. Вони підтримують спільне навчання, сприяючи груповим обговоренням, взаємному оцінюванню та спільним проектам. ІА можуть допомогти координувати діяльність групи та надати допомогу, коли це необхідно. Вони автоматизують оцінювання завдань, тестів та іспитів, надаючи миттєвий зворотний зв'язок студентам. ІА також можуть виявити поширені помилкові уявлення та запропонувати дії для їх

виправлення. Адміністративні ІА допомагають викладачам в управлінні курсами, складанні розкладу та моніторингу прогресу студентів. Це дає викладачам можливості зосереджуватися більше на навчанні, а не на адміністративних завданнях. ІА можуть створювати віртуальні лабораторні середовища та симуляції, що дозволяє студентам проводити експерименти чи симуляції безпечно та віддалено, інтегрувати навчальні ігри, щоб надавати вказівки, завдання та оцінки в ігровому середовищі.

Пропонується реалізація інтелектуальних підсистем навчання через проєктування MAS, яка б складалася з таких компонентів як інтерфейси викладача та студента, внутрішня шина обміну повідомленнями, гід студента, асистент викладача, профіль студента, супервізор, агент запитів до KB, агент навчальних ресурсів, адміністратор. При потребі до інтелектуальних підсистем можуть бути розроблені та додані нові типи ІА з необхідними функціями та поведінкою при умові виконання вимог відповідних протоколів та інтерфейсів.

Практичне розгортання системи планується з використанням контейнерних технологій та інструментів. Система обміну повідомленнями [16], сховища KB [18], web-сервери, сервіси web-апі, модулі агентів виконуються під управлінням системи управління контейнерами, таких як Docker. Така архітектура надає можливості розміщення системи в різноманітних середовищах, в тому числі хмарних, гнучкого масштабування, балансування навантажень, оптимізації використовуваних обчислювальних ресурсів. При розробці модулів агентів буде можливо використовувати широке коло мов програмування, пакетів, бібліотек, фреймворків забезпечуючи їхню взаємодію через уніфіковані протоколи та інтерфейси і систему обміну повідомленнями.

Підсумовуючи результати можна стверджувати, що досліджені та запропоновані підходи до застосування KBIA в галузі EL мають важливе інноваційне значення та актуальну практичну цінність і можуть бути

використані для створення реального програмного проєкту та продукту, який стане розвитком існуючих систем EL в бік більшої інтерактивності, адаптивності та інтелектуальності.

ВИСНОВКИ

В магістерській роботі виконано дослідження підходів до застосування KBIA в галузі EL. Визначено об'єкт - KBIA, які діють в змінюваному віртуальному середовищі, та предмет - аналіз підходів AOD та методів KR для застосування їх в підсистемах підтримки EL, таких як LMS, ITS та CBE.

Для досягнення поставленої мети дослідження проведено аналіз сучасного стану в галузі IA та вивчено засоби та інструменти AOD та KR; визначено можливості та межі застосування вивчених методів для побудови системи взаємодіючих KBIA; вироблено пропозиції з розробки та інтеграції підсистем інтелектуальної підтримки EL на основі знань та процесів предметної області сфери вищої освіти.

Дослідження проводилося шляхом виконання таких завдань.

На стадії аналізу було вивчено такі базові поняття, процеси та методи як KR, когнітивні процеси, міркування, адаптивність, навчання та вирішення задач. Показано структуру об'єкта та предмета дослідження. Запропоновано класифікацію IA та відмічено особливості KBIA. Обґрунтовано актуальність та доцільність застосування AOD та онтологій в EL.

На стадії моделювання досліджено та апробовано моделі та методи IA, KR, планування та прийняття рішень. Запропоновано модель KBIA як систему взаємодіючих агентів та їх можливу спеціалізацію. Запрограмовано прототипи, виконано навчальні та тестові приклади, підготовлено необхідні дані.

На стадії розробки визначено сценарії, функції та актори процесів в EL з інтелектуальними компонентами. Показана класифікація компонентів та їхня взаємодія. Запропоновано реалізацію KB як онтологічного графа знань засобами графової системи баз даних. Наведено приклади формування елементів KB для підтримки відношень на рівні освітніх стандартів та програм.

Намічено підходи до перетворення неструктурованої та слабоструктурованої текстової інформації в онтологічні графи KB засобами NLP.

Підсумовуючи та оглядаючи пройдений шлях можна зробити такі головні висновки.

- Сучасний стан розвитку інформаційних технологій, цифрової інфраструктури та об'єми доступної інформації вимагають автоматизації інтелектуальної праці.
- Перспективним підходом до створення систем підтримки інтелектуальної діяльності є застосування AOD та онтологічного KR.
- Галузь EL є важливою сферою впровадження KBIA.
- ITS та CBE - спричинять, на нашу думку, революцію в EL та в освіті в цілому, піднімуть їх на якісно новий рівень.

Не чекаємо змін, а робимо їх.

Нехай щастить!

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Harbouche K., Djoudi M. Agent-Based Design for E-learning Environment. Journal of Computer Science (3-6, pp. 383-389). 2007.
2. Rani M., Vaibhav K. Vyas O. P. An Ontological Learning Management System. 2016. URL:<https://arxiv.org/pdf/1708.09475.pdf>
3. Затверджені стандарти вищої освіти.
URL:<https://mon.gov.ua/ua/osvita/visha-osvita/naukovo-metodichna-rada-ministerstva-osviti-i-nauki-ukrayini/zatverdzeni-standarti-vishoyi-osviti>
4. Ehimwenma K., Krishnamoorthy S. Design and Analysis of a Multi-Agent E-Learning System Using Prometheus Design Tool. IAES IJ-AI (Vol.10-1). 2021.
5. Bourekkache S., Kazar O. Agent-Based Approach for E-Learning. 2018.
URL:https://www.researchgate.net/publication/260762449_Agent-Based_Approach_for_E-Learning
6. Shahbazi Z., Byun Y.-C. Agent-Based Recommendation in E-Learning Environment Using Knowledge Discovery and Machine Learning Approaches. Mathematics (10-1192). 2022.
7. Bordini R., Hübner J., Wooldridge M. Programming Multi-Agent Systems in AgentSpeak using Jason. John Wiley & Sons Ltd. 2007.
8. ACT-R Research Group. URL:<http://act-r.psy.cmu.edu/>
9. Laird J. E. Introduction to the Soar Cognitive Architecture. 2022.
URL:<https://arxiv.org/pdf/2205.03854.pdf>
10. Lehman J. F., Laird J., Rosenbloom P. A Gentle Introduction to Soar, an Architecture for Human Cognition. 2006.
URL:<https://courses.csail.mit.edu/6.034f/ai3/SOAR.pdf>

11. Noy N., McGuinness D. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory. 2001.

12. Russell, S. Norvig, P. *Artificial Intelligence: A Modern Approach* (3d ed.). Pearson Education. 2010.

13. Прикладные нечеткие системы (ред. Тэрано Т. , Асаи К. , Сугэно М.). Мир. 1993.

14. Haykin S. *Neural Networks: A Comprehensive Foundation* (2nd ed.). Pearson Education. 1999.

15. Sutton R., Bart A. *Reinforcement Learning: An Introduction* (2nd ed.). The MIT Press. 2012.

16. *Connective Technology for Adaptive Edge & Distributed Systems*.
URL:<https://nats.io/>

17. Barrasa J., Webber J. *Building Knowledge Graphs: A Practitioner's Guide*. O'Reilly. 2023.

18. Neo4j Graph Database. URL:<https://neo4j.com/product/neo4j-graph-database/>

19. *Industrial-Strength Natural Language Processing in Python*.
URL:<https://spacy.io/>

20. Репозиторій магістерської роботи.
URL:<https://github.com/youryharchenko/m-work>