

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Центр Науково навчальний центр заочної форми навчання
(повна назва)

Кафедра Програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів моделювання логічних мереж при розробці систем
віртуального навчання
(тема)

Виконала:

студентка 2 курсу, групи ППЗзм-22-1

спеціальності 121- Інженерія програмного
забезпечення

(код і повна назва спеціальності)

освітньо-наукової програми Інженерія
програмного забезпечення

(повна назва освітньої програми)

Бобракова А.В.

(прізвище, ініціали)

Керівник проф. Шубін І.Ю.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____

З.В.Дудар

2024 р.

Харківський національний університет радіоелектроніки
Факультет ННЦ ЗФН

Кафедра програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121– Інженерія програмного забезпечення

(код і повна назва)

Освітньо-наукова програма Інженерія програмного забезпечення с

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 20 ____ р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові Бобраковій Анжеліці Васильовні

(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів моделювання логічних мереж при розробці систем віртуального навчання»

Затверджена наказом по університету від 29.03.2024 р. № 42 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 20.06.2024 р.

3. Вихідні дані до роботи: опис досліджуваних мережевих баз даних, вимоги до розробки архітектури бази даних для проведення досліджень за обраною предметною областю, мови програмування Python, середовища розробки PyCharm

4. Перелік питань, що потрібно опрацювати в роботі: провести дослідження методів побудови логічних мереж даних та обрати технології і програмні рішення які будуть актуальними для реалізації гнучкого програмного застосунку збудувати архітектуру програмного забезпечення з урахуванням визначеної моделі

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі та постановка задачі	03 квітня 2024 р.	<i>виконано</i>
2	Огляд існуючих методів	10 квітня 2024	<i>виконано</i>
3	Розробка алгоритмів, проектування та розробка ПЗ	15 квітня 2024	<i>виконано</i>
4	Підготовка пояснювальної записки	20 квітня 2024 р.	<i>виконано</i>
5	Спецчастина	28 квітня 2024 р.	<i>виконано</i>
6	Підготовка презентації та доповіді	03 травня 2024 р.	<i>виконано</i>
7	Попередній захист	05 травня 2024 р.	<i>виконано</i>
8	Нормоконтроль, рецензування	07 червня 2024	<i>виконано</i>
9	Занесення роботи в електронний архів	08 червня 2024	<i>виконано</i>
10	Допуск до захисту в зав. кафедри	11 червня 2024 р.	<i>виконано</i>

Дата видачі завдання 30 березня 2024 р.

Студентка _____
(підпис)

_____ **Анжеліка БОБРАКОВА**

Керівник роботи _____
(підпис)
(підпис)

_____ **Ігор ШУБІН**
(посада, прізвище, ініціали)
(посада, прізвище, ініціали)

РЕФЕРАТ/ ABSTRACT

Пояснювальна записка: 68 с., 24 рис., 4 табл., 4 дод., 14 дж.

НАВЧАЛЬНА СИСТЕМА, АЛГЕБРА, ПРЕДИКАТ, БУЛЕВА ФУНКЦІЯ,
ЛОГІЧНА МЕРЕЖА, МНОЖИНА.

Метою роботи є розробка інформаційно-навчальної системи для побудови логічних мереж на основі алгебри скінченних предикатів та основних методів побудови мультиагентної системи підтримки віртуального навчання.

Методи розробки базуються на методах математичного моделювання, моделювання розподілених програмних систем та алгебрі скінченних предикатів.

У результаті роботи здійснена реалізація інформаційно-навчальної системи використання мультиагентної системи початкових матеріалів для аналізу логічними мережами

EDUCATIONAL SYSTEM, ALGEBRA, PREDICATE, BOOLEAN
FUNCTION, LOGIC NETWORK, SET.

The purpose of the work is the development of an information and learning system for building logical networks based on the algebra of finite predicates and the main methods of building a multi-agent virtual learning support system.

The development methods are based on the methods of mathematical modeling, modeling of distributed software systems and the algebra of finite predicates.

As a result of the work, the implementation of the information and educational system of the use of the multi-agent system of initial materials for the analysis of logical networks was carried out.

Я, Бобракова Анжеліка Василівна, студентка гр. ПЗзм-22-1, здобувачка вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів моделювання логічних мереж при розробці систем віртуального навчання», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомена з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ	6
1 Аналіз стану розв'язання проблеми та обґрунтування цілей дослідження	9
1.1 Аналіз моделей побудови віртуального навчального середовища.....	9
1.2 Аналіз математичного опису	11
1.3 Постановка задач дослідження	16
2 Аналіз методів побудови логічних мереж	19
3 Розробка логічної мережи взаємодії з штучними агентами	27
4. Опис програмних рішень	36
5. Опис можливості використання отриманих результатів.....	48
Висновки	53
Перелік джерел посилання	55
Додаток А Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії	57
Додаток Б Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ...	58
Додаток В Слайди презентації	59
Додаток Г Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015	68

ВСТУП

В Україні постійно зростає інтерес до систем електронного навчання, розроблено багато курсів, які орієнтовані на використання сучасних інформаційно-комунікаційних технологій для навчання. Тому питання ефективної організації таких систем є достатньо актуальним. Електронне навчання – це форма навчання, за якої основні носії навчальної інформації є електронними ресурсами, а засоби спілкування викладача та студентів ґрунтуються на використанні інформаційно-комунікаційних технологій, насамперед комп'ютерної техніки, мережевих технологій, зокрема Інтернет [1].

Важливою складовою якісної системи підтримки електронного навчання є спеціалізоване програмне забезпечення, яке повинне забезпечити колаборативний сценарій використання навчального середовища. Високі вимоги до колаборативних систем підтримки електронної освіти потребують програмних рішень з гнучкою архітектурою [2]. Більшість створених моделей та основаних на них програмних та апаратних засобів носять фрагментарний характер, діючи на обмежених підмножинах предметної області. Це пояснюється тим, що знання про мову ще недостатньо формалізовані, а виникнення мови навіть на формальному рівні описано не повністю. Вибір математичного апарату досить часто орієнтований на конкретний вид граматичної обробки чи використовується для формального описання зв'язків певного рівня.

Інший напрямок розробок полягає в тому, щоб забезпечити обчислювальну систему банком даних, імітуючи знання й досвід людини, отримані ним за життя. Такий банк даних з обслуговуючим його пакетом програм отримав назву лінгвістичного інтерфейсу. Незважаючи на практичне використання деяких з побудованих інтерфейсів, основні проблеми, пов'язані зі змістовою обробкою запитів на природній мові, в них вирішені далеко не повно [1].

Алгебра кінцевих предикатів и розроблені на її основі алгоритмічні й програмні рішення дозволяють вирішити більшість із зазначених вище протиріч.

Математичний апарат алгебри скінченних предикатів також можна використати на етапі поповнення словника новими поняттями та доповнення словарних статей існуючих понять новими визначеннями.

Лінгвістичне забезпечення автоматизованої галузевої бібліотечної системи є найважливішим інструментом, який надає користувачу доступ до інформаційних ресурсів галузевої бібліотеки. Однією зі складових комплексного лінгвістичного забезпечення сучасної автоматизованої галузевої бібліотечної системи є лінгвістичне забезпечення процесів класифікації та систематизації документів, базою для розробки якого є інформаційно-пошукові тезауруси. Масив ключових слів такого тезауруса формується на основі концептуальної моделі певної предметної галузі. Ця модель може бути представлена електронним галузевим термінологічним словником [3].

Сучасний стан системи освіти в Україні характеризується наявністю ієрархічної структури, розподіленним характером та великою кількістю взаємозв'язків як всередині освітньої системи, так і з зовнішнім світом (наприклад, з органами влади, з підприємствами, зі споживачами). Це говорить про актуальність проблем управління в системі освіти на основі нових принципів, які враховують розподіленість, ієрархічність та складність структури об'єкту управління, а також дозволяють визначити узгоджене рішення завдань управління функціонуванням та розвитком. Одним з напрямків, які активно розвиваються в сучасних умовах глобальної інформатизації та Інтернету, є електронне навчання. Електронне навчання – це форма навчання, за якої основні носії навчальної інформації є електронними ресурсами, а засоби спілкування викладача та студентів ґрунтуються на використанні інформаційно-комунікаційних технологій, насамперед комп'ютерної техніки, мережевих технологій, зокрема Інтернет, й мобільного зв'язку.

Питанням щодо формалізації управління ресурсами в системі електронного навчання присвячена дана робота.

З розвитком та поглибленням інформаційної бази даної області математики з'явилась проблема складності сприйняття матеріалу для студентів та зацікавлених осіб.

Метою кваліфікаційної роботи є підвищення ефективності процесів прийняття рішень в системах електронного навчання шляхом розробки моделей та програмних компонент для побудови відкритої мультиагентної системи для оперативної обробки даних та розробка інформаційно-навчальної системи для вивчення матеріалу з алгебри скінченних предикатів та основних методів мінімізації булевих функцій.

1 АНАЛІЗ СТАНУ РОЗВ'ЯЗАННЯ ПРОБЛЕМИ ТА ОБҐРУНТУВАННЯ ЦІЛЕЙ ДОСЛІДЖЕННЯ

1.1 Аналіз моделей побудови віртуального навчального середовища

Ефективність цифрових обчислювальних структур перебуває в прямій залежності від того, наскільки просто і надійно, в реальному масштабі часу зможуть вони безпосередньо сприймати і переробляти аудіовізуальну та текстову інформацію. Сформована в даний час диспропорція між обсягом знову поступаючої інформації і здатністю людини до її переробки, а також такі завдання як пошук інформації, машинний переклад і ін. змушують шукати нові шляхи для їх вирішення. Знання алгоритмів розумової діяльності людини багато в чому вирішить проблему при передачі рішення розглянутих завдань ЕОМ.

В даний час функціонування автоматизованих систем неможливо без організації великих за обсягом інформаційних банків. До складу інформації входить велика кількість неформалізованих даних, що представляють собою різні текстові повідомлення природною мовою для систем управління, це інформація про об'єкт управління, пов'язана з його структурою, функціонуванням і прийняттям рішень щодо стратегії управління. Сьогодні не виникає сумнівів, що обмін інформацією між користувачем і ЕОМ повинен проводитися на природній або хоча б близькій до природної мовою. У такому випадку машина повинна містити деякий набір знань, відповідний до нормативної граматики, і подібно людині успішно оперувати ним. Таким чином, необхідно створення процесора, який дозволить комп'ютеру сприймати природно-мовний запит і формувати відповідь. Тоді людині, яка працює з машиною, не знадобиться тривалих і серйозних занять з оволодіння мистецтвом програмування і вивчення специфіки ЕОМ.

Проблемам накопичення знань про природу мову, її устрій, принципи функціонування, вибір мови формального опису присвячено багато робіт. Успішне вирішення лінгвістичних проблем машинного перекладу, різного роду завдань по автоматичній обробці текстової інформації та мовного управління

виробничими процесами, а також лінгвістичних проблем спілкування з ЕОМ обіцяє в майбутньому величезну економію сил і засобів.

Створення сучасних технічних засобів для автоматичної обробки мовних та фонетичних закономірностей вимагає розробки нових методів синтезу обчислювальних структур. Застосування таких способів дозволить на одному і тому ж пристрої одночасно вирішувати цілий ряд задач обробки та моделювання лінгвістичних завдань. Створювані для подібних систем алгоритми і програмні засоби повинні мати властивості регенерації та часткової регенерації вхідних і вихідних даних, бути надійними в експлуатації, володіти можливістю корекції в процесі обігу. [3]

До тих пір відсутній єдиний послідовний підхід до моделювання природних і фонетичних закономірностей. Більшість створених моделей і заснованих на них програмних і апаратних засобів носили фрагментарний характер, діючи на обмежених підмножинах предметної області. Це пояснюється тим, що знання про мову ще недостатньо формалізовані, а поява мови навіть на формальному рівні описана не повно. Вибір математичного апарату найчастіше орієнтований на конкретний вид граматичної обробки або використовується для формального опису зв'язків певного рівня.

Однією з найважливіших проблем синтезу схем, пристроїв автоматики, вимірювальної та обчислювальної техніки, кібернетики є отримання найбільш простих (мінімальних) формул, що описують їх функціонування. При цьому проблема мінімізації є одним з основних етапів аналізу і синтезу схем автоматів, створення програмного забезпечення обчислювальних систем. Дане завдання, як показує науково-технічна література, їй присвячена, не вирішене в достатній мірі.

Система логічних рівнянь зводиться до розв'язання систем лінійних нерівностей з цілочисельними невідомими. Передбачуваний метод рішення не пов'язаний із застосуванням методів лінійного програмування. Суть методу - в узагальненні способу мажоритарної декомпозиції. Як недолік можна відзначити великий перебір при вирішенні систем лінійних нерівностей, що в цілому знижує

ефективність методу на практиці. Аналогічний підхід до вирішення системи логічних рівнянь шляхом заміни системи логічних функцій мережею безперервно-диференційовних функцій викладено в [5]. В цілому це дозволяє одержати завдання нелінійного програмування з лінійними обмеженнями, виключивши при цьому Великий перебір.

Велику роботу по розробці і створенню універсальних і багатофункціональних автоматів з практично необмеженими функціональними можливостями, які мають граничним структурним швидкодією, останнім часом провели автори робіт [6,7]. В основу їх практичних розробок були покладені методи синтезу багатофункціональних автоматів. В останні роки бурхливий розвиток найбільш поширеного виду великих інтегральних схем програмованих логічних матриць (ПЛМ) спонукав ряд дослідників модифікувати відомі методи рішення логічних рівнянь. Прикладом може служити метод множення диз'юнктивних нормальних форм, а також його варіації для синтезу комбінаційних схем або логічних перетворювачів кінцевих автоматів. Суть методу зводиться, з одного боку, до матричної реалізації комбінаційних схем для синтезу системи булевих функцій, заданих у диз'юнктивній нормальній формі. З іншого боку, аналіз і синтез схем з ПЛМ зводиться до дослідження векторно-матричних рівнянь, що описують схеми [8]. Однак, питання багатозначності і неоднорідності практично не знайшли місця при розгляді модифікації матричних методів.

1.2 Аналіз математичного опису

За структурними ознаками взаємодії навчальної системи з користувачем АНС поділяються на два базових класів (див. рис. 1.1): розімкнуті (без зворотного зв'язку) та замкнуті (зі зворотним зв'язком) системи, які відрізняються принциповим підходом до процесу навчання.

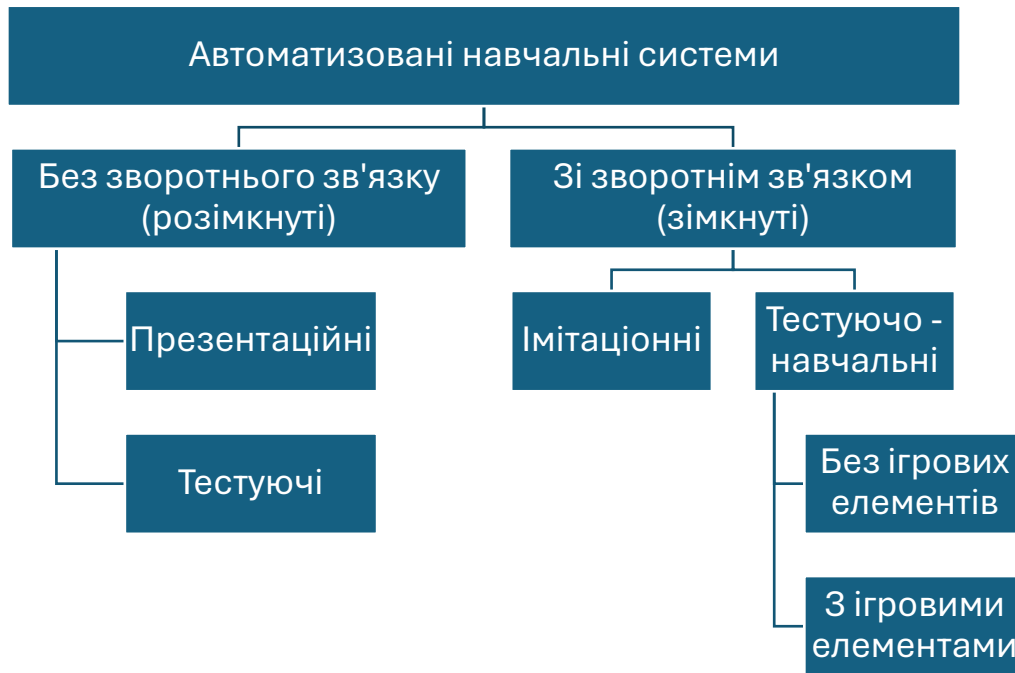


Рисунок 1.1 - Класифікація структурної побудови АНС

У розімкнутих АНС не враховуються відгуки учнів на поставлені питання і не корегується послідовність пред'явлення навчального матеріалу в функції ступеня засвоєння учням досліджуваної теми. Тут лише виконується певна заздалегідь задана програмним шляхом послідовність викладу уроку чи контрольних питань. При цьому найбільш простими і одночасно інформативними з числа розімкнутих АНС є системи з презентаційною структурою, яка представляє собою послідовне включення ланок "АНС" і "Учень".

У АНС даного типу присутній тільки прямий інформаційний зв'язок між системою і учнями, якому послідовно надається візуальна інформація з монітора ЕОМ. При цьому учень знаходиться в режимі пасивного спостерігача, від якого не потрібно ні яких відгуків по взаємодії з АНС. Прикладом презентаційної АНС може служити навчальна програма з курсу Visual Basic, розроблена автором, що представляє набір слайдів з демонстраційними прикладами.

При реалізації будь-якої зі структур АНС використовуються цілком певні алгоритмічні підходи, які диктуються методикою проведення навчального заняття. Зазвичай будь-яка навчальна система являє собою сукупність порцій

інформації, званої слайдами, які в тій чи іншій формі пред'являються учневі. Сучасна обчислювальна техніка володіє широкими функціональними можливостями і дозволяє використовувати у слайдах інформацію, представлену у вигляді звичайного тексту, графічного зображення, аудіо і відео фрагментів. При цьому у слайдах можна зосередити всі засоби представлення інформації, що існують в даний час для підвищення ефективності навчального процесу. З іншого боку, використання звукового супроводу і відеозаписів веде до подорожчання навчальної системи, що не завжди є виправданим, особливо при широкій цільовій аудиторії. В даний час, як найбільш оптимальний варіант в більшості курсів програмованого навчання, застосовують текстове та графічне представлення інформації.

Для розробки системи була обрана мова C#, так як має дуже зручне середовище розробки та являється універсальною мовою програмування, з використанням технології ASP.NET для графічного відображення матеріалу.

Освіта є однією з важливіших галузей економіки держави, яка обумовлює розвиток культури, науки, техніки та технологій. Можна стверджувати, що рівень розвитку держави залежить від якості освітньої системи, ефективності процесів управління та підтримки розвитку науки. Процеси розвитку в системі освіти є складними, з великою кількістю взаємних зв'язків та потребують нових прогресивних методів управління.

Відкритий характер сучасного інформаційного суспільства та розвиток економіки веде до прискорення науково-технічного прогресу та росту конкуренції на ринку. Це призводить до необхідності пошуку нових методів та засобів управління, які повинні більш якісно та ефективно задовольняти індивідуальні потреби споживачів. Один з таких підходів передбачає побудову мережевої структури організації, де окремі підрозділи розглядаються як автономні підприємства. Перевагами такої організації є прямий характер взаємодії окремих підрозділів з іншими підприємствами, свобода приєднання та виходу із організації. Це дозволяє підприємствам адаптуватися до ринкових змін

за рахунок гнучкості, відкритості, узгодженості та оперативності процесів прийняття рішень.

При цьому такий тип організаційної структури передбачає розробку спеціальних систем щодо управління ресурсами, тому що існуючі традиційні програмні системи не задовольняють вимоги щодо своєчасної ідентифікації потреб, оперативного прийняття рішень, реструктуризації ресурсів для ефективного вирішення різноманітних завдань в умовах динамічного середовища.

В останні роки широке розповсюдження отримав термін e-learning, який означає електронне навчання. Це поняття розглядається як розширення терміну дистанційне навчання. В Україні постійно зростає інтерес до систем електронного навчання, розроблено багато курсів, які орієнтовані на використання сучасних інформаційно-комунікаційних технологій для навчання [1], [3]. Тому питання ефективної організації таких систем є достатньо актуальним.

Сучасний підхід до вирішення завдань оперативної обробки інформації в процесах прийняття рішень пов'язано із застосуванням мультиагентних технологій, які сьогодні отримали інтенсивний розвиток [4], [5]. Інформаційні технології, які ґрунтуються на агентах, з'явилися на стику методів штучного інтелекту, об'єктно-орієнтованого програмування, паралельних обчислень та телекомунікацій. В основі цієї технології лежить поняття «агента», програмного об'єкта, здатного сприймати ситуацію, приймати рішення і взаємодіяти з собі подібними [6], [7]. Ці можливості кардинально відрізняють мультиагентні системи (МАС) від традиційно організованих систем, реалізуючи таку важливу властивість як здатність до самоорганізації. При цьому агенти можуть діяти від імені осіб, які приймають рішення, на основі виданих їм повноважень, тобто в автоматичному режимі вести переговори, знаходити варіанти рішень та узгоджувати ці рішення.

Вирішення завдання одним агентом представляє точку зору класичного штучного інтелекту, згідно якої інтелектуальна система (агент) має усі

необхідні здібності, знання та ресурси для вирішення задачі [6]. Розподілений штучний інтелект передбачає, що окремий агент може мати лише часткове уявлення загальної проблеми та здатен вирішити лише її частину [7]. Тому для вирішення складних завдань потрібна взаємодія множини агентів, що формує підґрунтя для концептуальної новизни новітніх комп'ютерних технологій із використанням мультиагентних систем.

Не зважаючи на велику кількість публікацій щодо проектування та застосування мультиагентних систем, зокрема в системах електронного навчання, питання оперативної обробки даних для вирішення завдань розподілу ресурсів в режимі реального часу залишаються відкритими.

Особливу увагу дослідників займає, в першу чергу, завдання мінімізації форм представлення багатозначних логічних функцій, що грають важливу роль в проблемі синтезу дискретних автоматів [9]. У зв'язку з актуальністю проблеми мінімізації багатозначних перемикальних функцій з'явилися публікації, в яких робляться спроби узагальнити відомі методи та алгоритми знаходження мінімальних форм представлення булевих функцій на багатозначний випадок. Але складність подачі інформації в даних публікаціях призводить до потреби створення навчального інтерфейсу, орієнтованого на широке коло зацікавлених, заснований на простоті і доступності представленого матеріалу.

Існує декілька видів навчальних програм. Підставою для класифікації є зазвичай особливості навчальної діяльності учнів при роботі з програмами. Багато авторів виділяють чотири типи навчальних програм:

- тренувальні та контролюючі;
- наставницькі;
- імітаційні та моделюючі;
- розвиваючі ігри.

Програми 1-го типу (тренувальні) призначені для закріплення умінь і навичок. Передбачається, що теоретичний матеріал вже вивчений. Ці програми у довільній послідовності пропонують учневі питання та завдання і підраховують кількість правильно і неправильно вирішених завдань (у разі

неправильної відповіді може видаватися заохочуюча учня репліка). При неправильній відповіді учень може отримати допомогу у вигляді підказки.

В даній роботі для реалізації було обрано програму 2-го типу (наставницькі). Навчальні системи наставницького типу пропонують учням теоретичний матеріал для вивчення. Завдання і питання слугують в програмах для організації людино-машинного діалогу, для управління ходом навчання. Учень завжди може переглянути пройдений матеріал для повторного вивчення теоретичної бази.

Програми наставницького типу є прямими спадкоємцями засобів програмного навчання 60-х років у тому сенсі, що основним теоретичним джерелом сучасного комп'ютерного або автоматизованого навчання слід вважати програмоване навчання. У публікаціях зарубіжних фахівців і сьогодні під терміном «програмоване навчання» розуміють сучасні комп'ютерні технології. Одним з основоположників концепції програмованого навчання є американський психолог Б.Ф. Скіннер.

Програми 3-го типу (моделюючі) засновані на графічно-ілюстративних можливостях комп'ютера, з одного боку, і обчислювальних, з іншого, і дозволяють здійснювати комп'ютерний експеримент. Такі програми надають учневі можливість спостерігати на екрані дисплея певний процес, впливаючи на його хід подачею команди з клавіатури, яка змінює значення параметрів.

Програми 4-го типу (ігри) надають в розпорядження учня деяку уявну середу, набір якихось можливостей і засобів їх реалізації. Використання наданих програмних засобів для реалізації можливостей, пов'язаних з вивченням світу гри і діяльністю в цьому світі, приводить до розвитку учня, формуванню у нього пізнавальних навичок, самостійного відкриття їм закономірностей, відносин об'єктів дійсності, що мають загальне значення [10].

1.3 Постановка задачі

У додатках важлива задача знаходження мінімальних форм часткових і

слабовизначених кінцевих предикатів, а також сукупності скінченних предикатів. Вирішення цих та інших питань, пов'язаних з мінімізацією запису формули алгебри предикатів, дозволяє, по-перше, зменшити обсяг необхідної пам'яті ЕОМ і витрати машинного часу на обробку вихідної інформації шляхом її попереднього стиску, по-друге, скоротити апаратні витрати, необхідні для побудови швидкодіючих спеціалізованих пристроїв. спеціалізованих пристроїв.

Метою роботи є представлення математичних моделей, алгоритмів мінімального представлення формул, що описують структури алгебри кінцевих предикатів і програмних засобів для вирішення завдань оптимізації інформаційних масивів в задачах обробки мовних повідомлень.

Після проведення аналізу предметної області та складання технічної характеристики, порівняльного аналізу результатів між різними програмними реалізаціями для представлення математичних формул було виявлено різноманітні недоліки. Так як кожна навчальна система використовує різні алгоритми та методи для підвищення акуратності та швидкості навчання, представлення інформації, математичних символів та ускладнених формул, можна побачити, яка саме система являється більш продуктивною для реалізації поставленого завдання. Головними етапами навчального процесу є такі етапи:

- надати перелік інформативних ресурсів користувачеві для вибору необхідного матеріалу;
- відобразити обраний матеріал у зрозумілому для користувача форматі.

Відомо, що будь-яка програма являє собою набір алгоритмів (компонентів), які взаємодіючи між собою вирішують поставлену задачу. При цьому програма буде програмною системою, якщо вона являє собою сукупність взаємопов'язаних компонентів, кожен з яких виконує цілком конкретні функції. У загальному випадку будь-яка навчальна програма може вважатися програмною системою, тому що в ній обов'язково присутня компонента інтерфейсу користувача, і компонента, що реалізує запропоновану методику. Автоматизованою навчальною системою буде будь-яка АНС, так як згідно, ряд завдань, наприклад відображення інформації або аналіз правильної відповіді,

виконуються без участі людини. Кожна АНС має певну структуру на основі групи елементів із зазначенням зв'язків між ними і дає уявлення про систему в цілому. Тому структура системи може бути охарактеризована за наявними в ній типами зв'язків.

На основі проведеного аналізу можна зробити висновок, що кожна навчальна система має чітко виражену структуру, і ці структури можна класифікувати наступним чином.

Для досягнення поставленої в роботі мети необхідно розробити принципи побудови відкритих мультиагентних систем підтримки прийняття рішень, які забезпечують оперативну ідентифікацію потреб та можливостей в процесі формування рішень, а також формалізувати взаємодію агентів в таких системах. Вирішення поставлених завдань дозволить розробити архітектуру мультиагентної системи, дослідити характеристики розроблених методів та взаємодії агентів, застосувати теоретичні розробки для побудови мультиагентної системи підтримки прийняття рішень в системі електронного навчання.

2 АНАЛІЗ МЕТОДІВ ПОБУДОВИ ЛОГІЧНИХ МЕРЕЖ

Формалізована модель мультиагентної системи електронного навчання. Ресурси є невід'ємною складовою організації навчального процесу як в традиційних освітніх системах, так і в сучасних системах електронного навчання. На відміну від традиційної системи освіти, де головними типами ресурсів є кадри та матеріальне забезпечення, в системах електронного навчання на перше місце виходять інформаційні ресурси, які є різними за змістом та формою, можуть бути розподіленими та зазвичай характеризуються динамічною потребою. Для ефективної роботи системи навчання необхідно реалізувати програмну систему, яка повинна забезпечити як функціонування так і завдання управління та координації процесів електронного навчання.

У порівнянні із традиційними, програмні системи, які базуються на агентному підході, мають певні переваги для вирішення складних слабо структурованих завдань управління [5].

По-перше, архітектура програмної мультиагентної системи являє собою велику мережу окремих автономних програмних агентів, що забезпечує простоту реалізації, відкритість, здатність до перенесення та масштабування.

По-друге, мультиагентна організація дозволяє реалізувати дійсно паралельне обчислення за рахунок незалежності роботи окремих програмних агентів.

По-третє, програмні системи, які основані на агентах, керуються системами на основі знань, на відміну від традиційних систем підтримки прийняття рішень, які керуються даними.

По-четверте, використання мультиагентної технології для вирішення складних завдань управління дозволяє отримувати розподілені рішення на відміну від централізованих механізмів традиційних програмних систем.

По-п'яте, мультиагентні системи, як клас систем штучного інтелекту, мають властивості самоорганізації та еволюціонування, що надає можливість використання переваг інтелектуальних технологій прийняття рішень.

Таким чином, сучасні тенденції розвитку інформаційних технологій та програмної інженерії пов'язані з використанням агентного підходу до вирішення різноманітних прикладних завдань автоматизації діяльності в різних галузях економіки.

Логічна мережа визначається як пристрій, призначений для розв'язування рівнянь алгебри предикатів, які були попередньо перетворені в двійкову систему. Логічна мережа складається з полюсів і гілок і має алгоритм, що описує їх взаємодію [4]. Всі логічні мережі мають однаковий алгоритм роботи, але склад полюсів і відгалужень залежить від моделі. Тому кожна логічна мережа будується за заданою моделлю. Така модель визначає довільне скінченне дискретне відношення неявно.

Як дискретний автомат з пам'яттю логічну мережу в загальних рисах можна описати наступним чином. Полюси мережі є елементами пам'яті, гілки мережі (кожна з яких містить пару протилежно спрямованих перетворювачів інформації), взяті разом, складають основну частину логічного перетворювача. До ЛП також входять елементи, які перетинаються і порівнюють множини. При цьому багато вхідних і вихідних сигналів вводиться і зчитується безпосередньо з елементів пам'яті (мережевих полюсів). Тому число входів логічної мережі завжди збігається з числом виходів логічної мережі. З точки зору визначеності логічна мережа є детермінованим автоматом (на відміну від імовірнісних автоматів).

Модель задається парою $\langle X, R \rangle$, де X – скінченна непорожня множина рівнянь АКП з однією суб'єктною змінною, яка представляє суб'єктні змінні моделі і області їх зміни; R – це набір рівнянь АСР з двома змінними, який описує бінарні зв'язки між змінними моделі. Модель $\langle X, R \rangle$ буде називатися моделлю логічної мережі. Множина X визначає полюси мережі, R гілки мережі. Загальна структура мережі може бути виражена неорієнтованим зв'язним графом з X вершинами і ребрами R , так як всі змінні моделі повинні бути пов'язані двійковими рівняннями.

Полюс мережі – це комірка пам'яті, яка в кожен момент часу містить підмножину значень з області зміни відповідної змінної. Гілка мережі - це пара функцій: від булевого значення області зміни x змінної до булевого значення області зміни y і навпаки. Кожна функція реалізує лінійний логічний оператор [5], ядром якого є характеристичний предикат бінарного відношення, що зв'язує змінні x і y . Такі оператори визначають повний образ або прототип множини відносно бінарного відношення.

Розглянемо два рівняння АКПП:

$$\begin{aligned}x^{(a_1)} \vee x^{(a_2)} \vee \dots \vee x^{(a_n)} &= 1 \\ y^{(b_1)} \vee y^{(b_2)} \vee \dots \vee y^{(b_m)} &= 1.\end{aligned}$$

Перше рівняння визначає предметну змінну x з областю зміни $N = \{a_1, \dots, a_n\}$, друге рівняння визначає змінну y з областю $M = \{b_1, \dots, b_m\}$. Припустимо рівняння двійкового рівняння АСР $P(x, y) = 1$, яке виражає двійкове відношення між змінними x і y . Тоді в термінах АКП формули лінійних логічних операторів, що описують гілку мережі, мають вигляд

$$\exists x \in N (P(x, y) A(x)) = B^{\wedge'}(y), \quad (2.1)$$

$$\exists y \in M (P(x, y) B(y)) = A^{\wedge'}(x), \quad (2.2)$$

Для $A(x)$ наведемо $A^{\wedge'}(x)$, $A, A^{\wedge'} \subseteq N$ характерні предикати множин; $B(y) \in B^{\wedge'}(y)$ характерними предикатами множин. $B, B^{\wedge'} \subseteq M$. Предикат $\in P(x, y)$ фіксованим, множиною A і B множиною довільно, множинами $B^{\wedge'}$ і $A^{\wedge'}$ визначаються формулами (2.1) і (2.2) відповідно.

Позначимо $P(N)$ через множину всіх підмножин (булевих значень) множини N . Тоді формулу (2.1) можна розглядати як опис функції виду $P(N) \rightarrow P(M)$, що визначає повний образ множини $A \in P(N)$ щодо двійкового відношення, $P \subseteq N \times M$ формулу (2.2) як функцію виду $P(M) \rightarrow P(N)$, що визначає повний прототип множини $B \in P(M)$ відносно такого ж ставлення. Перетворення виду (2.1) і (2.2) називаються лінійними логічними операторами [4]. Задана рівнянням гілка логічної мережі $P(x, y) = 1$ реалізує цю пару операторів.

Функціонування логічної мережі, яка виконує, наприклад, обробку текстових повідомлень природної мови, зводиться до наступного: користувач

вводить значення деяких ознак словоформи (які можуть бути задані неоднозначно), а мережа визначає інші властивості словоформи (з точністю вихідних даних). кожна ітерація називається кроком або «тіком» [МУ]. Зупинка виникає, коли стан мережі повторюється на наступному такті.

У [4] розвинена і описана логічна мережа. Загальна структура цієї логічної мережі показана на рис. 1.1.

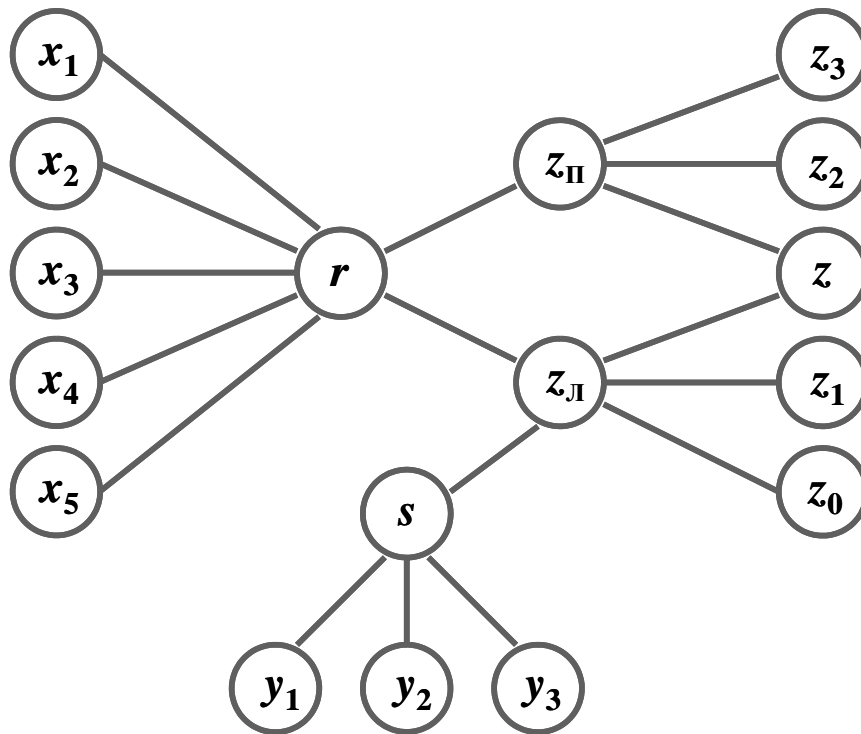


Рисунок 2.1 – Загальна структура логічної мережі

Розглянемо приклад функціонування цієї логічної мережі за допомогою тиків. Для прикладу функціонування цієї мережі представлено в наочній формі для ознайомлення не підготовленого читача з принципами роботи логічних мереж. Тому подробиці опису моделі логічної мережі відмінювання прикметників не наводяться.

Робота $x_3 \in y_1 \in y_2 \in z_2$ логічної мережі відповідно до алгоритму показана на рис. 2.2, 2.3, з затемненими кружечками, що позначають полюси, куди були введені вихідні дані, або де відбувалася зміна станів полюсів (стан «б»).

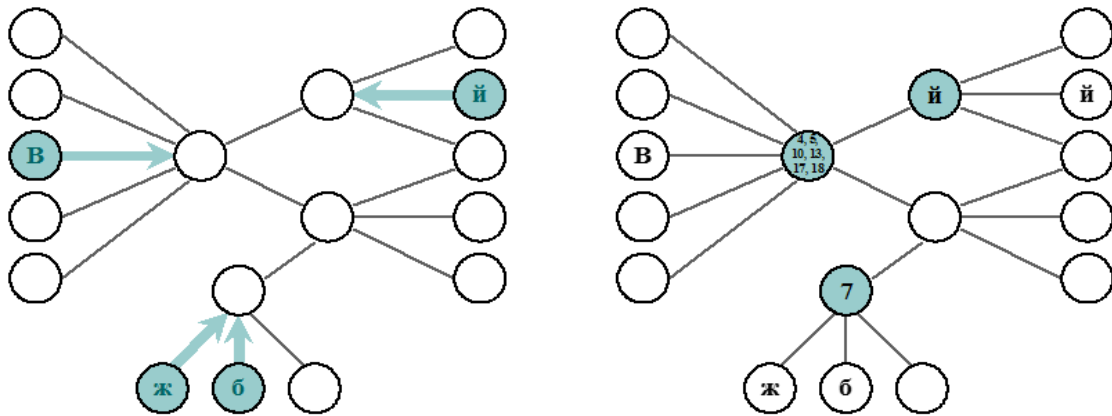


Рисунок 2.2 – Перший такт

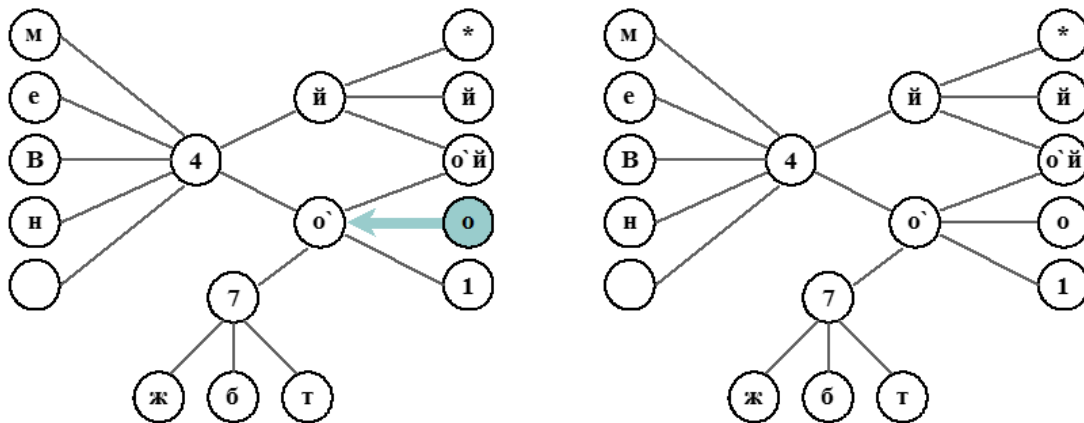


Рисунок 2.3 – П'ятий такт

Після виконання п'ятого тикю не відбулося жодної зміни станів мережевих полюсів, тому мережа перестала працювати. Рішення проблеми міститься в полюсах.

Алгоритм роботи логічної мережі зручно описувати за допомогою наступних термінів. Джерелами цього циклу є ті полюси мережі, в яких стан змінився за попередній цикл. Активними на даній планці є всі ті функції, які починаються біля витоків цього циклу. Приймачами на цьому циклі є ті полюси, на які спрямовані активні функції (тобто всі сусіди джерел). На кожному циклі змінюється розташування джерел, приймачів і активних функцій.

Установка початкових станів полюсів вважається нульовим циклом. Джерелами на першій галочці будуть всі ті полюси, в які користувач ввів інформацію на нульовій галочці. Кожен такт включає в себе наступні дії:

- для кожного джерела обчислюються всі вихідні функції, які з ним пов'язані. Аргументом цих функцій є стан даного полюса;
- для кожного приймача обчислюється новий стан: формується перетин старого стану і значень всіх активних вхідних функцій;
- для кожного приймача порівнюються попередні і нові стани: ті з приймачів, для яких стани не збігаються, будуть джерелами в наступному циклі. Якщо набір джерел для наступного циклу порожній (тобто є збіг для всіх приймачів), то мережа перестає працювати і рішення знайдено.

Рішенням є стани всіх полюсів мережі після завершення алгоритму. Алгоритм логічної мережі можна віднести до сімейства хвильових алгоритмів.

У програмному забезпеченні має сенс реалізовувати тільки логічні мережі з відносно невеликою кількістю полюсів і відгалужень. Будь-яка програма на персональному комп'ютері виконує обхід полюсів і гілок на кожному такті послідовно. Зі збільшенням числа полюсів і відгалужень швидкість обробки мережі буде знижуватися. Коли кількість мережевих полюсів досягне порядку 10^4 (а це далеко не межа), швидкість роботи програми, що моделює таку логічну мережу, навряд чи буде прийнятним. Тим часом багато проблем інтелектуальної обробки природної мови потребують вирішення в режимі реального часу. Наприклад, при організації діалогу з комп'ютером на природній мові завдання аналізу і синтезу тексту повинні вирішуватися практично миттєво, так само, як це робить людина.

ЛС як дискретний автомат з пам'яттю. З точки зору класифікації пристроїв логічною мережею можна назвати дискретні (або скінченні) автомати з пам'яттю [5]. Нижче наводиться визначення дискретного автомата, взяте в усіченому вигляді.

«Дискретний автомат» – це пристрій для перетворення дискретної інформації. Це орієнтований (m, n) полюс, який може мати k елементи, що

виконують функції зворотного зв'язку і є елементами пам'яті. Частина структури дискретного автомата, що не має контурів зворотного зв'язку (елементи пам'яті), називається логічним перетворювачем (ЛП), а сукупність елементів пам'яті – пам'яттю (Р).

Основною якістю, що відрізняє дискретні автомати від всіх інших перетворювачів інформації, є наявність дискретного (завжди кінцевого в реальних автоматах) набору внутрішніх станів і властивість стрибкоподібного переходу автоматів з одного стану в інший (перше припущення). Друге припущення полягає в тому, що після переходу автомата в довільний стан перехід в наступний стан можливий не раніше, ніж через деякий дискретний проміжок часу $\tau > 0$, званий інтервалом дискретності автомата. Це припущення дозволяє розглянути функціонування дискретного автомата в так званому дискретному часі. Для синхронних автоматів моменти часу, в які можлива зміна внутрішнього стану, визначаються спеціальним пристроєм – генератором синхронізуючих імпульсів. При цьому суміжні моменти часу розділені рівними проміжками.

Можна підкреслити такі особливості мультиагентного підходу як:

- функціонування програмної системи будується залежно від подій, які відбуваються в реальному часі;
- програмні агенти працюють асинхронно та квазіпаралельно, аналізуючи і перебудовуючи поточний стан шляхом встановлення або перегляду зав'язків, що призводить до їхньої самоорганізації;
- рішення агентами приймаються еволюційно, при цьому можливо змінюються прийняті раніше рішення.

Такий підхід дозволяє розглядати отримане рішення як «нестійку рівновагу» і спостерігати такі феномени поведінки складних систем, як порядок і хаос, катастрофи та інші нелінійні ефекти. При цьому структура і логіка роботи окремих програмних агентів вкрай прості, але вони демонструють надзвичайно складну поведінку системи в цілому, фактично утворюючи

емерджентний інтелект. На відміну від традиційного, коли проводиться пошук деякого чітко визначеного алгоритму, що дозволяє знайти найкраще вирішення проблеми, в мультиагентних технологіях рішення виходить автоматично в результаті взаємодії множини самостійних програмних модулів (агентів).

Програмні агенти функціонують у віртуальному світі і взаємодіють між собою шляхом посилки повідомлень. Як результат взаємодії агентів, формується поточне рішення проблеми, яке гнучко змінюється відповідно до динаміки середовища. Зазвичай взаємодія агентів моделюється як переговори або аукціон [7]. Одним з можливих різновидів переговорів агентів є торгівля, коли існують агенти потреб і агенти можливостей (наприклад, ресурсів), між якими відбуваються переговори, в рамках віртуального ринку.

Систему електронного навчання будемо розглядати як відкриту систему, яка об'єднує розподілені інформаційні навчальні ресурси та до якої постійно звертаються споживачі. При цьому запити щодо інформаційних ресурсів виникають постійно, але потреба в ресурсах змінюється динамічно, та не може бути передбачена заздалегідь. Тому для формалізації розподілу ресурсів у системі електронного навчання автор пропонує використати модель мережі, яка формується динамічно та у кожний момент часу відображає взаємозв'язок між потребами та можливостями системи.

Визначимо два типи автономних агентів, що взаємодіють у системі – агенти-потреби (АП) та агенти-можливості (АМ). Усі агенти діють відповідно до власних цілей та за певними правилами, що дозволяє їм діяти самостійно та взаємодіяти один з одним. На множині агентів можна визначити відношення, які пов'язують агентів. Залежно від типу та складності задачі, яка вирішується, можна визначити різноманітні відношення. В даній роботі розглядається найпростіше відношення між двома типами агентів – це «відповідність».

3 РОЗРОБКА ЛОГІЧНОЇ МЕРЕЖИ ВЗАЄМОДІЇ З ШТУЧНИМИ АГЕНТАМИ

У загальному вигляді запропонована модель мультиагентної системи функціонує наступним чином. У кожний момент часу система формує набір можливостей – це ресурси, які є вільними на цей момент. В систему поступають заявки на використання ресурсів – це потреби. Усі потреби та можливості в системі представляються як автономні агенти, які шляхом переговорів встановлюють між собою відношення відповідності. У разі коли агент- потреба та агент-можливість встановили відношення «відповідність», вони перестають бути вільними та не приймають участь у подальших переговорах з іншими агентами. Множина агентів обох типів та зв'язки між ними формують у кожен момент часу стан динамічної системи розподілу ресурсів в системі електронного навчання.

Формально таку мультиагентну систему можна представити множиною

$$G = \{A, R, M, V\}, \quad (3.1)$$

де A – множина агентів (АП та АМ);

R – множина можливих відношень між агентами;

M – множина правил прийняття рішень та встановлення або розірвання зв'язків між агентами;

V – множина цілей, згідно яких діють агенти.

Наявність логічного аналізу і прийняття рішень в тестовому завданні моделює поведінкове правило екзаменатора. Прийнятий ситуаційний підхід ґрунтується на правилах для конкретного поведінкового питання (передача контролю на інше питання) в рамках одного завдання. Це дозволяє враховувати не тільки елементарний набір результатів відповідей (істинно чи хибно), а й те, яка відповідь з набору неправильних відповідей обрана з урахуванням стану протоколу іспиту перед взаємодією з конкретним питанням.

Додаткові питання можуть бути обрані з основних будь-якого рівня

складності та/або створені окремо (під конкретний тест). Рівень складності повинен бути призначений як для них, так і для основних. Таким чином, можна узагальнити питання в наборі тестових завдань у вигляді структури (класу) даних і визначити їх поведінку. Питання повинні бути:

- властивість, що визначає, чи є питання основним або додатковим;
- властивість, що визначає рівень складності;
- властивість, що визначає приналежність і зв'язок з іншими темами і проблемами;
- властивість, що визначає питання, з якими було розірвано з'єднання в процесі тестування;
- властивість, що визначає, чи здатна неправильна відповідь перервати процес тестування;
- властивість, що визначає зв'язки (ті питання, на які можна переключитися), обумовлені умовиводом;
- властивість, що визначає зв'язки (ті питання, на які можна переключитися звичайним для логічних мереж способом);
- властивість «зупиняти» питання в логічній мережі, тобто випадок, коли питання вже поставлено;
- властивість, що визначає, чи є питання унікальним для даної мережі;
- властивість, яка визначає, яке інше питання (або питання) може бути викликане поточним, тобто яке питання може отримати контроль.

Обов'язковими є такі положення:

- забезпечити приймання контролю з попереднього питання з отриманням чинного протоколу обстеження;
- переконатися, що тестове завдання та варіанти відповідей надані;
- забезпечити, щоб відлік часу розпочався в момент видачі завдання;
- підтримувати логічну функцію, що визначає зниження рівня складності у разі перевищення лімітів часу, відведених для відповіді на дане запитання

та/або прийняття рішення (за певних умов) щодо поточного питання про правильність/неправильність отриманої відповіді;

– затвердити рішення про продовження випробування або його припинення;

– прийняти рішення і вибрати наступне питання за заданим стохастичним алгоритмом або зробити висновок, чи активна властивість «стоп» для цього елемента;

– передати протокол обстеження на наступне питання і на мережевий контролер;

– передавати властивості, що визначають стан питання, мережевому контролеру;

– перевірити працездатність питання (очікується, що управління буде передано мережевому контролеру або наступному питанню).

На рисунку 3.1 зображено фрагмент логічної мережі запитань (основне питання поряд з додатковими).

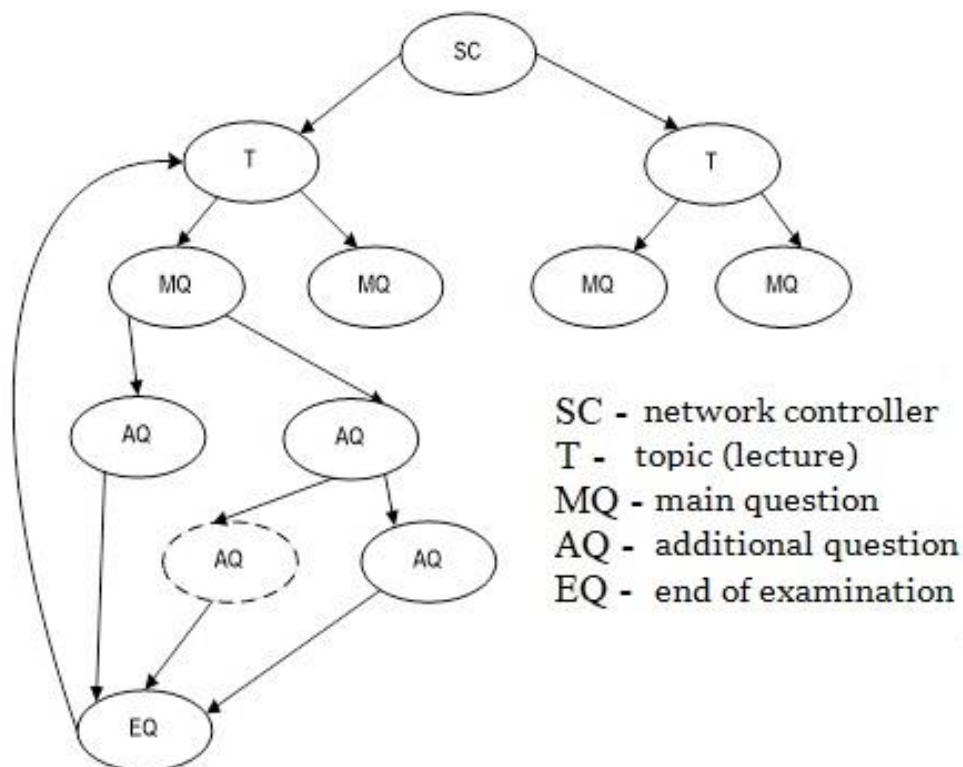


Рисунок 3.1 – Фрагмент логічної мережі

Використання логічних мереж, заснованих на скінченному численні предикатів першого порядку та операціях над предикатами, передбачає існування моделі системи як абстракції, що має множину станів $S = \{s_1, \dots, s_n\}$. Ці стани взаємовиключні і перехід з одного стану мережі в інший здійснюється по-різному, що обумовлено виникненням події, яка в загальному випадку являє собою предикат події. Кожен стан системи s_i повинен мати як мінімум на вході і один вихід, щоб були відсутні так звані «заборони» або кінцеві стани. По суті, в якості початкового може бути обраний будь-який стан системи.

Однак на практиці вибір початкового стану визначається апріорною інформацією про модель студента. Після ініціалізації деякого початкового стану робота моделі, створеної в заданих рамках алгебраїчної системи логічної мережі, заснована на нескінченному обході мережі станів системи, в якій вона залишається протягом певного періоду.

Слід зазначити, що пряме використання застосування логічного мережевого представлення не дає розв'язання даної задачі, оскільки існує ряд логічних обмежень, які визначають перехід з одного стану в інший і видалення стану, а також обмеження переходів (перевірка не може виконуватися безперервно і питання не повинні повторюватися).

Останній метод є кращим, оскільки перевіряється, чи сформована множина заданої характеристики (наприклад, складність питань для всієї групи учнів і чи формується набір предикатів для розподілу відповідей залежно від їх складності для конкретного учня). Крім того, зростання складності алгоритму тестування не призведе до значного зростання споживання ресурсів.

Різниця між рівнями складності основного і додаткового питань і запропонований зв'язок між основними питаннями і відгалуженнями додаткових, дозволяє мінімізувати кількість відповідей учнів, необхідних для визначення рівня їх знань, а також значно поліпшити адаптивні якості тесту.

Найкращі результати досягаються тоді, коли різниця між прогнозованим і

фактичним рівнем знань групи учнів і різниця між прогнозованою і фактичною складністю завдань є значною.

Однак, при невеликій різниці між вищезазначеними прогнозованими і фактичними характеристиками досліджуваної групи, застосування запропонованих методів дає позитивний ефект, з якого можна зробити висновок, що застосування цих методів створення логічної мережі для реалізації етапу тестування знань в адаптивних тестових системах є ефективним для будь-яких потужностей та інших характеристик, які визначають тестування як процес.

У загальному вигляді така мультиагентна система представляє собою орієнтований граф, де вершини – це агенти, а дуги – це відношення між агентами. Таке представлення дозволяє використовувати весь математичний апарат теорії графів, а також задавати будь-які відношення для моделювання певної задачі управління в системі електронного навчання.

Загальна схема взаємодії агентів. З метою спрощення моделі встановимо наступні обмеження (рис. 2.5). По-перше, в системі функціонує два типи агентів – АМ і АП, між якими можливо встановити тільки один тип відношень – «Відповідність». По-друге, усі агенти взаємодіють в межах віртуального середовища (ринку), де вони можуть шукати відповідних агентів, проводити торги та встановлювати зв'язок, утворюючи відповідну пару. По-третє, кожна потреба та кожна можливість характеризується вектором індивідуальних особливостей, відповідно до яких агенти шукають собі пару. По-четверте, усі агенти мають власну функцію корисності, яку вони прагнуть максимізувати.



Рисунок 3.2 – Варіанти використання агентів

Однією з проблем, що виникають в системі електронного навчання, є проблема координації. Частково вирішення цієї задачі можливо при деталізації множини правил прийняття рішень агентами [5], [6]. У загальному випадку можливі такі види координації:

- координація відносно задачі, яка вирішується на верхньому рівні управління;
- координація відносно задач, які вирішуються в кожній з підсистем ієрархічної системи управління;
- координація відносно компромісного значення цільових функцій підсистем ієрархічної системи управління.

Тоді загальна координаційна задача трансформується таким чином: потреби з характеристиками h , при чому пропозицію надають усі АМ, які мають необхідні характеристики h у повному обсязі або частково.

Крок 3. Прийняття рішення АП згідно власної функції корисності та встановлених правил прийняття рішень.

Крок 4. Прийняття рішення АМ, якщо є конкуруючі заявки, то зв'язок встановлюється з тим АП, для якого функція корисності АМ буде мати найліпше значення.

Крок 5. Встановлення зв'язку між АП та АМ, якщо обидва агенти дійшли згоди, не мають ліпших пропозицій та повністю згодні з умовами.

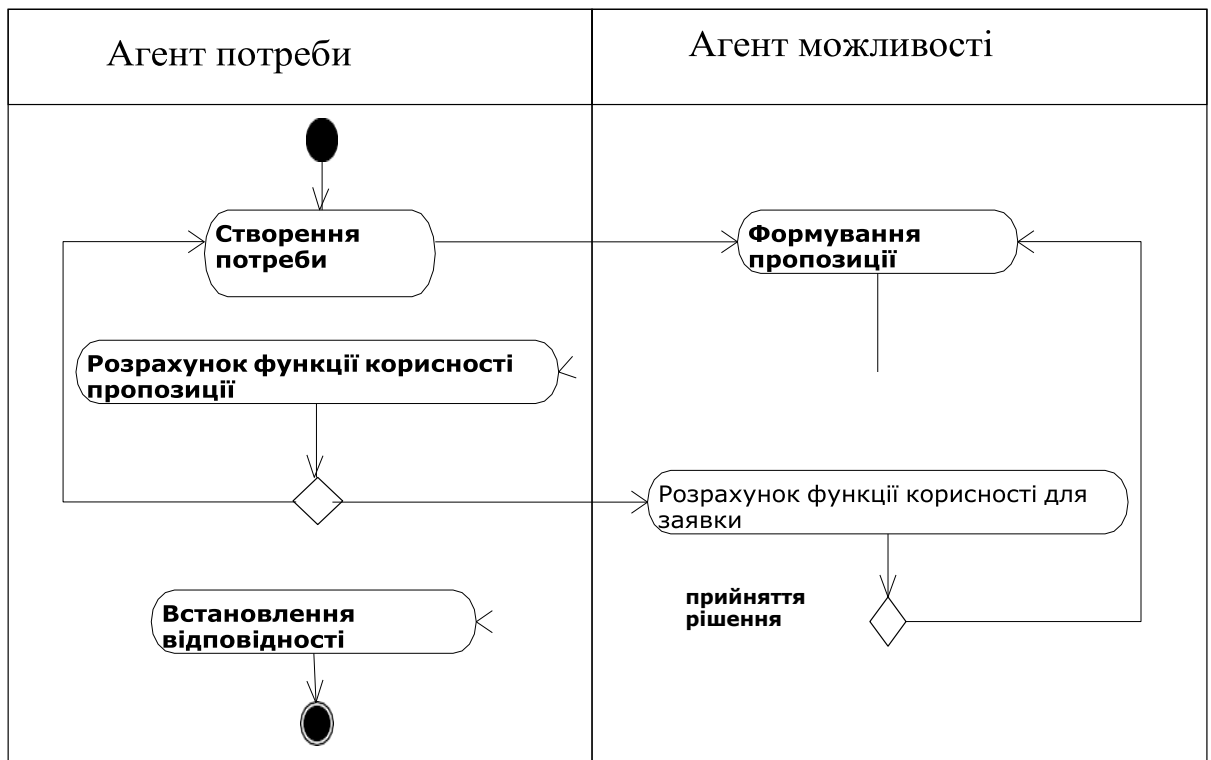


Рисунок 3.3 – Взаємодія агентів

Крок 6. Розрив зв'язку виконується після повідомлення агентом про відмову та підтвердження іншою стороною згоди, або у разі коли потреба або ресурс видаляється із системи, то агент відправляє повідомлення про розрив зв'язку, а зв'язаний з ним агент починає знову шукати відповідну пару.

Крок 7. Процес виконується до тих пір, доки усі потреби не будуть задоволені та розраховується загальносистемна вартість.

Для реалізації запропонованого нами підходу на основі мережі взаємодії агентів-потреби та агентів-можливості у системі електронного навчання

необхідно розробити мову взаємодії агентів, яка базується на визначеній онтології предметної галузі.

Виділено наступні спеціалізовані компоненти для роботи агентів в системі.

Компонент комунікації агентів – забезпечує взаємодію агентів, встановлення зв'язку між двома агентами, перевіряє допустимість варіантів рішень.

Компонент прийняття рішень – динамічно формує можливі варіанти прийняття рішень, оцінює рішення за встановленими критеріями та дозволяє обрати найліпший варіант.

Компонент потокових розрахунків – дозволяє проводити розрахунки значень атрибутів агентів, які необхідні для встановлення зв'язку та прийняття рішень, враховуючи взаємозв'язок понять з предметної онтології.

Описаний вище підхід може бути адаптований до вирішення будь-якого завдання управління ресурсами в системі електронного навчання. Запропонований метод дозволяє гнучко та оперативно реагувати на будь-які зміни у відкритій системі управління або у зовнішньому середовищі.

Крім переваг застосування мультиагентного підходу для вирішення завдань оперативної обробки даних має декілька суттєвих невирішених питань, а саме:

- достатньо важко оцінити рішення щодо його оптимальності;
- рішення чутливі до історії подій;
- невеликі зміни на вході системи можуть привести до суттєвих змін на виході;
- можливі затримки часу на вирішення завдання як наслідок великих ланцюгів змін;
- можливо отримати неідентичні рішення за умов однакового входу при повторному запуску моделі;
- певні труднощі при корегуванні рішення у

- «ручному» режимі;
- можливі непорозуміння при поясненні результатів як наслідок складних причинно-наслідкових зв'язків.

Перелічені проблеми потребують вирішення з метою реалізації мультиагентної системи та є напрямками майбутніх досліджень.

4 ОПИС ПРОГРАМНИХ РІШЕНЬ

Найбільш типовим рішенням щодо гнучкої архітектури є компонентно-орієнтована архітектура (рис.4.1), що запропонована в даній роботі.

Організація агентної системи електронного навчання надає переваги щодо індивідуалізації та адаптації навчального процесу до потреб користувача. Загалом можна виділити такі складові електронної навчальної системи: пошук навчального контенту, доставка навчального контенту, збереження, формування навчальної траєкторії, контроль, інтерфейс користувача та інше.

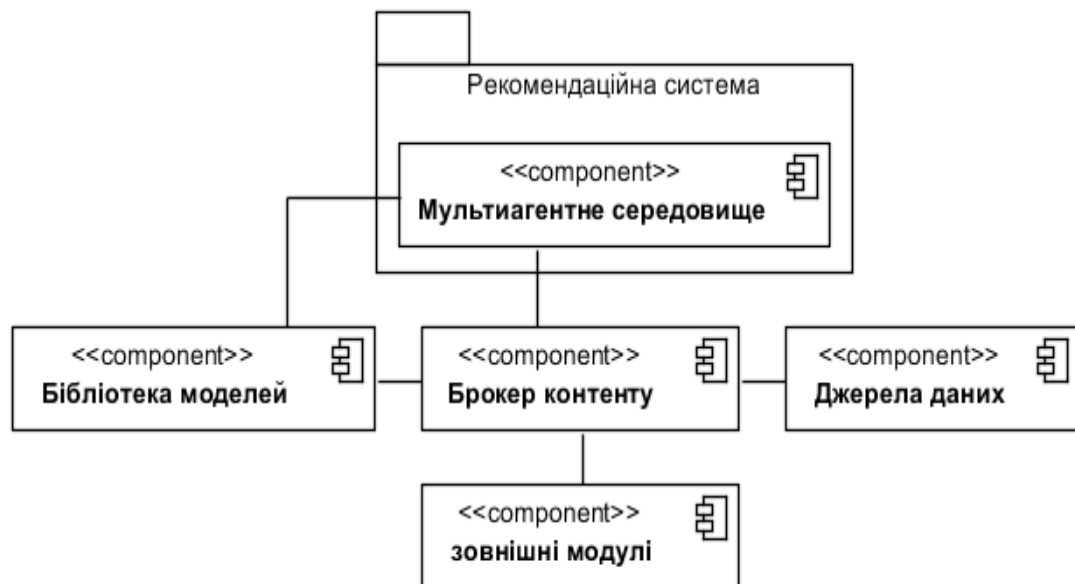


Рисунок 4.1 – Архітектура колаборативної системи

Користувачі колаборативної системи електронного навчання прагнуть використовувати матеріали із численних джерел. Для забезпечення цього повинні існувати механізми, які дозволяють знаходити та використовувати необхідні матеріали як в локальних архівах, так і в мережі Inretnet.

Пошук навчального контенту для інтеграції та багаторазового використання є одним з важливих завдань вбудованих інтелектуальних інформаційних агентів. Сучасні дослідники поділяють проблему пошуку навчальних об'єктів на два типи задач:

- пошук навчальних ресурсів у єдиному локальному архіві;

- розподілений пошук навчальних об'єктів у мережі Інтернет.

Вирішення першої задачі призвело до певних теоретичних та практичних результатів [5], що дозволяє впроваджувати їх застосування в системах підтримки електронного навчання. Друга проблема поки що не має ефективного вирішення на основі узагальненого теоретичного підходу. Це призводить до необхідності дослідження процесів пошуку даних у мережі Інтернет, а також розробки теоретичних засад вирішення задачі пошуку навчальних об'єктів.

Рекомендаційна система багатоагентного типу дозволяє агрегувати набір автоматизованих та орієнтованих на користувача підходів для контролю за семантичними ресурсами. Різні класи агентів і методи їх взаємодії розглядалися в роботах багатьох науковців.

Одним з ключових понять, що характеризує вибір тієї чи іншої моделі аналізу текстової інформації, а також реалізацію конкретного варіанту пошуку, є модель пошуку. Модель пошуку – це поєднання наступних складових:

- спосіб подання документів;
- спосіб подання пошукових запитів;
- вид критерію релевантності документів.

Варіації цих складових визначають велике число різних реалізацій систем текстового пошуку. Розглянемо деякі з них, найбільш популярні в даний час. Найпростіші моделі пошуку – це моделі, в яких документ представляється у вигляді набору асоційованих з ним зовнішніх атрибутів. До логічних моделей пошуку належить модель дескрипторного пошуку. У найпростіших системах дескрипторного пошуку уявлення документа описується сукупністю слів або словосполучень лексики предметної області, які характеризують зміст документа. Ці слова і словосполучення називаються дескрипторами. Індукування документа в таких системах реалізується призначенням для нього сукупності дескрипторів. У деяких дескрипторних системах індукування документів здійснюється вручну експертами в предметній галузі, в інших воно виконується автоматично.

Документ в даній моделі представляється у вигляді сукупності

асоційованих з ним атрибутів. Атрибутами є ідентифікатори класів, до яких відноситься даний документ. Класи формують ієрархічну структуру класифікатора. Запит може бути представлений двома способами:

Запитом є ідентифікатор-якого класу із заданого класифікатора. Критерій релевантності документа запитом – клас документа збігається з класом у поданні запиту або є його підкласом. У запиті можна вказати декілька класів класифікатора. Критерій релевантності документа запитом – клас документа збігається з яким-небудь із зазначених у запиті класів або є його підкласом.

Моделі, основані на класифікаторах, близькі до булевих моделей. У булевських моделях пошуку користувач може формулювати запит у вигляді булевського вираження, використовуючи для цього оператори І, АБО, НІ. Терми запиту залежать від конкретного варіанту моделі пошуку. У булевої моделі, орієнтованої на пошук "по тексту", термами будуть слова, відповідно, критерієм релевантністю буде умова входження деякого слова або словосполучення в текст документа. У булевій моделі, орієнтованої на пошук за класифікаторами, термами вираження будуть ідентифікатори класів класифікатора. У булевій моделі пошуку з використанням Дублінського ядра термом буде значення елементів метаданих. Документ, що має співпадаючі значення елементів метаданих зі значеннями, заданими у запиті, вважається релевантним.

У загальному випадку критерієм релевантності документа запиту в булевих моделях пошуку є істинність булевського вираження, заданого в запиті.

Однією з безперечних переваг булевої моделі пошуку є простота її реалізації. Головними недоліками вважаються [16]:

- відсутність можливості ранжирування знайдених документів за ступенем релевантності, оскільки відсутні критерії її оцінки;
- складність використання: далеко не кожен користувач може вільно оперувати булевими операторами при формулюванні своїх запитів.

Варто відзначити, що були спроби розвитку булевої моделі пошуку для забезпечення можливості ранжирування множини документів, що видаються користувачеві. А саме, запропоновано кілька варіантів так званих розширених

булевих моделей. У цих моделях вводяться спеціальні узагальнення булевих операторів для додавання підвищеної ваги документам, які задовольняють запиту точно, і знижену вагу – всім іншим документам.

Наразі векторні моделі є найпоширенішими і вживаними на практиці моделями пошуку [4, 14]. Векторні моделі, на відміну від булевих, дозволяють ранжувати результуючу множину документів запиту. Суть таких моделей зводиться до подання документів і запитів у вигляді векторів. Кожному терму t_i в документі d_j і запиту Q співставляється деяка ненегативна вага w_{ij} (w_i для запиту). Таким чином, кожен документ і запит може бути представлений у вигляді k -мірного вектора:

$$\vec{d}_j \stackrel{\text{def}}{=} (w_{1j}, w_{2j}, \dots, w_{kj}),$$

де k – загальна кількість різних термів у всіх документах.

Згідно векторної моделі, близькість документа d_i до запиту Q оцінюється як кореляція між векторами їх описів. Ця кореляція може бути обчислена, наприклад, як скалярний добуток відповідних векторів описів. Варіації різних способів призначення ваг термів та оцінки міри близькості векторів визначають широкий спектр різних модифікацій даної моделі пошуку.

Вперше ідеї ймовірнісних моделей були запропоновані в 1960 році. В їх основі лежить принцип ймовірнісного ранжування. Цей принцип полягає в наступному: найвища загальна ефективність пошуку досягається у випадку, коли результуючі документи ранжуються за зменшенням ймовірності релевантності запиту. Спочатку для кожного документа оцінюється ймовірність того, що він релевантний запиту, а потім за цими оцінками виконується ранжування документів.

Існують різні способи отримання цих оцінок, а також додаткові припущення і гіпотези на основі апріорних відомостей щодо документів колекції, які і визначають конкретну реалізацію ймовірнісної моделі пошуку.

Наприклад, ця оцінка може бути обчислена відповідно до теореми Байєса.

Так само, як і ймовірнісні моделі, мережі виведення основані на принципі ймовірнісного ранжирування результуючих документів пошуку [4, 14]. Головна їхня відмінність полягає в тому, що використовується оцінка не ймовірності релевантності документа запиту, а ймовірності того, що він задовольняє інформаційним потребам користувача. У рамках даної моделі процес пошуку документів описується як процес міркувань в умовах невизначеності. У процесі такого міркування оцінюється ймовірність того, що інформаційні потреби користувача, виражені за допомогою одного або декількох запитів, задовольняються.

Таким чином, існуючі моделі інформаційного пошуку надають інструментарій, який може бути реалізований в пошуковій системі. Враховуючи наявність стандартів опису електронних навчальних ресурсів, які є розширенням моделі Дублінського ядра, в даній роботі запропоновано використання дескриптивної моделі пошуку. Для формалізації даної моделі та використання її для реалізації пошукових агентів, доцільно розглянути методологію інтелектуальної обробки даних, зокрема методи теорії інтелекту [17].

Таким чином, мета статті полягає в дослідженні особливостей інформаційного пошуку навчального контенту у веб-просторі та аналізу можливості застосування методу компараторної ідентифікації для реалізації концепції агентної системи інтелектуального пошуку навчальних об'єктів в системах електронного навчання.

Навчальний об'єкт – це самодостатні інформаційні одиниці, які можуть бути використані для навчальних цілей та мають теги у якості внутрішніх зв'язків [2]. Створення навчального курсу вимагає зіставлення в одне ціле обумовленої послідовності навчальних об'єктів. Для зберігання навчальних курсів використовують різноманітні навчальні репозитарії (архіви, інформаційні портали). До особливостей навчальних об'єктів можна віднести здатність до багаторазового використання, можливість агрегування або декомпозиції відповідно до навчальних цілей та згідно метаопису, відповідність стандартам.

В цій роботі пропонується архітектура інформаційної системи інтелектуального пошуку навчальних матеріалів на базі мультиагентної системи (МАС). Агенти спілкуються один з одним через службу транспортування повідомлень. Робота агентів координується та контролюється службою імен та сервісом авторизації. Каталог агентів необхідний для інформування агентів про оточуюче середовище, тобто про усіх агентів, їхні функції та цілі. Онтологія необхідна для того, щоб забезпечити спільний опис предметної області для різних агентів. Дані, необхідні для функціонування інформаційної системи, зберігаються у базі даних та знань.

На концептуальному рівні будемо розглядати наступні ситуації використання МАС пошуку навчальних об'єктів (рис. 2):

- пошук (метаданих, документації, навчальних матеріалів) у мережі;
- перегляд змісту електронних навчальних ресурсів;
- редагування пакетів;
- збір та зберігання пакетів.

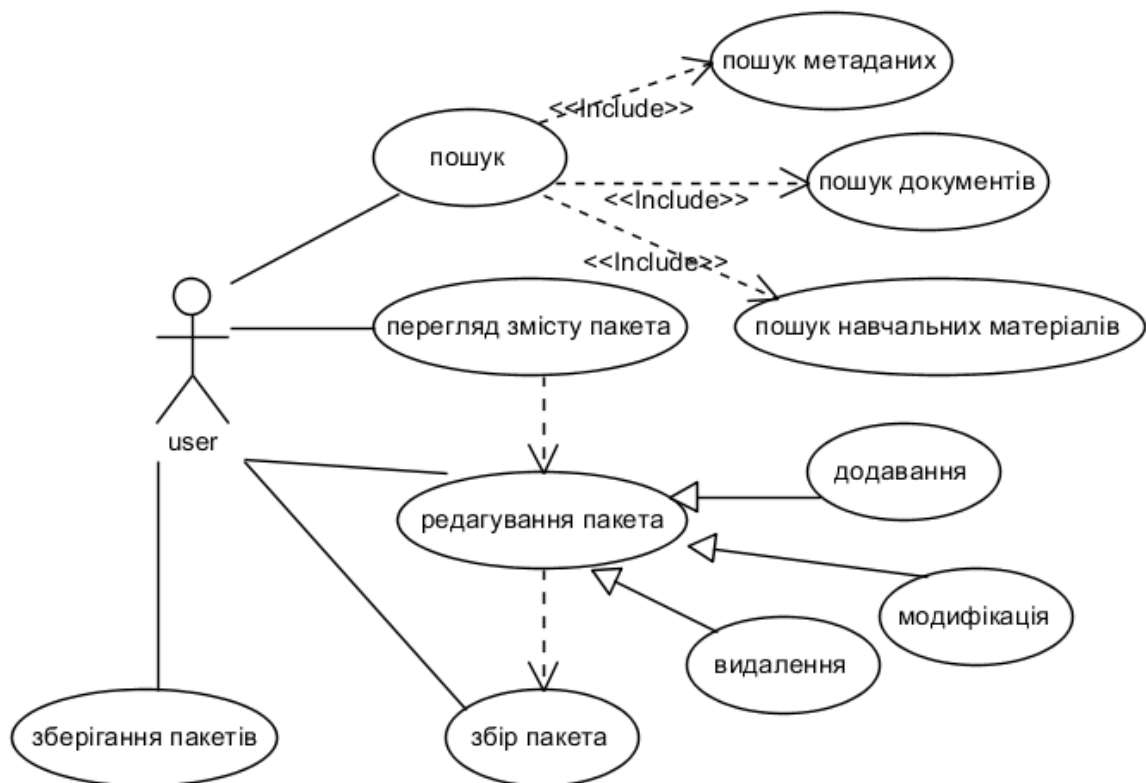


Рисунок 4.2 – Варіанти використання

Пропонується мультиагентна пошукова система, яка побудована на

агентах чотирьох типів: краулер, оцінювач, індексатор та координатор (рис. 4.3).

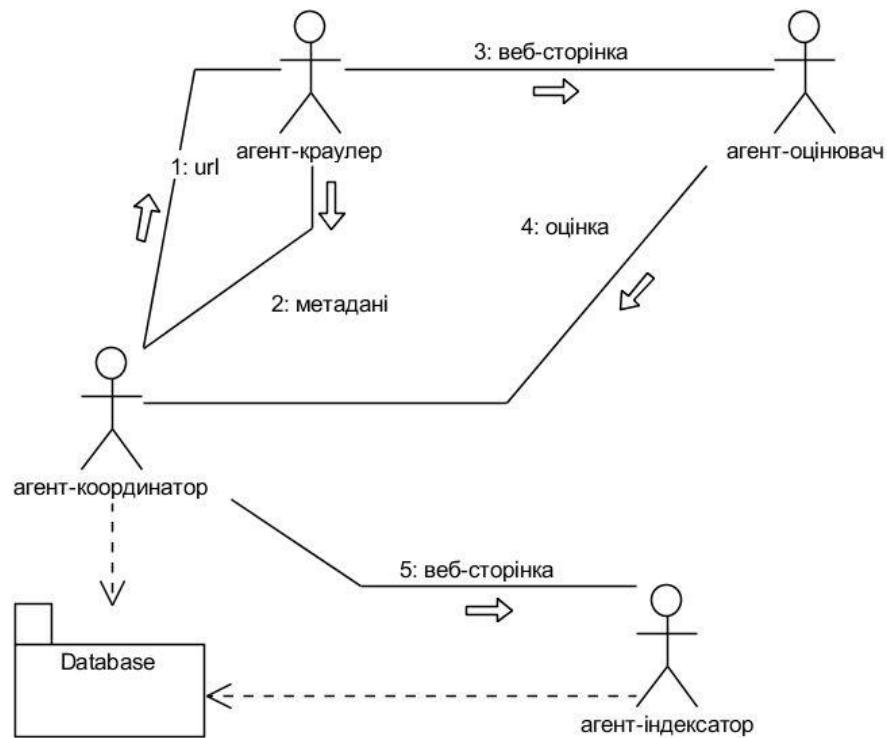


Рисунок 4.3 – Структура пошукової системи

Агент-краулер здійснює перехід на задану агентом-координатором url-адресу, видобуває метадані, якщо вони є наявні, додаткові дані та передає ці дані агенту-оцінювачу. Агент-оцінювач застосовує компаратор, побудований на основі моделі предметної області та дескрипторної моделі пошуку, що надає ефективний механізм для визначення перспектив подальшого пошуку з оцінюваної сторінки. Головна задача агента-індексатора полягає у автоматичному формуванні метаданих, які описують знайдений документ відповідно до обраної моделі.

Процес пошуку навчальних матеріалів можна розглядати у двох основних напрямках використання: пошук нових навчальних ресурсів (рис. 4.4) та пошук навчальних матеріалів за наявними метаданими (рис. 4.5).

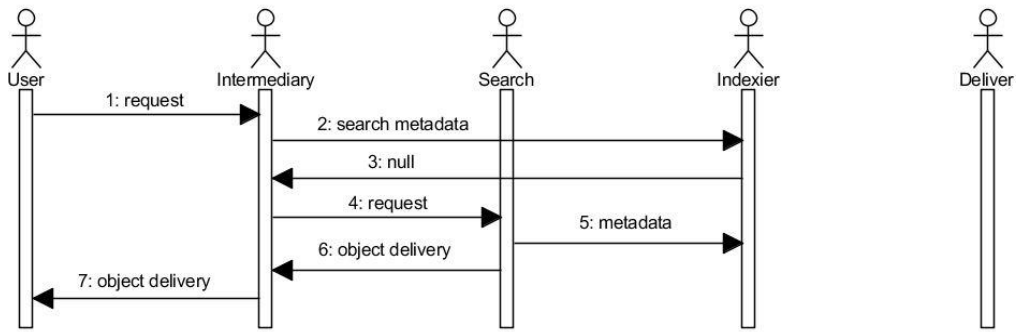


Рисунок 4.4 – Випадок використання: пошук нового навчального об’єкта

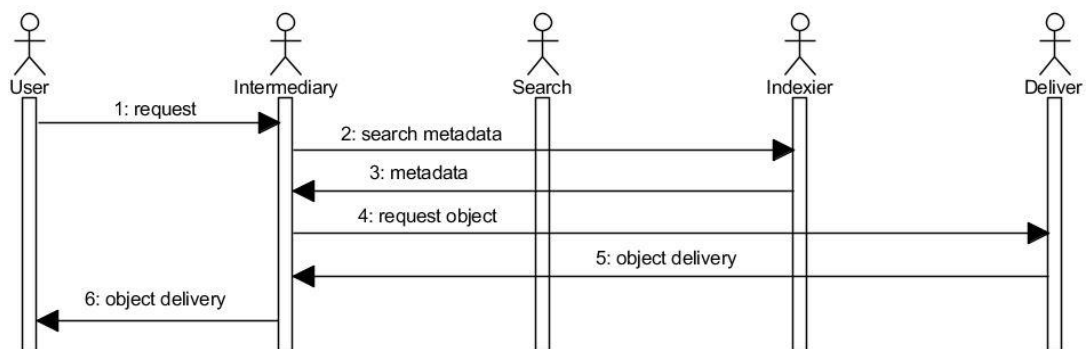


Рисунок 4.5 – Пошук навчального контенту за метаданими

У випадку пошуку навчального контенту за метаданими задачу пошуку можна розділити на такі кроки: 1) пошук архіву навчальних матеріалів у мережі Internet; 2) пошук навчального контенту у одному репозитарії; 3) пошук у веб-просторі електронних ресурсів, які можуть бути використані для навчальних цілей. Перша задача може розглядатися як задача пошуку тематичних спільнот у веб-просторі, Друга задача – питання щодо пошуку навчальних ресурсів в одному репозитарії. Третя задача, з точки зору пошуку навчального контенту, має певні особливості.

По-перше, відповідно до рекомендацій стандартів (LOM, SCORM) навчальні електронні ресурси характеризуються певним набором метаданих, які описують ці ресурси відповідно до навчального контенту, але, з іншого боку, ці метадані мають гетерогенну природу та є слабоформалізованими.

По-друге, зазвичай стандарти носять рекомендаційний характер та

надають певну свободу розробникам навчальних ресурсів, що призводить до неточності, невизначеності та неоднозначності при автоматичній обробці метаданих.

По-третє, існують певні труднощі щодо встановлення відповідності між пошуковим запитом та значенням метаданих.

Все це обумовлює необхідність досліджень у напрямку вдосконалення пошукових алгоритмів у системах підтримки електронного навчання. У даній роботі пропонується використання концепції семантичного пошуку за метаданими як теоретичного обґрунтування для подальшої програмної реалізації пошукової системи.

Враховуючи накопичений науковою спільнотою досвід у даному напрямку, в роботі пропонується використовувати семантичну модель на основі онтології. Пошук навчальних об'єктів потребує наявності двох типів онтологій. По-перше, предметно-орієнтовані онтології, які описують деяку область знань навчального простору (наприклад, математика, фізика та ін.) або її частину. По-друге, загальна онтологія інформаційного пошуку, яка відображає загальні поняття та їх відношення, що є знанням про процес пошуку навчальних матеріалів та не залежить від навчальної дисципліни.

Узагальнена архітектура системи на основі пропонованого підходу наведена на рис. 4.6. На цій схемі позначено три базові функції пошукової системи, а саме: індексування, формування запитів та порівняння. Реалізація цих функцій визначається обраною моделлю пошуку. В даній роботі пропонується продовжити дослідження семантичної моделі пошуку з використанням метаопису навчальних об'єктів на базі метаданих.

Програмна система буде призначена для відображення графічного матеріалу у вигляді слайдів та навчання користувачів. Вона розрахована на широке коло користувачів, що мають за мету розглянути матеріал з алгебри предикатів та використовувати його в подальшій роботі. Для якісного виконання поставленої задачі потрібно, щоб система могла відображати коректно більше слайдів, різні шрифти, графічні ефекти, переміщуватися по вкладкам без помилок и затримок.

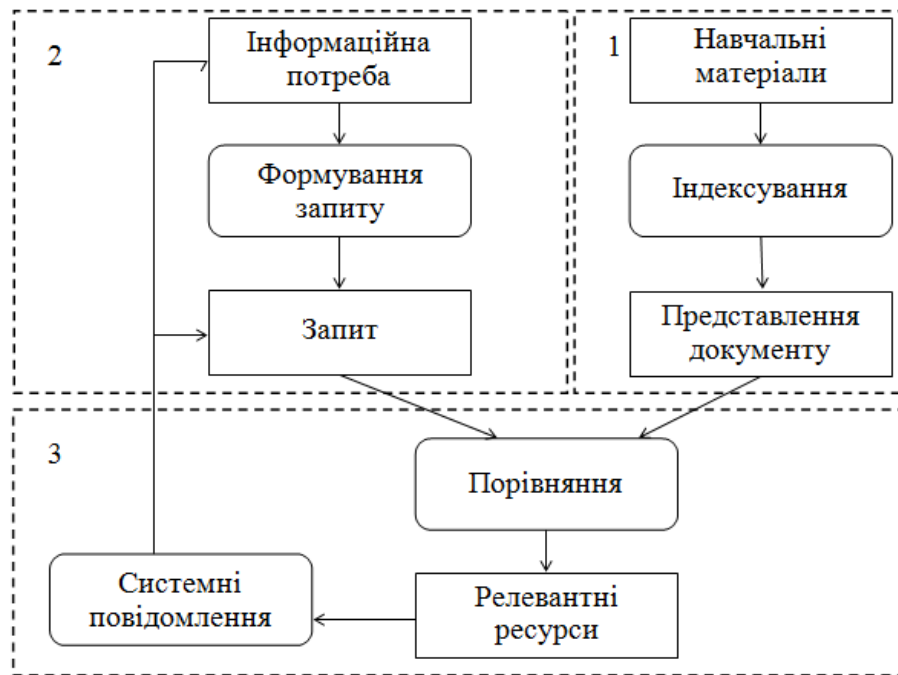


Рисунок 4.5 – Узагальнена архітектура системи пошуку

Коротко згадаємо про деякі з можливостей роботи з мовою. У мові для роботи з подіями та зворотними викликами представлена нова концепція об'єктно-орієнтовано і безпечно з точки зору системи типів посилання на функцію - "делегат". Саме ця концепція дозволяє описувати події C# компонента без зворотних інтерфейсів і додаткових зусиль, досить використовувати ключове слово event. Індикатори і властивості органічно доповнюють події до створення описуваного сигнатурою компонента.

C# має C стиль синтаксису (для керуючих конструкцій, блоків коду, опису сигнатури методів тощо), багато спільного з Java (відсутність множинного успадкування та шаблонів, наявність збирача сміття) і Дельфі (орієнтованість на створення компонент), в той же час має і свій колорит. При створенні мови в основу дизайну лягла легкість використання, домінуюча над потужністю мови та швидкістю виконання. Звідси і збирач сміття з керованими об'єктними посиланнями, який автоматично звільняє пам'ять, відбираючи при цьому процесорний час. Ви також отримуєте безпеку роботи з типами, а це на думку багатьох є другим найважливішим фактором уникнення помилок.

C# об'єктно-орієнтована мова, як і вся платформа .NET. C# створений для програмування в керованому середовищі з присутнім сміттярем, але дозволяє писати і некерований (unmanaged) код. При створенні мови розглядалася не тільки простота написання додатків, але і їх підтримки - у зв'язку з чим в мову включили підтримку XML коментарів і контролю версій. Справжній подарунок для програмістів.

Крім збирача сміття і безпеки роботи з типами для уникнення помилок у мові також застосовується автоматична ініціалізація змінних. Всі типи, які розміщуються в стеку (value types, до яких відносяться примітивні типи, структури та інші спадкоємці класу ValueType), ініціалізуються нулями. Об'єкти інших типів розміщуються в купі, доступ до них здійснюється по об'єктному посиланням, і посилання ці ініціалізуються в значення null.

У C# представлена концепція просторів імен, аналогічна пакетів в Java. Це дозволяє ієрархічно структурувати систему типів, роблячи код набагато більш зрозумілим і дозволяючи уникнути проблем з ім'ям. Ви можете розглядати простір імен як директорії, а мовні типи як файли в цих директоріях.

C# дозволяє використовувати при розробці парадигму атрибутивного програмування, коли в компіляційних модулях разом з типом може зберігатися мета-інформація. Дана інформація згодом може бути отримана і використана під час виконання. Більше того, цією інформацією може скористатися компілятор або, скажімо, дизайнер, що дозволяє, не запускаючи компонент, отримувати інформацію про нього. Робота з атрибутами дозволяє використовувати COM, COM + і DLL в C # додатках, що працюють в середовищі .NET. При написанні стандартних додатків під .NET важко підібрати більш відповідну мову, якщо Ви звикли до синтаксису C-подібних мов [11,12].

Також слід додати декілька слів про засоби візуального оформлення даної роботи, технологію WPF.

Windows Presentation Foundation (WPF) - це система наступного покоління для побудови клієнтських додатків Windows з візуально привабливими

можливостями взаємодії з користувачем. За допомогою WPF можна створювати широкий спектр як автономних, так і додатків, розміщених у веб-браузері.

В основі WPF лежить векторна система візуалізації, яка не залежить від розрішення і створена з розрахунком на можливості сучасного графічного обладнання. WPF розширює базову систему повним набором функцій розробки додатків, в тому числі Extensible Application Markup Language (XAML), елементами управління, прив'язкою даних, макетом, 2-D-і 3-D-графікою, анімацією, стилями, шаблонами, документами, мультимедіа, текстом і оформленням. WPF входить до складу Microsoft .NET Framework і дозволяє створювати додатки, що включають інші елементи бібліотеки класів .NET Framework.

У WPF додатково вдосконалюється процес програмування для розробки клієнтських додатків Windows. Одним очевидним удосконаленням є можливість розробляти програми за допомогою розмітки та коду програмної частини, з якими розробники ASP.NET повинні бути вже знайомі. Розмітка Extensible Application Markup Language (XAML) зазвичай використовується для реалізації зовнішнього вигляду програми при реалізації його поведінки за допомогою керованих мов програмування (коду програмної частини). Це розділення зовнішнього вигляду і поведінки має такі переваги:

- витрати на розробку та обслуговування знижуються, так як розмітка певного зовнішнього вигляду тісно не пов'язана з кодом певної поведінки;
- розробка більш ефективна, так як розробники зовнішнього вигляду програми можуть працювати одночасно з розробниками поведінки програми;
- для реалізації та спільного використання розмітки XAML застосовується безліч засобів конструювання, щоб задовольнити вимогам учасників розробки додатків. Microsoft Expression Blend призначається для конструкторів, в той час як Visual Studio орієнтується на розробників.
- глобалізація і локалізація для додатків WPF істотно спрощені [13].

5 РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ

Програма складається з двох модулів:

- перший модуль Business – логіка програми. В цьому модулі дані обробляються та створюються конвертовані презентації;
- другий модуль Presentation – інтерфейс програми. Відповідає за відображення матеріалу в програмі.

В перший модуль входить головна сторінка застосування та файл `index.jsp`. Саме цей файл дає змогу отримати і передати дані в модуль функцій, адже через особливості архітектури ми не можемо передавати данні напряму з `html`-сторінки.

Перелік файлів модуля Business:

Файл `PresentationSettings.cs` – налаштування програми, містить назву, тему і розділи.

Файл `Chapter.cs` – файл, в якому описується структура класу вкладок з інформацією, містить поле теми презентації, ім'я файлу презентації і шлях до даного файлу.

Файл `XpsToImageConverter.cs` – в даному файлі реалізований програмний конвертер презентацій з формату `.pdf` у формат `.xps`, читає файл презентації і перетворює його в набір картинок для відтворення в візуальному середовищі програми.

Перелік файлів модуля Presentation:

Файл `ChoicePanel.xaml` – в даному файлі знаходиться код з візуальною розміткою, анімаціями і стилями панелі, що дозволяє вибрати потрібну вкладку з навчальним матеріалом. Панель дозволяє вибрати потрібну главу (динамічно створює кнопки)

Файл `ChoicePanel.xaml.cs` – програмна реалізація вище зазначеного елемента інтерфейсу.

Файл `ImagePlayer.xaml` - вміє виводити картинки, які ми отримали за допомогою конвертера

Файл ImagePlayer.xaml.cs - програмна реалізація вище зазначеного елемента інтерфейсу/функціоналу.

Файл TitleControl.xaml - елемент управління який показує верхню частину програми.

Файл App.xaml - точка входу в програму.

Файл MainWindow.xaml - головне вікно програми, містить всю загрузку, ініціалізацію і пов'язує всі компоненти програмної системи воєдино.

Файл settings.xml – глобальні налаштування для коректного відображення програми.

На сьогодні реалізовано досить багато алгоритмів конвертації вхідних файлів в ресурси для відображення програмною середою, але дослідження в даній області не зупиняються не на секунду, так як жодне з програмних рішень не реалізує максимальної універсальності у роботі з різними типами файлів. У своїй роботі я використовую алгоритм перетворення pdf презентацій в програмні ресурси (див. рис. 5.1)



Рисунок 5.1 – Процес отримання програмних ресурсів з pdf презентації

Код конвертера для отримання кінцевих програмних ресурсів:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Media.Imaging;
using System.Windows.Xps.Packaging;
using System.Windows;
using System.Windows.Media;

namespace Business.Logic
  
```

```

{
    public class XpsToImageConverter
    {
        private BitmapSource[] _images;

        public XpsToImageConverter(string fileName)
        {
            Convert(fileName);
        }

        public BitmapSource[] Images
        {
            get
            {
                return _images;
            }
        }

        private void Convert(string fileName)
        {
            var document = new XpsDocument(fileName,
System.IO.FileAccess.Read);
            var seq = document.GetFixedDocumentSequence();

            _images = new BitmapSource[seq.DocumentPaginator.PageCount];

            for (int i = 0; i < seq.DocumentPaginator.PageCount; i++)
            {
                var visual = seq.DocumentPaginator.GetPage(i).Visual;
                FrameworkElement fe = (FrameworkElement)visual;

                RenderTargetBitmap bmp = new
RenderTargetBitmap((int)fe.ActualWidth,
                    (int)fe.ActualHeight, 96d, 96d,
PixelFormats.Default);

                bmp.Render(fe);

                _images[i] = bmp;
            }
        }
    }
}

```

В програмній реалізації можна побачити, що кінцевим ресурсом, що відображається у відповідному полі програми, являється масив растрових зображень `BitmapSource`, в якому зберігаються отримані з xps файлу зображення. Далі масив зображень передається візуалізатору `ImagePlayer`, фрагмент його коду приведений нижче:

```

public BitmapSource[] Images //отримання масиву картинок
{
    get
    {

```

```

        return _images;
    }
    set
    {
        if (value != null && value.Length > 0)
        {
            _images = value;
            MoveFirst();
        }
    }
}
//далі йдуть відгуки на натискання контролюючих кнопок
protected void MoveFirst()
{
    _currentImageIndex = 0;
    var img = new Image();
    img.Source = Images[_currentImageIndex];
    img.Name = "current";
    img.MouseUp += new MouseButtonEventHandler(img_MouseUp);
    this.Content = img;
    _currentImage = img;
}

void img_MouseUp(object sender, MouseButtonEventArgs e)
{
    if (e.ChangedButton == MouseButton.Left)
    {
        MoveNext();
    }
    if (e.ChangedButton == MouseButton.Right)
    {
        MovePrevious();
    }
}

public void MoveNext()
{
    _currentImageIndex++;
    if (_currentImageIndex >= _images.Length)
    {
        _currentImageIndex--;
        return;
    }

    _currentImage.Source = _images[_currentImageIndex];
}

public void MovePrevious()
{
    _currentImageIndex--;
    if (_currentImageIndex < 0 )
    {
        _currentImageIndex++;
        return;
    }
}

```

```
        _currentImage.Source = Images[_currentImageIndex];  
    }  
}
```

Функції MoveNext() і MovePrevious() реалізують переключення сторінок слайдів лекції за допомогою кнопок миші (ліва – вперед, права – назад).

Метод img_MouseUp(), отримуючи в якості одного з параметрів, вирішує, яка кнопка миші була натиснута в даний момент, і виходячи з результату перевірки, викликає одну з вищезазначених функцій.

Спочатку запускаємо програму файлом Presentation.exe. Перший етап роботи програми – запуск та вибір необхідного навчального матеріалу. Після цього можна переглядати текст презентацій. Програма може бути згорнута та відкрита заново в повноекранному режимі за допомогою відповідних елементів керування в правому верхньому кутку форми, що дозволяє зручно використовувати її поряд з іншими десктопними додатками.

База даних та матеріалів інтегрована в програмний код у вигляді хрс-бібліотек, конвертованих стандартним засобом передачі інформації у відмінний від оригіналу тип файлів.

ВИСНОВКИ

Задачею даної роботи було проведення аналізу предметної області та детально дослідження матеріалу з алгебри скінченних предикатів та основних методів мінімізації булевих функцій, вивчення відомих типів інформаційно – навчальних систем та обрати потрібну програмну реалізацію, а також інтегрувати даний матеріал в автоматизовану інформаційно – навчальну систему. Метою роботи була розробка інформаційно – навчальної системи для вивчення матеріалу з алгебри скінченних предикатів та основних методів мінімізації булевих функцій.

Після дослідження предметної галузі та порівняння інформаційно-навчальної системи з існуючими аналогами був виділений ряд переваг реалізованого додатка перед схожими програмними рішеннями:

- досить висока швидкодія на середніх апаратних конфігураціях;
- максимальна простота користування;
- інтуїтивність та зрозумілість інтерфейсу;
- безвтратна робота з великими за об'ємом даними;
- можливість додавання нових навчальних тем;
- мобільність додатку.

Застосування спеціального конвертера презентаційних файлів в програмні ресурси значно спростило відображення кінцевої інформації з вихідного файлу без втрат повноти та якості матеріалу, що дозволило зберегти швидкодію додатку та уникнути можливих проблем з відображенням інформації.

Таким чином, в рамках даної роботи отримані наступні результати:

- отримала подальший розвиток постановка задачі розробки мультиагентної системи оперативної обробки даних за рахунок формалізації проблеми використання ресурсів електронного навчання;

- запропоновано використання мережі потреб та властивостей для вирішення завдань розподілу ресурсів в системі електронного навчання в

режимі реального часу, що на відміну від існуючих підходів дозволяє будувати вирішення будь-якої складної задачі як динамічної мережі зв'язків, які гнучко змінюються в реальному часі;

– розроблено формальну модель взаємодії агентів в процесі встановлення балансу між потребами та можливостями системи електронного навчання.

Найбільші перспективи в цій галузі бачаться у постійному підвищенні рівня інтерактивності автоматизованих навчальних систем та інтеграції їх з веб-додатками, а також в налагодженні комунікацій між користувачами системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бондаренко, М.Ф., Котляров С.О. Эвристический алгоритм решения уравнений алгебры конечных предикатов / М.Ф. Бондаренко, С.О. Котляров; Харьков, 1984. - 9 с. - Деп. в УкрНИИИТИ N 337 - УК 84.
2. Янголенко О. В., Лютенко І. В., Яковлева О. В. Аналіз стану інформаційних технологій в системі вищої освіти // Вісник Національного технічного університету «Харківський політехнічний інститут». Харків: НТУ «ХПІ». 2012. № 30. С. 105- 109.
3. University Management System – UMS. URL : <http://www.techxact.com/university-management-system-ums.html>.
4. Henderson-Seller B., Giorgini P. Agent-Oriented Methodologies. London : Idea Group Publishing, 2022. 413 p.
5. Wooldridge M. J. An introduction to multiagent systems John Wiley & Sons, LTD, 2009. 461 p.
6. Russell S., Norvig P. Artificial Intelligence: A Modern Approach, 3rd Edition. Paris : Pearson Education France, 2019. 1152 p.
7. Alkhateeb F., Maghayreh E. Al., Abu Doush I. Multi-Agent Systems Modeling, Interactions, Simulations and Case Studies. InTech, 2011. 512 p.
8. Weiss G. Multiagent Systems. MIT Press, 2013. 867 p.
9. Тарасов В. Б. От многоагентных систем к интеллектуальным организациям. URL : <http://www.yugzone.ru/x/tarasov-v-b-ot-mnogoagentnykh-sistem-k-intellektual-nym-organizacsiyam>.
10. Субботін С. О., Олійник А. О., Олійник О. О. Неітеративні еволюційні та мультиагентні методи синтезу нечітко логічних і нейромережних моделей / Під заг. ред. Субботіна С. О. Запоріжжя : ЗНТУ, 2019. 375 с.
11. І.Ю. Шубін, А.Д. Козирєв, О.В. Галак. Методи створення інтелектуальної системи екологічного моніторингу та аналізу побудови складних границь територій. // Науково-технічний журнал «Біоніка інтелекту»

ХНУРЕ, 2020

12. Биков В. Ю. Моделі організаційних систем відкритої освіти. Київ Атіка. 2018. 684 с.
13. Symeonidis A. L. Mitkas P. A. Agent Intelligence through Data Mining. Aristotle University of Thessaloniki, Greece: Springer, 2021. 201 p.
14. Li Z., Duan Z. Cooperative Control of Multi-Agent Systems: A Consensus Region Approach. CRC Press, 2014. 252 p.