

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: **Механізм прийняття рішень в інтелектуальній системі аналізу**
проблемних ситуацій для ІТ-служби підприємства

Виконав: студент **IV** курсу, групи **СНЗс-42**
спеціальності **122 Комп'ютерні науки**

(шифр і назва спеціальності)

(підпис) **Рибка І.В.**
(прізвище та ініціали)

Керівник _____
(підпис) **Приймак М.В.**
(прізвище та ініціали)

Нормоконтроль _____
(підпис) **Шимчук Г.В.**
(прізвище та ініціали)

Завідувач кафедри _____
(підпис) **Боднарчук І.О.**
(прізвище та ініціали)

Рецензент _____
(підпис) _____
(прізвище та ініціали)

Тернопіль - 2021

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних наук

(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.

(підпис)

(прізвище та ініціали)

«__» _____ 2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

за спеціальністю

122 Комп'ютерні науки

(шифр і назва спеціальності)

Студенту

Рибці Ігорю Васильовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Механізм прийняття рішень в інтелектуальній системі аналізу проблемних ситуацій для ІТ-служби підприємства

Керівник роботи

Приймак Микола Володимирович, д.т.н., проф. каф. КН

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «02» 03 2021 року № 4/7-170

2. Термін подання студентом завершеної роботи

14.06.2021р.

3. Вихідні дані до роботи

наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз досліджуваної області. 1.1 Система управління ІТ-інфраструктурою підприємства. 1.2. Постановка завдання на проектування. 1.3. Короткий аналітичний огляд за темою роботи. 1.4. Аналіз окремих програмних продуктів в сфері підтримки ІТ-

інфраструктури підприємства. 2. Проектна частина. 2.1. Архітектура TU. 2.2. Аналіз модулів TU. 2.3. Взаємодія компонентів системи TU. 2.4. Алгоритм взаємодії компонентів системи.

2.5. Механізм прийняття рішення. 2.6. Порівняльний аналіз механізмів міркувань.

3. Програмна реалізація механізму. 3.1. Реалізація компоненту Reasoner. 3.2. Основні компоненти модуля Reasoner. 3.3 Взаємодія компонентів. 3.4. Експериментальні дані.

4. Безпека життєдіяльності, основи хорони праці.

Висновки. Перелік використаних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титулка. 2. Актуальність. 3. Мета, задачі дослідження. 4. Вимоги до ІС.

5. Аналіз окремих програмних продуктів підтримки ІТ- інфраструктури підприємства

6. Архітектура Thinking Understanding (TU). 7. Рівні мислення.

8. Програмна реалізація механізму. 9. UML діаграми компонентів та класів.

10. Діаграми діяльності. 11. UML діаграма основних компонентів модуля Reasoner.

12. Діаграма моделі даних семантичних зв'язків контексту запиту.

13. Експериментальні дані. 14. Висновки по роботі.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	02.03 – 05.03	Виконано
2.	Підбір джерел про механізми прийняття рішень	06.03 – 25.03	Виконано
3.	Опрацювання джерел про прийняття рішень в сфері підтримки ІТ- інфраструктури підприємства	26.03 – 12.04	Виконано
4.	Виконання дослідження щодо прийняття рішень в для інтелектуальній системі аналізу проблемних ситуацій ІТ-служби підприємства	13.04 – 29.04 30.04 – 15.05	Виконано
5.	Розроблення програмного коду	15.05 – 20.05	Виконано
6.	Оформлення розділу «Аналіз досліджуваної області»	21.05 – 28.05	Виконано
7.	Оформлення розділу «Проектна частина»		Виконано
8.	Оформлення розділу «Програмна реалізація механізму»	29.05 – 06.06	Виконано
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи хорони праці»	12.05 – 22.05	Виконано
10.	Оформлення кваліфікаційної роботи	28.05 – 10.06	Виконано
11.	Нормоконтроль	11.06 – 14.06	Виконано
12.	Перевірка на плагіат	11.06 – 14.06	Виконано
13.	Попередній захист кваліфікаційної роботи	08.06 – 10.06	Виконано
14.	Захист кваліфікаційної роботи	14.06	

Студент

(підпис)

Рибка І.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Приймак М.В.

(прізвище та ініціали)

АНОТАЦІЯ

Механізм прийняття рішень в інтелектуальній системі аналізу проблемних ситуацій для ІТ-служби підприємства // Кваліфікаційна робота бакалавра // Рибка Ігор Васильович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, кафедра комп'ютерних наук, група СНзс–42 //Тернопіль, 2021 // С. – 55, рис. – 14, табл. – 9, слайдів – 14, бібліогр. – 28.

Ключові слова: АУТСОРСИНГ, ІНТЕЛЕКТУАЛЬНА СИСТЕМА, МІРКУВАННЯ, МОДЕЛЬ МИСЛЕННЯ, СЕМАНТИЧНІ ЗВ'ЯЗКИ, ПРИЙНЯТТЯ РІШЕНЬ

Кваліфікаційна робота присвячена реалізації механізму підсистеми прийняття рішень для підвищення ефективності ІТ-служби підприємства. Даний механізм дозволить інтелектуальній системі з високою ймовірністю забезпечувати досягнення вірних висновків для вирішення завдання на основі наявних знань в системі.

Досліджено поняття системи управління ІТ-інфраструктурою підприємства. Здійснено аналіз окремих програмних продуктів в сфері підтримки ІТ- інфраструктури. В основі системи прийняття рішень для реєстрації і аналізу проблемних ситуацій в ІТ-інфраструктурі підприємства лежить модель мислення Thinking Understanding. Описана архітектура моделі, проаналізовано її модулі та взаємодія її складових компонентів. Виконана програмна реалізація механізму прийняття рішень.

Розроблена бібліотека є універсальним рішенням і може бути використана в будь-яких інших системах, де необхідний механізм логічного виводу або механізм міркувань.

ANNOTATION

Mechanism of decision making in the intelligent system of problem cases analysis for enterprise IT service // Rybka Ihor // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2021 // P. - 55, Fig. - 14, Table - 9, Slides - 14 , References - 28.

Keywords: OUTSOURCING, INTELLECTUAL SYSTEM, REASONING, MODEL OF THINKING, SEMANTIC RELATIONSHIPS, DECISION MAKING

Thesis deals with the implementation of the mechanism of the decision-making subsystem to improve the efficiency of the IT service of the enterprise. This mechanism will allow an intelligent system with a high probability to ensure the achievement of correct conclusions to solve the problem based on existing knowledge in the system.

The concept of enterprise IT infrastructure management system is studied. The analysis of separate software products in the field of IT infrastructure support is carried out. The decision-making system for registration and analysis of problem situations in the IT infrastructure of the enterprise is based on the Thinking Understanding thinking model. The architecture of the model is described, its modules and interaction of its constituent components are analyzed. The software implementation of the decision-making mechanism has been completed.

The developed library is a universal solution and can be used in any other systems where a logical inference mechanism or reasoning mechanism is required.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

AGI (Asterisk Gateway Interface) – інструментарій для запуску програм, написаних практично на будь-якій мові програмування.

k-line – концепція, що дозволяє «запам'ятовувати» вдалі рішення, пов'язує отримані знання.

NARS (Non-Axiomatic Reasoning System) – інтелектуальна система міркувань.

PLN (Probabilistic Logic Network) – імовірнісна логічна мережа.

SOAP (Simple Object Access Protocol) – простий протокол доступу до об'єктів для обміну структурованими повідомленнями в розподіленому обчислювальному середовищі.

T³ – триплет Критик - Селектор - Спосіб мислення.

TU (Thinking Understanding) – платформа для побудови інтелектуальних інформаційних систем, написана на мови Scala з використанням технології Akka Concurrency.

UML (Unified Modeling Language) – мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення, моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

URI (Uniform Resource Identifier) – уніфікований (однаковий) ідентифікатор ресурсу. Це послідовність символів, яка ідентифікує абстрактний або фізичний ресурс.

БД – база даних.

ІС (Інтелектуальна Система) – технічна або програмна система, здатна вирішувати завдання, що традиційно вважаються творчими, належать до конкретної предметної області, знання про яку зберігаються в пам'яті такої системи.

ІТ (Інформаційні Технології) – сукупність методів і засобів, котрі використовуються для збору, зберігання, обробки і поширення інформації.

СУ – система управління.

ІІІ (Штучний Інтелект) – наука і технологія створення інтелектуальних машин, особливо інтелектуальних комп'ютерних програм; властивість інтелектуальних систем виконувати творчі функції, які традиційно вважаються прерогативою людини.

ЗМІСТ

Вступ.....	8
1 Аналіз досліджуваної області	10
1.1 СУ IT-інфраструктурою підприємства	10
1.1.1 Інтелектуальні СУ IT-інфраструктурою.....	11
1.1.2 Особливості управління IT-інфраструктурою	12
1.2 Постановка завдання на проектування	13
1.3 Короткий аналітичний огляд за темою роботи.....	15
1.4 Аналіз окремих програмних продуктів в сфері підтримки IT-інфраструктури підприємства.....	16
2 Проектна частина	18
2.1 Архітектура TU	18
2.1.1 Модель мислення TU 1.0.....	18
2.1.2 Критик - Селектор - Спосіб мислення	18
2.1.3 Рівні мислення.....	20
2.1.4 Концепція короткострокової і довгострокової пам'яті	22
2.1.5 Концепція k-line	22
2.2 Аналіз модулів системи TU	23
2.3 Взаємодія компонентів системи TU.....	24
2.4 Алгоритм взаємодії компонентів системи.....	25
2.5 Механізм прийняття рішення	26
2.6 Порівняльний аналіз механізмів міркувань	28
2.6.1 OpenCog	29
2.6.2 OpenNARS	30
3 Програмна реалізація механізму	33
3.1 Реалізація компоненту Reasoner	33
3.2 Основні компоненти модуля Reasoner.....	34
3.2.1 ReasonerLibrary.....	35
3.2.2 Reasoner.....	39

3.2.3 KnowledgeProvider	43
3.3 Взаємодія компонентів	43
3.4 Експериментальні дані	44
4 Безпека життєдіяльності, основи охорони праці	47
4.1 Навчання працюючих і інструктажі з охорони праці.....	47
7.2 Вимоги ергономіки до організації робочого місця оператора ПК.....	49
Висновки	52
Перелік використаних джерел	53

ВСТУП

В умовах економічної нестабільності багато підприємств та організацій змушені оптимізувати свої витрати. Одним з таких інструментів зниження витрат є передача функціоналу окремих структурних підрозділів і відділів для обслуговування сторонній організації (аутсорсинг). Найбільшою популярністю серед видів аутсорсингу користується ІТ-послуги, що пов'язано з активізацією використання ІТ і високим вимогам до фахівців у даній сфері. ІТ-аутсорсинг забезпечує досягнення безперебійної працездатності корпоративних інформаційних систем і мінімізацію ризиків збоїв в їх роботі.

Згідно з наведеними досліджень Gartner [1] аутсорсинг в сфері ІТ дозволяє знизити витрати організації, в середньому на 30 - 40%. Однак з поширенням самого ІТ-аутсорсингу, зростає конкуренція і між компаніями, котрі здійснюють ІТ-аутсорсинг, і вже останні змушені підлаштовуватися під кон'юнктуру ринку, пропонуючи своїм клієнтам більш низькі ціни на обслуговування та впроваджуючи всередині організації систему зниження витрат.

В рамках кваліфікаційної роботи розглядається реалізація однієї з основних вимог до ІС - механізму підсистеми прийняття рішень для ІС підвищення ефективності служби підприємства. Даний механізм дозволить ІС з високою ймовірністю забезпечувати досягнення вірних висновків для вирішення завдання на основі наявних знань в системі, що є актуальним в даний час для компаній, що розвиваються.

Метою кваліфікаційної роботи є розробка механізму прийняття рішень в ІС підвищення ефективності діяльності ІТ-служби підприємства.

1 АНАЛІЗ ДОСЛІДЖУВАНОЇ ОБЛАСТІ

1.1 СУ ІТ-інфраструктурою підприємства

Під управлінням ІТ-інфраструктурою і бізнес-процесами прийнято розуміти ефективний механізм керування та організації надання ІТ-послуг, який від початку спрямований на вирішення бізнес-потреб сучасного підприємства.

СУ ІТ-інфраструктурою, розроблені на базі рекомендацій ITIL v2 / v3, здатні вирішити найскладніші завдання управління інформаційною структурою компанії та співробітниками, що займаються наданням і технічною підтримкою послуг. При цьому ті фахівці, котрі здійснюють управління інформаційною системою підприємства (штатні працівники власної ІТ служби або зовнішня аутсорсингова компанія) повинні володіти необхідним рівнем кваліфікації і знань.

Основними функціями сучасних СУ ІТ-інфраструктурою підприємства є:

- автоматичне встановлення ПЗ (дають змогу проводити автоматичне виконання однотипних повторюваних операцій, до того ж вони містять засоби автоматизованого розповсюдження ПЗ і оновлень, часто з можливістю скасування змін);
- керування ліцензіями (містяться інструменти, необхідні для проведення інвентаризації ПЗ і його перевірки на відповідність заданій ліцензійній політиці, отримання статистичних даних та їх аналізу для виявлення програм, котрі рідко застосовуються, а також знаходження застарілих версій програмних продуктів);
- обробка подій різного характеру (автоматизована обробка мережевих подій, а також автоматичного виявлення і усунення причин їх виникнення, виправлення наслідків неполадок, виконання діагностики і відповідних попереджувальних заходів);

– керування доступністю і продуктивністю (мережевих пристроїв, серверів і іншого устаткування, що надають можливість вмикати і вимикати їх за розкладом або при появі в цьому необхідності).

1.1.1 Інтелектуальні СУ ІТ-інфраструктурою

Інтелектуальні інтегровані засоби, що призначаються для моніторингу та керування інфраструктурою ІТ на підприємстві, дозволяють виконувати вимоги бізнесу до роботи інформаційних сервісів, ефективно оптимізуючи витрати на їх надання. Виникнення неполадок і збоїв або низька продуктивність ІТ-інфраструктури негативно впливають на продуктивність здійснення бізнес-операцій і рівень якості надання сервісів.

З цих причин важливо не тільки мати вичерпну інформацію про функціонування ІТ-інфраструктури в режимі реального часу, а й засоби, що використовуються для аналізу її стану. Не менш цінні й механізми управління ІТ, а також життєвим циклом сервісів і можливістю зміни конфігурацій.

Для комплексного вирішення перерахованих вище завдань використовуються СУ, що автоматизують такі процеси управління:

- неполадками;
- продуктивністю;
- ІТ сервісами;
- активами сервісів і конфігураціями;
- структуроване збереження і обробка даних про ІТ-інфраструктури.

1.1.2 Особливості управління ІТ-інфраструктурою

Розвинена інфраструктура ІТ сучасного підприємства містить велику кількість апаратних і програмних складових, кількість яких постійно зростає. Одночасно взаємозв'язки між різними компонентами структури неухильно ускладнюються на фоні зростаючої інтеграції ІТ в бізнес-процеси. У підсумку все це визначає необхідність переходу завдання управління інфраструктурою ІТ з підтримуючого (технічного) рівня на рівень стратегічного характеру.

Отже, системи моніторингу та управління надають ІТ фахівцям можливість управляти інформаційною системою з єдиного центру. Рішення про впровадження таких систем є практично для кожної компанії на сьогоднішній день стратегічно правильним, оскільки дозволяє значно зменшити навантаження на системних адміністраторів, оперативно виявляти проблеми і швидко визначати причини їх виникнення та усувати, формувати статистику по відмовах, щоб на її підставі приймати рішення про модернізацію або впровадженні нових складових ІТ-інфраструктури.

Сучасні центри обробки даних (ЦОД) складаються з сотень одиниць серверного, мережевого устаткування, а також потужних систем зберігання даних, територіально розташованих в декількох ЦОД. Щоб ефективно управляти такою великою кількістю техніки і програм, потрібно багато тимчасових і трудових ресурсів. Багато підприємствам просто не під силу впоратися з такою складною і масштабним завданням. Тому найкращим виходом стане звернення в спеціалізовану аутсорсингової компанії.

Крім надання технічного обслуговування ІТ сервісів, фахівці компанії-аутсорсера зможуть запропонувати рішення, що дозволяє максимально ефективно автоматизувати управління інфраструктурою, а саме: БД і додатками, серверами, мережевими пристроями і системами зберігання інформації. Застосування сучасних СУ ІТ-інфраструктурою дає можливість виконувати автоматизацію повного циклу керування, інтегруючи їх з системами моніторингу і служби підтримки за допомогою загальної консолі управління.

Інші рішення управління ІТ інфраструктурою підприємства:

- загальний моніторинг;
- управління віртуальними середовищами;
- управління контентом;
- провіжінінг і активація послуг;
- управління інфраструктурою ЦОД;
- управління інженерною інфраструктурою;
- моніторинг програм.

1.2 Постановка завдання на проектування

Одним з інструментів оптимізації витрат організацій можна розглядати автоматизацію процесу технічної підтримки, яка передбачає впровадження апробованої ІС підвищення ефективності ІТ-служби. Під ІС мається на увазі програмна система, котра володіє здатністю до вирішення задач, котрі належать конкретній предметній області, а власне знання неї є збереженими в пам'яті системи.

Як правило, структура ІС має три основних складових [2]:

- базу знань;
- механізм виведення рішень;
- інтелектуальний інтерфейс.

ІС вивчаються групою наук, що об'єднуються під назвою ІІІ.

Залежно від компетенцій фахівців технічної підтримки виділяють кілька ліній підтримки. Система підвищення ефективності ІТ-служби підприємства передбачає заміну типових дій, які часто зустрічаються, котрі виконуються співробітниками першої лінії технічної підтримки компанії, наприклад: встановити або перевстановити додаток; вирішити проблему з доступом до того чи іншого ресурсу. Впровадження ІС дозволить оптимізувати ресурси компанії, а також буде сприяти підвищенню ефективності роботи фахівців при вирішенні складних завдань.

Варто навести основні властивості ІС:

- здатна на основі сформованої або заданої мети визначати методи, шляхи і засоби досягнення кінцевого результату;
- повинна мати у своєму розпорядженні можливостями проводити міркування в умовах неповноти інформації з використанням правил як достовірного, так і правдоподібного виводів і, таким чином, породжувати нові знання;
- повинна володіти здібностями (механізмами, методами, алгоритмами, програмами і т.п.) до узагальнення, концептуалізації знань та

апроксимації, що представляє собою науковий підхід, котрий полягає в заміні одних об'єктів іншими, близькими до вихідних, але більш простим;

- в рамках своєї компетенції повинна вміти осягати, ставити і розв'язувати задачі;

- є інформаційно відкритою системою, що розширює обсяг і зміст моделі світу про навколишньому середовищі і сфері діяльності;

- ІС в процесі функціонування не тільки використовує вже відому інформацію, але і генерує нову (знання, дані), тобто виступає в якості виробника і джерела інтелектуальних інформаційних ресурсів.

Вже згадана система повинна відповідати наступним вимогам:

- процедура розгляду заяв на природній мові – первинна необхідна умова в системах аналізу проблем користувача з побудовою статистики «Задоволеність користувача програмним продуктом» [3]. Надання правильного вирішення проблеми багато в чому залежить від розуміння мови, якою спілкуються фахівці;

- здатність системи навчатися - функція, що припускає спрощення в експлуатації та розширення кругозору самої системи. Відповідно до результатів досліджень [4], здатність до навчання є одним з важливих аспектів для будь-якої ІС, що забезпечує гнучкість і універсальність;

- діалог з фахівцем – здатність підтримування комунікації, що є одним зі складових механізму навчання ІС [5];

- проведення логічних міркувань - можливість узагальнення вирішення конкретної проблеми та поширення висновків, котрі зроблені на основі результатів проведеного моніторингу, для розв'язування інших завдань при використанні логічних методів міркувань.

Одним з головних вимог до ІС пред'являється наявність механізму міркувань. Даний механізм має на увазі вміння узагальнення рішень конкретної проблеми і поширення висновків, зроблених за результатами проведеного моніторингу, для вирішення інших завдань застосовуючи аналогію, дедукцію та індукцію і інші методи логічного мислення.

Кінцевим завданням роботи є реалізація механізму міркувань і логічних правил необхідних підсистемі прийняття рішень. Для реалізації такого механізму потрібно:

- провести аналіз предметної області;
- розробити допоміжну бібліотеку, що дозволяє обробляти запити і повертати результат на підставі наявних знань;
- інтегрувати розроблену бібліотеку в ІС реєстрації та аналізу проблемних ситуацій, побудовану на фреймворку TU [8].

Таким чином, результатом виконання кваліфікаційної роботи буде універсальна бібліотека, призначена для пошуку найбільш ймовірного рішення на основі наявних знань. Також буде проведена інтеграція бібліотеки з модулем прийняття рішень в системі.

Областю дослідження роботи є система прийняття рішень. Дана система є інформаційною автоматизованою системою, метою якої є надання допомоги користувачеві при розв'язуванні поставлених задач.

При вирішенні поставленої мети використовувалися такі методи дослідження: інформаційний пошук, інтелектуальний аналіз даних, пошук знань в БД, міркування на основі прецедентів, імітаційне моделювання, експериментальний, ситуаційний аналіз, і когнітивне моделювання.

1.3 Короткий аналітичний огляд за темою роботи

В рамках роботи автором вивчені праці вчених в галузі ІІІ і їх рішень. Однією з провідних організацій в цьому напрямку досліджень є консорціум OpenCog [6] (США). Цими роботами керує Бен Герцель (голова Artificial General Intelligence Society і OpenCog Foundation) - один зі світових лідерів в області ІІІ.

Дослідження в області машинної логіки також ведуться в рамках проекту NARS [7] під керівництвом професора університету Темпла (США) Пея Вонга. Інтерес до сфери ІІІ і когнітивістики підтверджується щорічним збільшенням кількості публікацій, процитованих в БД Scopus.

Для того щоб система могла працювати з запитами користувача, вона повинна «розуміти» мову, на якій вони складені. Подібні проблеми досліджуються в області обробки природної мови в університеті Шеффілда (Великобританія). Інший напрямок - це семантичний пошук, дослідження в цій області також активно ведуться в цьому ж університеті, зокрема, розроблено підхід Mimir, який реалізує можливості пошуку за принципом «пошук і відкриття».

Для організації пошуку рішень відповідно до запитів користувачів в таких системах використовуються онтології, наприклад, широко застосовується підхід, запропонований в Каліфорнійському університеті (США), заснований на застосуванні дерев тегів в онтології. В університеті Ганновера (Німеччина) розробляють ІС для обробки запитів до служби порятунку з метою зменшення часу реакції на подію. В Інституті Чіная (Індія) ведуться дослідження ІС обробки запитів користувачів в сфері телекомунікацій. У Сінгапурі проведено аналіз ефективності ІТ-служби підтримки великої компанії і показана можливість автоматизації ряду процесів.

Дослідження в області ІС підвищення ефективності ІТ-служби підприємства ведуться також лідерами ІТ-галузі: компаніями HP [9] і IBM [10]. Зокрема відома багатоцільова ІС IBM Watson, розробкою і дослідженням якої займається група під керівництвом професора А. Гоель.

1.4 Аналіз окремих програмних продуктів в сфері підтримки ІТ-інфраструктури підприємства

На сьогоднішній день на ринку представлені засоби для автоматизації в сфері підтримки інформаційної інфраструктури підприємства. Серед них варто виділити наступні:

- HP Open View [9];
- ServiceNOW [11];
- IBM Watson [10].

При аналізі наведених вище програмних продуктів встановлено, що всі розглянуті системи не відповідають повного набору необхідних вимог, наведених у вступі. Таблиця 1.1 містить зведені дані за розглянутими системам - вказані наявність або відсутність у них тієї чи іншої функції [12]. Тут + означає наявність ознаки, – - її відсутність.

Таблиця 1.1 – Порівняльний аналіз існуючих програмних засобів

Ознака аналізу	HP Open View	Service NOW	IBM Watson
Моніторинг	+	+	+
Регістрація інцидентів	+	+	+
Управління системами	+	–	–
Створення ланцюга обробки (Workflow) інцидента	+	+	–
Розуміння і формалізація запитів на природній мові	–	–	+
Пошук рішення	–	–	+
Застосування рішень	–	–	–
Навчання	–	–	+
Уміння проводити логічні міркування: генералізацію, спеціалізацію, синонімічний пошук	–	–	–

Як видно із даних таблиці 1.1, жодне з розглянутих рішень не здатне проводити логічні міркування. Найбільш розвиненою на сьогоднішній день програмної системою є комплекс IBM Watson.

2 ПРОЕКТНА ЧАСТИНА

2.1 Архітектура TU

В основі ІС прийняття рішень для реєстрації і аналізу проблемних ситуацій в ІТ-інфраструктурі підприємства лежить модель мислення TU 1.0 [13], побудована із застосуванням теорії Марвіна Мінскі [14, 16].

2.1.1 Модель мислення TU 1.0

Американський вчений в галузі ШІ Марвін Мінскі в своїй роботі "The emotion machine" [16] представив концепцію системи мислення і пам'яті людини. Дана концепція заснована на моделі триплета Критик - Селектор - Спосіб мислення (далі T^3), k-line (зв'язок між набутими знаннями, наприклад, сонце - тепло) для зіставлення знань і рівнів мислення.

Схематичне зображення T^3 показано на рисунку 2.1.

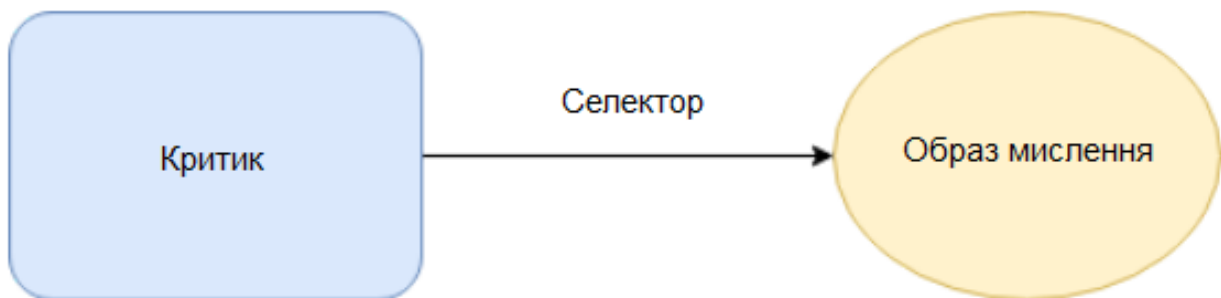


Рисунок 2.1– Критик - Селектор - Спосіб мислення

2.1.2 Критик - Селектор - Спосіб мислення

Критик в даному випадку є якимось перемикачем: зовнішні фактори, дія чи інший вплив. Наприклад, «надлишок вуглекислого газу в крові і почастішало дихання», «поворот голови на звук». Активація - тобто, включення дії критика відбувається при наявності певного набору факторів. Необхідно відзначити можливість існування ситуацій з одночасним включенням декількох критиків.

Наприклад, при вирішенні деяких завдань людиною, відбувається активація набору критиків, таких як виконання розрахунку, конкретизація технічних моментів. Також одночасно активізується критик, що нагадує про необхідність перерви на відпочинок, сон або прийом їжі.

Селектор визначає необхідні ресурси, в тому числі і образ мислення.

Спосіб мислення - це спосіб вирішення проблеми. Образ мислення не завжди є простим, у багатьох випадках він ініціює функціонування інших критиків. Так наприклад, співробітник компанії при вирішенні нового завдання усвідомлюючи значний обсяг роботи для отримання необхідного результату, починає займатися пошуком готового вирішення такого завдання, не «винаходячи велосипед». В даному випадку, «пошук готового рішення» - це критик всередині образу мислення «пошук рішення».

Розширена модель роботи T^3 зображена на рис. 2.2.

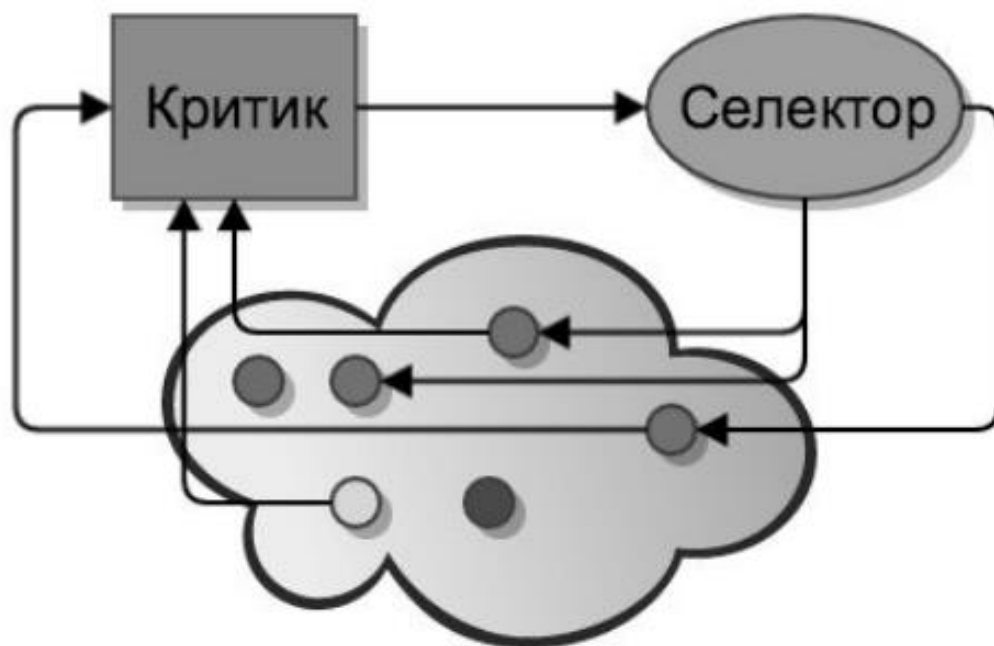


Рисунок 2.2 – T^3 в розрізі ресурсів

Образ мислення активується селектором, який в свою чергу активується критиком. Також образ мислення здатний ініціювати роботу інших критиків або

здійснювати конкретні маніпуляції. Наприклад, якщо піднявся рівень цукру в крові у діабетика необхідно ввести внутрішньом'язово інсулін містить препарат. В даному випадку селектори, критики, образ мислення і готові рішення - ресурси, що представляють собою набір знань з бази знань. При активації декількох критиків збільшується ступінь невизначеності і виникає необхідність конкретизації проблеми. За схожої вирішеної проблеми можливо розглядати і виконувати дії аналогічно.

2.1.3 Рівні мислення

Іншою важливою складовою моделі Мінські є концепція рівнів мислення, кожен з яких визначає більш комплексну поведінку і включає в себе складність попереднього.

Існують наступні рівні:

- інстинктивний;
- набутих знань;
- роздуми;
- рефлексивний;
- саморефлексивний;
- самосвідомий.

Необхідно підкреслити, що в моделі Мінські поділ рівнів є умовним. Дана концепція була доповнена і перенесена на комп'ютерну систему.

Перший з шести рівнів є низьким і являє собою вроджені інстинкти. На цьому етапі система ґрунтується на входженні ключових фраз в текстовий вираз, намагається вловити суть проблеми, використовуючи регулярні вирази, наприклад, стандартні запити сформульовані зовнішніми користувачами.

У випадку неможливості визначення системою рішення на першому рівні відбувається перехід на другий. На даному етапі передбачена побудова семантичної сітки вхідного запиту і активація критика, який класифікує проблеми. Загальну формулу для двох перших рівнів можна виразити конструкцією «якщо ..., то зробити так».

Третій рівень виконує контрольну функцію з пошуку рішення поточної проблеми і визначає для системи нові цілі. Головна мета - «Надання допомоги користувачеві». На даному рівні порівнюються наслідки декількох рішень і вибирається оптимальне. З цього моменту система запускає аналіз підцілей для розуміння запиту, проблеми і знаходження рішення. Загальну формулу для цього рівня можна подати у вигляді «якщо ..., то зробити так, тоді буде так».

За контроль часу виконання вхідного запиту відповідає четвертий рівень. Здійснюються міркування з урахуванням аналізу минулих подій. При перевищенні тимчасового інтервалу система перерозподіляє ресурси.

На п'ятому рівні визначаються контексти запитів і здійснюється комунікація з користувачем. Відбувається побудова певної моделі, за допомогою якої йде оцінка своїх вчинків.

Шостий рівень є вищим. Він відповідальний за стан системи, ресурсів, функціонування апаратного комплексу, та визначає загальний статус системи. Дається оцінка своїм вчинкам з точки зору вищих ідеалів і оцінок оточуючих. У разі виконання запитів за певний інтервал часу виставляється позитивний статус, при перевищенні часового ліміту - негативний. За даним статусу приймається рішення про потребу зовнішнього втручання в функціонування системи, наприклад, необхідність збільшення ресурсів.

В системі існує Поступовий контроль: наступні рівні контролюють попередні, так як мають доступ до параметрів управління попередніх.

Для програмної експертної системи дуже важливо мати здатність мислити і міркувати, наприклад, діяти за аналогією. Множина запитів, як правило, є типовою, запити відрізняються лише параметрами. Наприклад, таким є запит «Будь ласка, встановіть Office, Antivirus» і т. д.

Також для експертної системи важливо вміти абстрагувати спеціалізовані рецепти вирішення. Наприклад, система навчилася вирішувати інцидент "Будь ласка, встановіть Chrome", абстрагуватися даний інцидент до ступеня "Будь ласка, встановіть браузер", система зможе тим же способом усунути проблему "

Будь ласка, встановіть Opera", так як концепції " Chrome " і " Opera " пов'язані через концепцію "Браузер".

2.1.4 Концепція короткострокової і довгострокової пам'яті

Створена концепція короткострокової і довгострокової пам'яті, представляє собою розширення моделі мислення Мінські, передбачає можливість обміну інформацією між наведеними вище рівнями і запитами. Спосіб мислення взаємодіють з короткостроковою пам'яттю, видозмінюють і перетворюють дані. Далі в загальній базі знань зберігається успішно оброблений запит, переписаний в довгострокову пам'ять з короткострокової, тобто в базу знань записується лише протестована, вірна інформація.

У даній системі модель даних представлена семантичної мережею, реалізованою над нереляційною БД [17], і описує «знання» системи: проблеми, рішення, образи мислення, критики, селектори і т. д. Необхідно відзначити, що складові моделі триплета T^3 можуть бути «Набутими знаннями».

Варто також підкреслити, що система має можливість навчатися. Результат досягнутий також за рахунок розширення моделі мислення Марвіна Мінські. На першому етапі система має основні концепції, такі як об'єкт і дію.

Завдяки здатності до навчання системи можливе додавання нових концепцій. Навчання, також як і запити від користувача, здійснюється на природній мові. Наприклад, «лебідь - це об'єкт. Шипун- це лебідь». Тепер система знає дві нові концепції лебідь і шипун, а також те, що шипун - це лебідь.

2.1.5 Концепція k-line

Крім того, важливою складовою моделі Мінські є концепція k-line, зазначена вперше Марвіном Мінські в журналі Cognitive Science і детально описана в роботі "The Society of Mind" [18]. K-line є складовою двох подій, шляхом узагальнення їх в знання. Так наприклад, об'єднання образу мислення, виявленого рішення і активованої проблеми.

Концепція k-line здатна «запам'ятовувати» вірні рішення. На рис. 2.3 зображено схематичне уявлення k-line, що об'єднують образи мислення, рішення та інші критики.

Для програмної експертної системи одним з основних вимог є вміння мислити і аналізувати. Так здатність, проведення відповідності між раніше успішно вирішеними проблемами і новими, є необхідною вимогою до системи. Наприклад, «Встановіть Chrome», «Встановіть Firefox» і так далі. Модель мислення TU 1.0 має програмну реалізацію у вигляді платформи для побудови інтелектуальних інформаційних систем, названої TU.

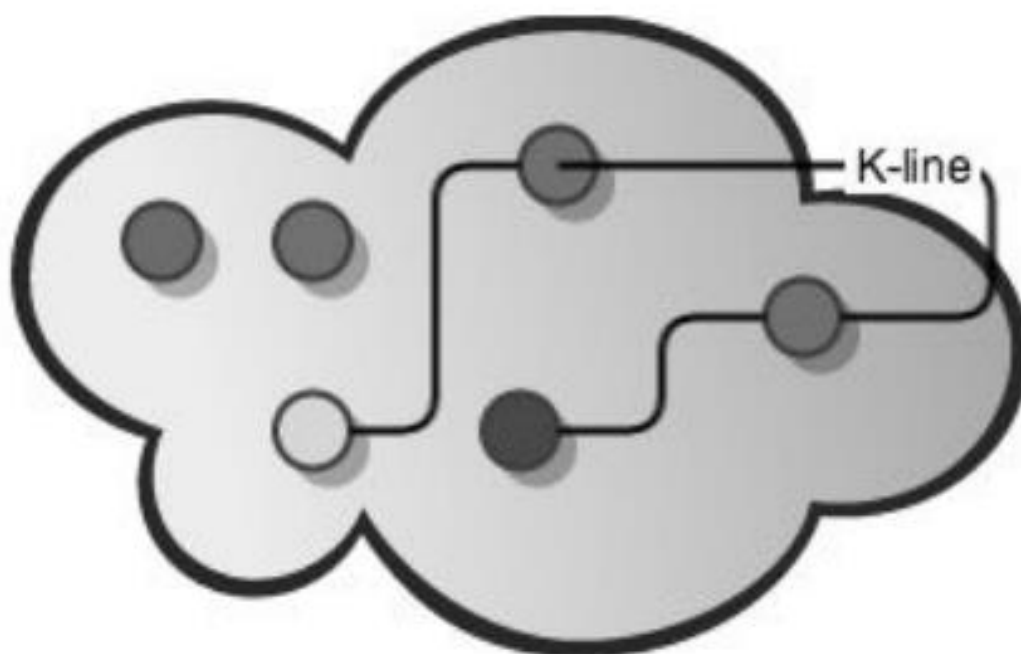


Рисунок 2.3 – Зображення концепції k-line

2.2 Аналіз модулів системи TU

Реалізація системи TU складається з декількох модулів. Коротко розглянемо опис компонентів системи, представлених в таблиці 2.1.

Таблиця 2.1 - Опис компонентів системи TU

Компонент	Опис
TU Webservice	Основний компонент взаємодії із зовнішніми системами, включаючи користувача
CoreService	Ядро системи містить основні класи
DataService	Компонент роботи з даними
Reasoner	Компонент ймовірнісної логіки
ClientAgent	Компонент виконання <u>скриптів</u> на цільовій машині
MessageBus	Шина даних для системи

2.3 Взаємодія компонентів системи TU

Система TU складається з декількох компонентів. Схематичне зображення даних компонентів і їх верхньорівнева взаємодія представлено на рис. 2.4.

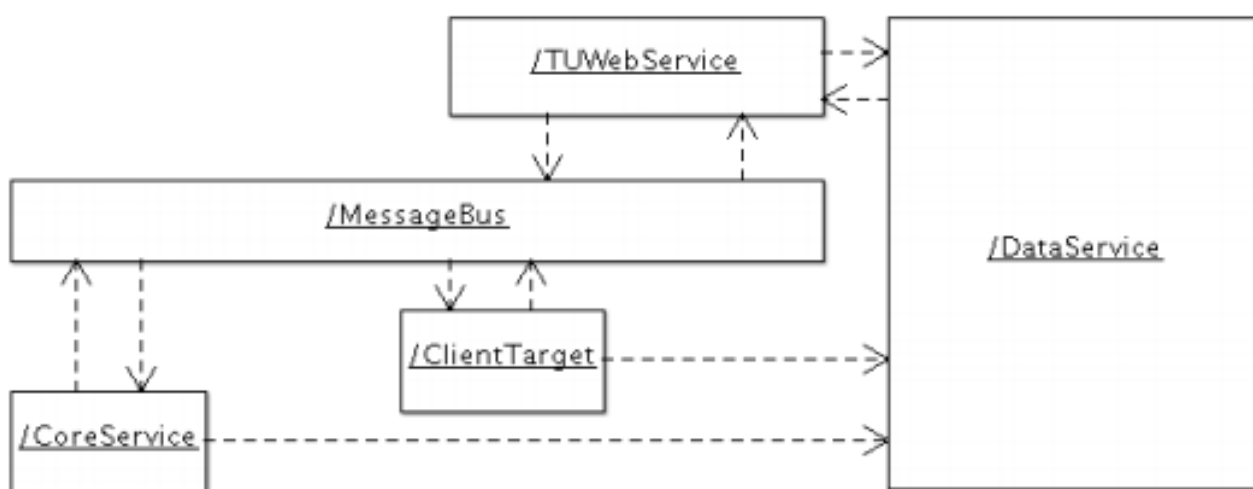


Рисунок 2.4 – Діаграма взаємодії компонентів

2.4 Алгоритм взаємодії компонентів системи

З позиції користувача точкою взаємодії з системою є компонент WebService. WebService реалізований із застосуванням стандарту WS SOAP для використання в різних системах [19].

Схема взаємодії складається з таких пунктів.

1. У WebService надходить запит від користувача і потім цей компонент записує запит до бази знань.

2. WebService відправляє запит у вигляді об'єкта Request в шину повідомлень (MessageBus).

3. Запит обробляється в CoreService.

4. CoreService записується результат в базу знань і відправляє в шину повідомлення у вигляді об'єкта RequestCompleted і повідомлення в вигляді об'єкта ActionsToExecute з переліком дій, необхідних для виконання.

5. При отриманні об'єкта RequestCompleted с результатами виконання запиту, компонент WebService повідомляє передплатників (кінцевих користувачів).

6. Повідомлення у вигляді об'єкта ActionsToExecute, що містить список дій, необхідних для виконання на клієнтських машинах, надходить в компонент ClientAgent.

Компонент MessageBus є шиною даних для передачі повідомлень обробникам. TUWebService - компонент, призначений для взаємодії з іншими інформаційними системами та надає набір функцій і типів, що дозволяють обробляти запити в системі. За інтеграцію з БД відповідає компонент DataService. Ядром додатка є компонент CoreService. CoreService управляє основним життєвим циклом системи. ClientTarget призначений для виконання команд на цільових машинах.

Кожен з перерахованих компонентів складається з більш дрібних частин.

Наприклад, CoreService складається з наступних частин:

ThinkingLifeCycle - компонент, побудований на фреймворку Akka [20]. Він відповідає за виконання критиків (Critic), селекторів (Selector), образів мислення (WayToThink), здійснює обмін даних між компонентами.

Таким чином, даний компонент управляє життєвим циклом системи: потоками і подіями додатки.

Дії, об'єкти і різні ресурси зберігаються в базі знань. Для їх отримання створений компонент Selector .

Для вирішення завдань оновлення і збереження оновлених даних відповідає компонент WayToThink. Крім операцій з даними він бере відповідальність за комунікацію з користувачем і всі дії, які змінюють операційний контекст даних системи.

Critic є основним компонентом для аналізу в T^3 . Critic використовується для класифікації вхідної інформації, рефлексії, самоаналізу і служить певним імовірнісним предикатом. Аналіз включає в себе прийняття рішень. Підсистема прийняття рішень виділена в окремий компонент, названий Reasoner.

2.5 Механізм прийняття рішення

Основою для механізму прийняття рішень повинна бути система, котра реалізує певні форми подання знань, механізми їх придбання та інтерпретації з використанням процедурного аналізу і метапроцедур, що лежать в основі інтелектуальної діяльності людини:

- дедукції;
- індукції;
- абдукції.

Останні можна розглядати як три взаємно доповнюючі один одного форми міркування (форми логічного виводу), тобто побудови послідовності аргументів, що приводять до деякого твердження - мети міркування.

Дедукція - це логічний вивід, що базується на точних знаннях загального плану і дозволяє за вихідними посиленнями (аксіомами) отримувати достовірні

виводи. При цьому мова може йти як про формальні міркування на рівні абстрактних законів, так і про інтерпретовані конкретні міркування [21].

Індукція являє собою механізм узагальнення, який реалізує побудову деякого загального правила на підставі кінцевої множини спостережуваних фактів (конкретних прикладів). Це форма правдоподібного виведення від часткового до загального, використовувана для формування емпіричних знань інтелектуальної системи [22].

Іншою формою правдоподібного виведення є абдукція – механізм формування гіпотези, що пояснює спостережувані факти на основі існуючих теоретичних положень (законів). Це вивід від часткового до часткового, найважче піддається алгоритмізації і використовує для верифікації (оцінки) висунутих гіпотез (пояснювальних гіпотез) методи індукції [23].

Міркування на базі аналогій можна розглядати як метод виведення, який дозволяє виявити подібність між кількома визначими об'єктами, та через перенесення фактів і знань, котрі є справедливими для одних об'єктів, на базі цього подібності на інші об'єкти, визначити спосіб вирішення завдань, або передбачити невідомі факти і знання. Саме такий природний метод умовиводів часто використовує людина, коли стикається з невідомим завданням.

Зазначені форми міркування можна проілюструвати на прикладах наступних трьох силогізмів.

Приклад №1.

Всі автомобілі, що зійшли з конвеєра 1 квітня, мають дефект в рульовому управлінні.

Ці автомобілі зійшли з конвеєра 1 квітня.

Ці автомобілі мають дефект в рульовому управлінні.

Приклад №2.

Ці автомобілі зійшли з конвеєра 1 квітня.

Ці автомобілі мають дефект в рульовому управлінні.

Всі автомобілі, що зійшли з конвеєра 1 квітня, мають дефект в рульовому управлінні.

Приклад №3.

Всі автомобілі, що зійшли з конвеєра 1 квітня, мають дефект в рульовому управлінні.

Ці автомобілі мають дефект в рульовому управлінні.

Ці автомобілі зійшли з конвеєра 1 квітня.

Перший силізм представляє дедуктивне міркування, другий -індуктивне узагальнення, третій - абдуктивне міркування.

Таким чином, зазначені механізми покликані забезпечити:

- аналітичний (дедуктивний) логічний вивід, спрямований на розв'язання задач;
- синтетичний (індуктивний і абдуктивний) вивід, що забезпечує розвиток бази знань і розширення можливостей дедуктивного методу за рахунок відмови від чітко фіксованої множини аргументів на ланцюжках міркувань, використання стратегій і правил правдоподібного виведення, застосування металогічних засобів управління виводу.

Таким чином, механізм прийняття рішень може бути реалізований у вигляді бібліотеки і не залежати від конкретної області застосування. Ця бібліотека буде працювати з певним набором знань і обробляти запити інформаційної системи або користувача. При використанні бібліотеки в TU буде використовуватися база знань TU.

2.6 Порівняльний аналіз механізмів міркувань

На поточний момент існує два механізми міркувань: PLN і NARS.

Між PLN і NARS існує історичний зв'язок. Розробка PLN почалася в рамках співпраці з творцем NARS Пий Вангом в спробі створити імовірнісний аналог NARS. PLN свій початок з NARS і має багато спільних ідей, але розвивалася в зовсім іншому напрямку. Між цими інструментами є істотне семантичне розходження: семантика PLN є ймовірнісною, тоді як семантика NARS - ні [24].

PLN і NARS мають аналогічний поділ на вивід першого порядку порівняно з виводом вищого порядку і мають компоненти першого порядку, які строго базуються на логіці терміна. Однак вивід PLN більш високого порядку вводить предикатні і комбінаторні логічні ідеї, тоді як вивід NARS більш високого порядку також заснований на логіці.

І PLN, і NARS реалізують способи міркувань таких як: індукція, дедукція і абдукція в своїх компонентах першого порядку з ідентичними графічними структурами. Однак в PLN індукція і абдукція отримані з допомогою обчислень за правилами Байєса, тоді як в NARS вони мають свої власні повністю незалежні функції істинності. Обидва інструменти використовують багатоконпонентні значення істинності, але семантика кожного компонента дещо відрізняється [7, 24]. Також, кожен з них має свою реалізацію: відповідно OpenCog [6] і OpenNARS [25].

2.6.1 OpenCog

Метою проекту є побудова рамок ІІІ на основі open-коду. Використовується OpenCog Prime, архітектура для пізнання роботів та втілення у віртуальному світі, котра визначає множину взаємодіючих складових, які здатні породжувати еквівалентний ІІІ особи як абсолютний феномен всієї системи.

OpenCog має деякі обмеження:

- масштабованість. Містить в собі сховище знань, яке погано масштабується;
- обмеження архітектури. Всі когнітивні процеси працюють в одному потоці;
- обмін знаннями. Складність передачі знань.

2.6.2 OpenNARS

Вирішальним фактором при виборі системи міркувань OpenNARS стала наявність документації і технічний опис, а також стек використовуваних технологій при розробці TU.

Open-NARS - це реалізація NARS з відкритим вихідним кодом, націлена на створення інтелектуальних інформаційних систем, тобто, «Мислячої машини», яка відповідає тим самим принципам, що й людська свідомість і здатна вирішувати завдання в різних областях [24].

Парадигма NARS заснована на переконанні, що сутністю інтелекту є принцип адаптації до навколишнього середовища при роботі з недостатніми знаннями і ресурсами. Відповідно, інтелектуальна система повинна спиратися на кінцевий процесинговий ресурс, працювати в режимі реального часу, бути відкритою для несподіваних завдань, і вчитися на власному досвіді.

Це робоче визначення інтерпретує "інтелект" як форму "відносної раціональності", оскільки обґрунтованість виведення оцінюється відповідно до наявних знань і ресурсів системи в даний момент, а не у відповідності з об'єктивною реальністю або деякими абсолютними критеріями. Навіть якщо рішення забезпечується системою для даного завдання, система як і раніше не володіє достатніми знаннями і ресурсами для цієї проблеми, звідси вона може змінити своє рішення, якщо стануть доступними нові докази або більший процесинговий час. Дана дефініція "інтелекту" не тільки узгоджується з його розумінням інтелекту в цілому, а також вона дає області III власне тотожність, звідси, власне, і визначено таким чином.

NARS вирішує завдання, що надходять від користувачів або інших обчислювальних систем.

Оскільки NARS використовує Narsese (Мова комунікації з NARS) немає обмежень на зміст завдання.

NARS вирішує наступні типи завдань:

- Judgment - приймає переконання щодо визначення вірним, а також приймаючи нові переконання переглядає старі;

- Question – шукає найкращу відповідь на питання відповідно до існуючих переконань;

- Goal - виконує деякі системні операції для реалізації мети.

Як система міркувань, архітектура NARS складається з бази знань, механізму виведення і механізму управління. Пам'ять містить набір концепцій, список операторів і буфер для нових завдань. Кожен концепт визначається термом. Концепт містить в собі завдання і переконання для цього терму, а також посилання на пов'язані завдання і терми.

Механізм логічного виводу формує результат відповідно до набором вбудованих правил. Кожне правило логічного виводу створює нове завдання, ґрунтуючись на заданій завдання і переконання, пов'язані з одним і тим же поняттям.

Механізм управління багаторазово виконує робочий цикл системи, що складається з наступних етапів:

- вибирає завдання в буфері, які можуть включати в себе створення нових понять і переконань, а також безпосередню обробку завдання;

- вибирає концепції з пам'яті, потім вибирає завдання і переконання з поняття;

- передає завдання і переконання в механізм логічного виводу для рішення переданої завдання;

- якщо завдання забезпечує найкращу відповідь на вхідний питання або вказує на реалізацію мети, то додає вирішені завдання в буфер задач і відправляє результат виконання звіт в вередовище;

- зберігає оброблене переконання, завдання і концепцію в пам'ять.

Всі вибори на 1 і 2 кроці є ймовірними. Всі елементи (завдання, переконання або поняття) в рамках вибору мають пріоритетні значення, а ймовірність вибору кожного з них пропорційна її пріоритету. При створенні нового елемента, його значення пріоритету визначається відповідно до його батьківських елементів, а також типом механізму, який його виробляє. На кроці

5 значення пріоритету для всіх задіяних елементів коригуються відповідно до зворотного зв'язку поточного циклу.

OpenNARS - реалізація системи AGI [25], яка захоплює сприйняття, міркування, прогнозування, планування і прийняття рішень за допомогою єдиного принципу. Зокрема, обробка тимчасового виведення, являє собою новий підхід і демонструє багато аспектів, необхідні для того, щоб система могла вчитися і діяти в середовищі реального часу.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕХАНІЗМУ

3.1 Реалізація компоненту Reasoner

Для логічних обчислень і прийняття рішень в системі використовується компонент Reasoner. Наприклад, логічні обчислення використовуються в компоненті Critic. На рис. 3.1 представлений інтерфейс компонента.

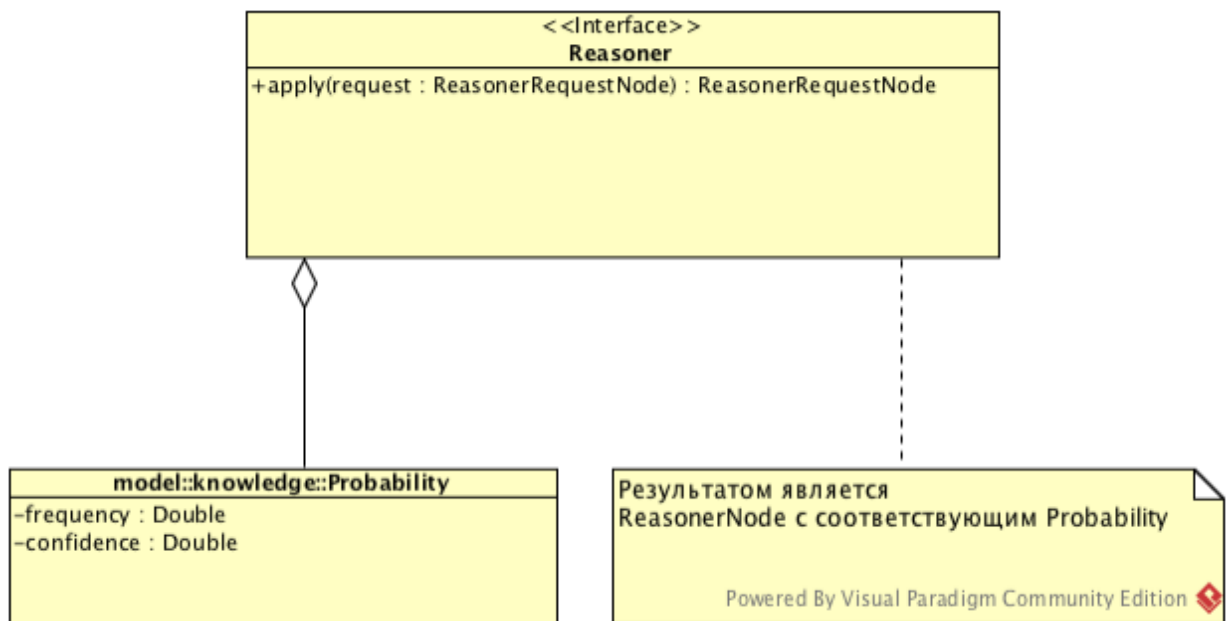


Рисунок 3.1 – Інтерфейс компонента Reasoner

У таблиці 3.1 наведено опис методів компонента.

Таблиця 3.1 - Опис методів компонента Reasoner

Метод	Опис
apply (request: ReasonerRequestNode): ReasonerRequestNode	Даний метод проводить обробку правил і вважає ймовірність (Probability) і впевненість (Confidence)

Основним застосуванням даного компонента є правила для Critic. Логіка правил обробляється за допомогою згаданого вище компонента. Цей результат використовується для визначення ймовірності активації даного Critic. Подібне використання дає гнучкість в побудові набору правил.

Для реалізації механізму прийняття рішень необхідний механізм міркувань. Він призначений для проведення логічних міркувань з використанням різних способів міркувань:

- аналогія - метод пізнання шляхом порівняння;
- дедукція - спосіб мислення, наслідком якого є логічний вивід, в якому частковий вивід виводиться із загального;
- індукція - процес логічного виводу на основі переходу від часткового положення до загального;
- абдукція - спосіб міркування, орієнтований на пошук правдоподібних пояснювальних гіпотез.

Основним завданням даного механізму є вміння узагальнювати рішення однієї проблеми і, екстраполюючи його, застосувати для вирішення інших. Іншими словами, це можливість для системи прийняти правильне рішення.

Для реалізації цього компонента необхідно:

- проаналізувати існуючі механізми міркувань;
- розробити модель даних.

3.2 Основні компоненти модуля Reasoner

На рис. 3.2 відображена UML діаграма компонентів підсистеми прийняття рішень TU.

На цьому рисунку наведено два модулі, які є основними компонентами:

- Reasoner Module – елемент TU відповідальний за прийняття рішення;
- Reasoning Library – розроблена бібліотека для прийняття рішень, що містить OpenNARS.

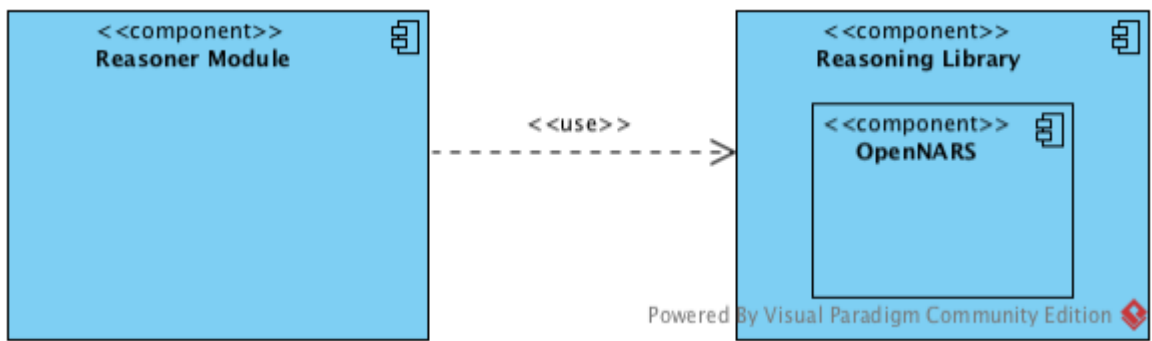


Рисунок 3.2 – UML діаграма компонентів

При отриманні запиту Reasoner Module трансліює запит в ReasonerLibrary і отримує результат виконання запиту.

3.2.1. ReasonerLibrary

Модель даних. На рис. 3.3 представлена UML діаграма класів розробленої бібліотеки. У таблиці 3.3 наведено опис класів моделі даних.

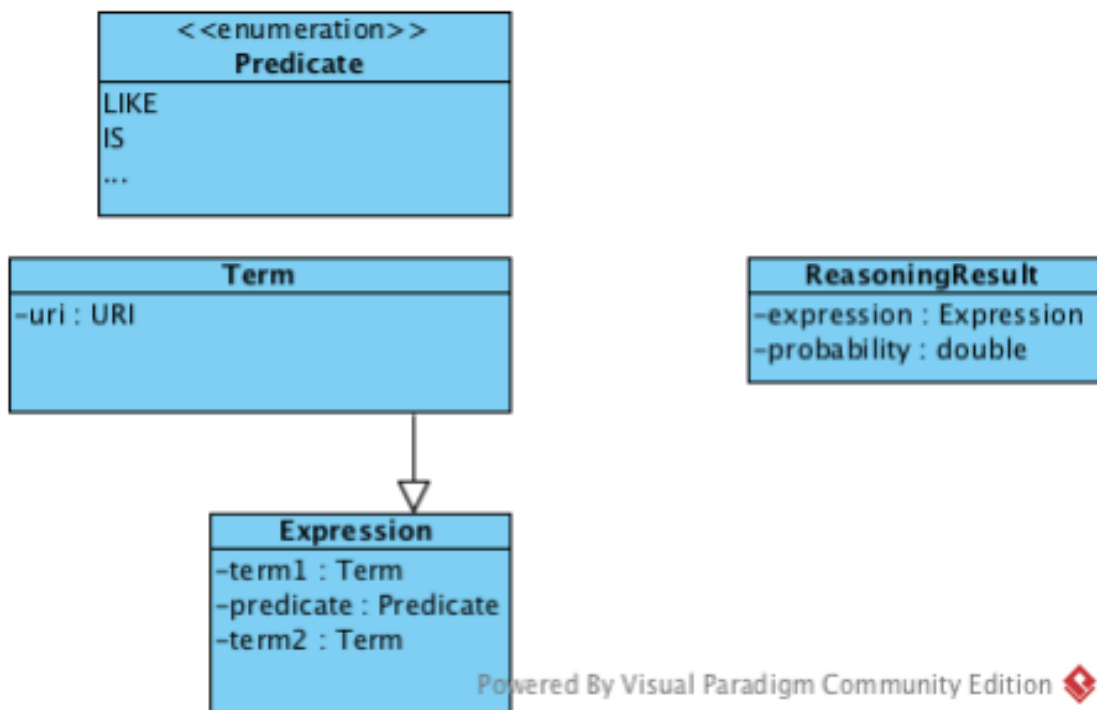


Рисунок 3.3 – Модель даних бібліотеки

Таблиця 3.2 - Опис класів моделі даних

Клас	Опис
Term	Атомарний об'єкт. Для ідентифікації цього об'єкта в базі знань використовується URI
Predicate	Тип взаємин між термами. наприклад, «Є», «схожий» та інші
Expression	Взаємовідносини між термами. Кожне взаємовідношення представлено двома термами і одним предикатом
ReasoningResult	Результат виконання запиту. Містить запит і ймовірність його правдоподібності

Взаємодія компонентів в ReasonerLibrary. Бібліотека ReasonerLibrary складається з декількох компонентів. Ці компоненти і їх взаємодія представлено на рисунку 3.4.

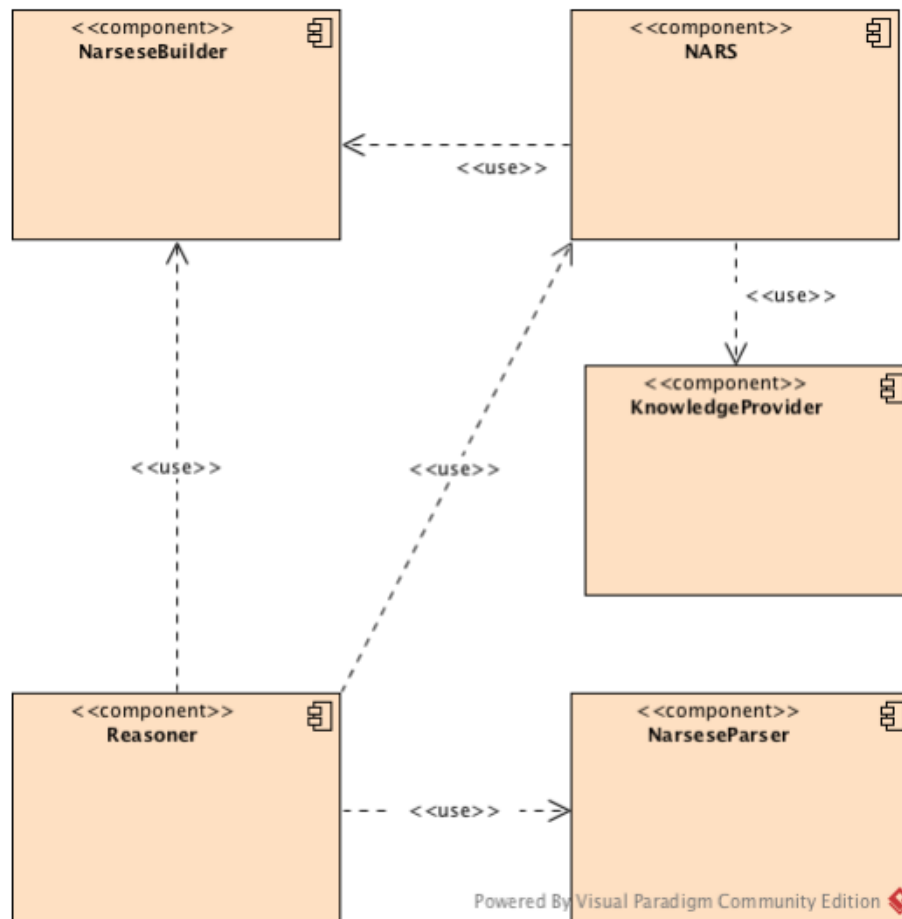


Рисунок 3.4 – Діаграма компонентів ReasonerLibrary

Reasoner - інтерфейс взаємодії з бібліотекою. Компонент є сполучною ланкою інших компонентів.

NarseseBuilder – елемент, який перетворює Expression в стрічковий вираз на мові Narsese.

Розглянемо його роботу на конкретних прикладах.

Пропозиція "Swan is a type of swimmer." є об'єктом Expression в наступному вигляді:

Expression e 1 =

{Predicate = 'IS',

term 1 = 'tu: resource.concept.object. swan',

term 2 = 'Tu: resource.concept.object. Swimmer'}

Компонент NarseseBuilder перетворює поточний вираз в наступний вид:

<Swan -> swimmer>.

Вираз "Swan is a type of bird." є об'єктом Expression в наступному вигляді:

Expression e1 =

{Predicate = 'IS', term 1 = 'tu: resource.concept.object. swam',

term 2 = 'Tu: resource.concept.object. Bird'};

Елемент NarseseBuilder перетворює цей вислів в наступний вид:

<Swan -> bird>.

NARS - елемент, що містить логіку роботи з OpenNARS.

Результатом роботи цього елемента даного компонента на попередніх прикладах буде вираз на мові Narsese:

<Bird <-> swimmer>. % 0.90;

Інтерпретація цього виразу на природну мову має вигляд:

" I guess that bird is similar to swimmer."

NarseseParser - елемент призначений для перетворення стрічкового виразу на мові Narsese в ReasoningResult.

Наприклад, розглянемо результат роботи компонента NARS:

<Bird <-> swimmer>. % 0.90;

Елемент NarseseBuilder перетворює цей вислів в наступний вид:

ReasoningResult result = {Expression e 3 = {predicate = 'LIKE',

term 1 = 'Tu: resource.concept.object. Bird',

term 2 = 'Tu: resource.concept.object. Swimmer'},

probability = 0,90};

KnowledgeProvider - компонент, який містить методи для роботи з базою знань. Реалізація інтерфейсів залежатиме від програми, в якій він використовується. В даному випадку буде використовуватися база знань TU.

Бібліотека ReasonerLibrary виконує певний порядок дій, представлений на рис. 3.5.

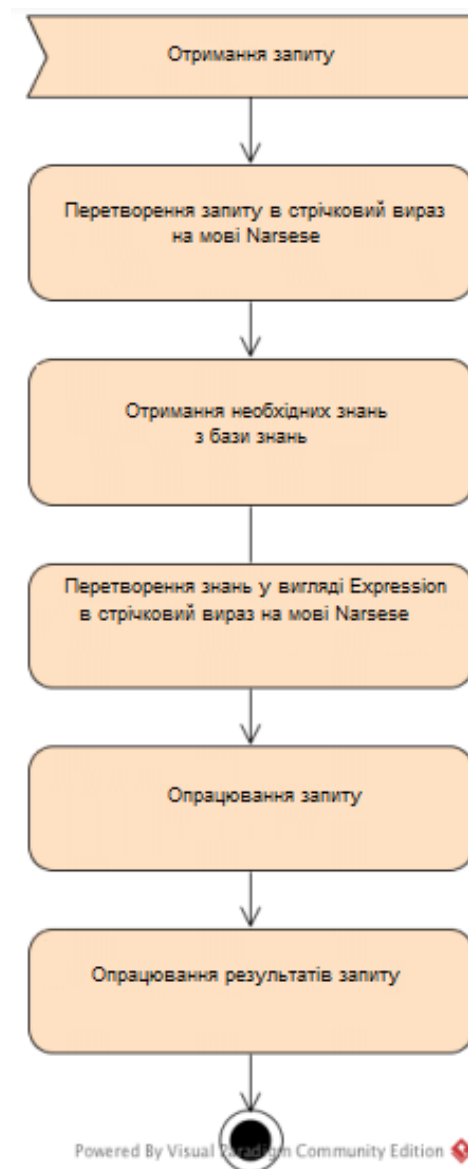


Рисунок 3.5 - Діаграма діяльності ReasonerLibrary

3.2.2 Reasoner

На рис. 3.6 наведена UML діаграма основних компонентів модуля Reasoner.

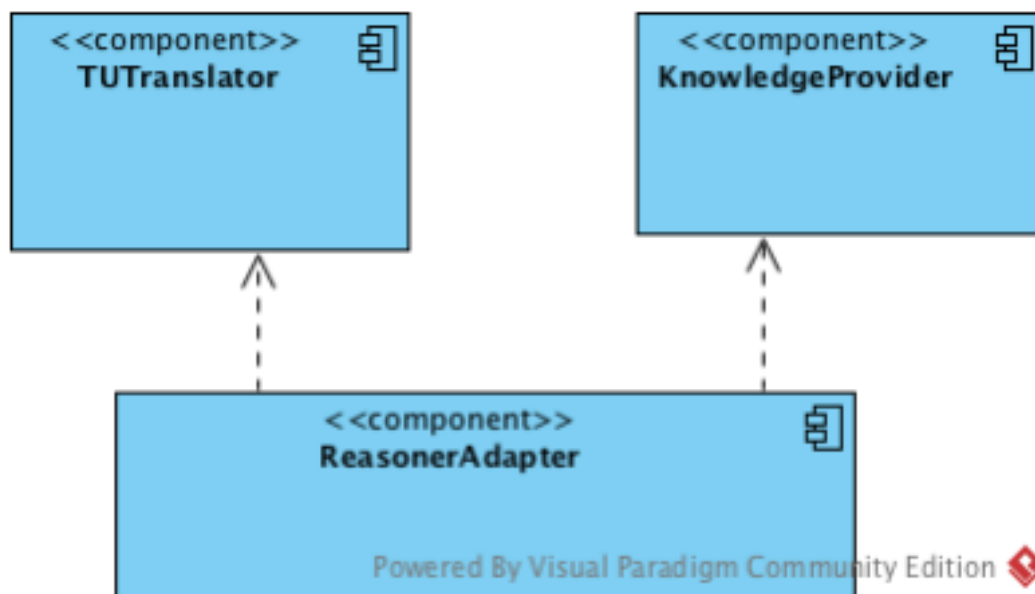


Рисунок 3.6 – Основні компоненти модуля Reasoner

Модуль Reasoner містить три компоненти:

- TUTranslator призначений для перетворення семантичних зв'язків контексту запиту в Expression;
- KnowledgeProvider надає наявні знання в бібліотеці;
- ReasonerAdapter використовує модуль KnowledgeProvider для надання бібліотеки всього накопиченого досвіду (знань) і TUTranslator для перетворення поточного запиту в зрозумілий для бібліотеки вид.

Компонент Reasoner виконує певний порядок дій, представлений на рис. 3.7.

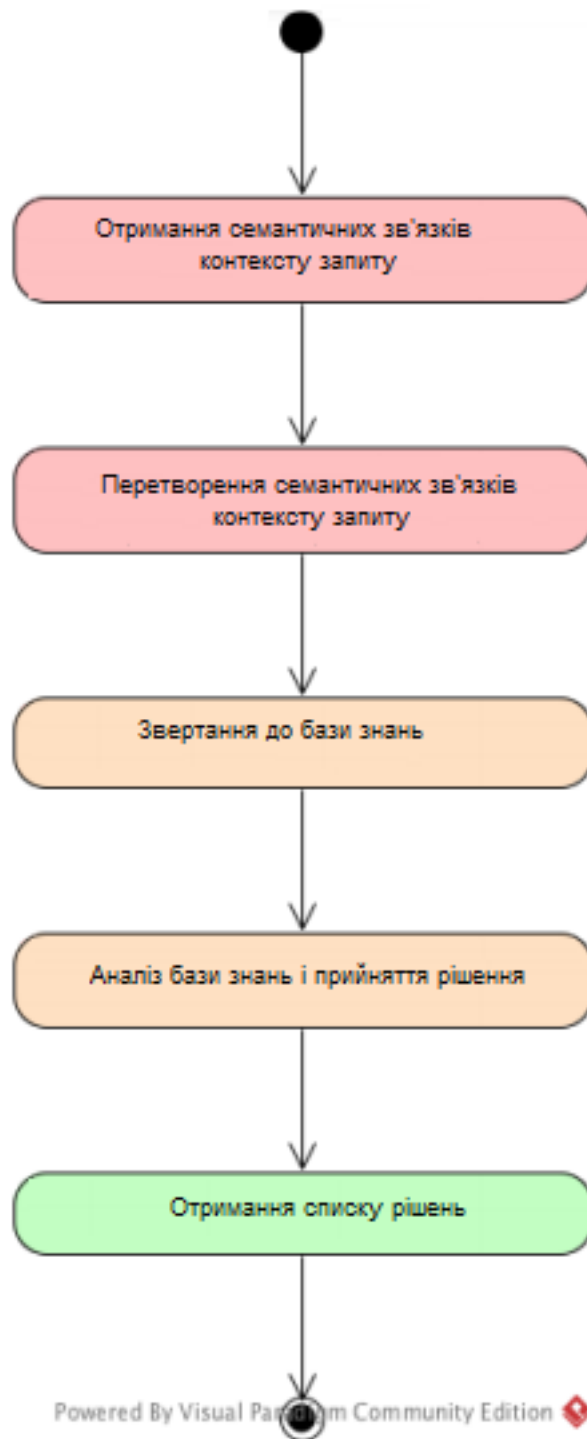


Рисунок 3.7 – Діаграма діяльності модуля Reasoner

За формування семантичних зв'язків запиту в ТУ відповідає образ мислення - LinkParser. Сформовані семантичні зв'язки представляють собою структуру зображену на рис. 3.8. У таблиці 3.3 наведено опис класів моделі даних, з якою працює образ мислення.

Таблиця 3.3 - Опис класів моделі даних

Клас	Опис
Resource	Базовий клас TU, в якому зберігається ідентифікатор об'єкта
Concept	Сутність. Наприклад, "Firefox"
ConceptLink	Зв'язок між сутностями. наприклад, в "Firefox is a browser" "is" є зв'язком між сутностями "Firefox" і "Browser"
AnnotatedPhrase	Пропозиція з розібраними на зв'язки між словами
AnnotatedSentence	Текст, що складається з пропозицій

При отриманні запиту модуль Reasoner працює з цим контекстом.

Для перетворення семантичних зв'язків в зрозумілий для бібліотеки вид відповідає модуль TUTranslator. Даний модуль перетворює модель даних представленої на рис. 3.8 в модель даних зазначеній на рис. 3.2.

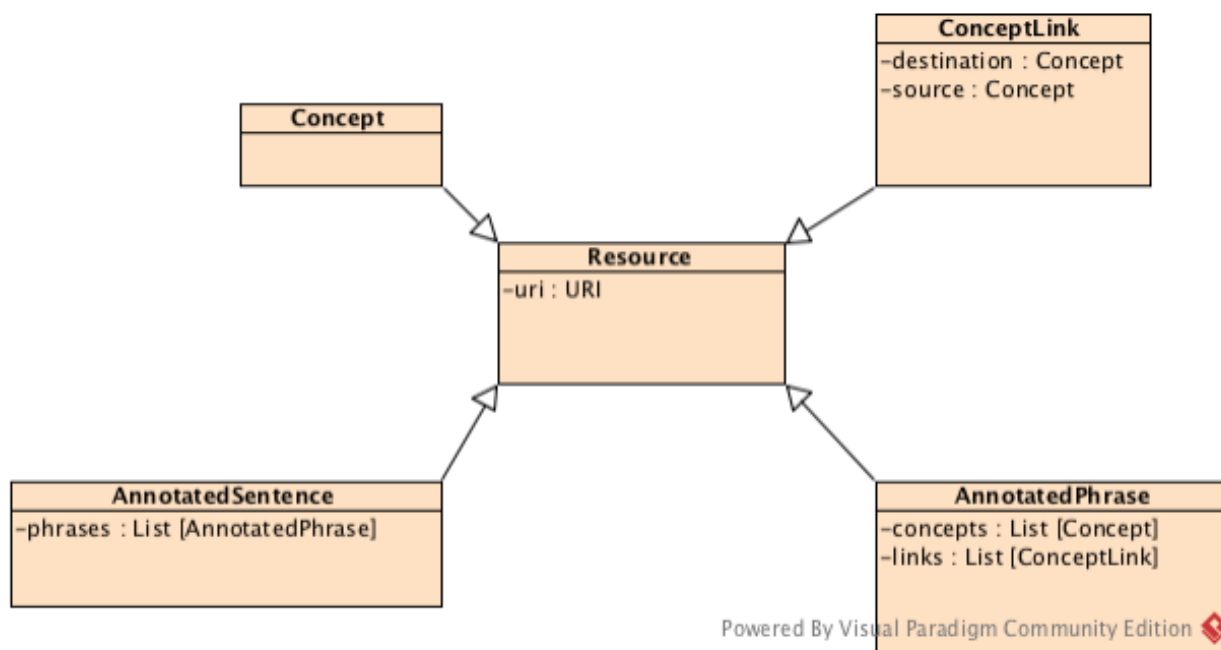


Рисунок 3.8 - Діаграма моделі даних семантичних зв'язків контексту запиту

3.2.3 KnowledgeProvider

Для роботи бібліотеки необхідна база знань. Оскільки база знань зберігається на стороні TU, необхідний провайдер бази знань TU в базу знань бібліотеки. Цей функціонал реалізований в моделі KnowledgeProvider, і представлений на рис. 3.9.



Рисунок 3.9 - Діаграма інтерфейсу KnowledgeProvider

У таблиці 3.4 наведено опис методів інтерфейсу KnowledgeProvider.

Таблиця 3.4 - Опис методів інтерфейсу KnowledgeProvider

Метод	Опис
getKnowledge ()	Повертає список, що складається з Expression

3.3 Взаємодія компонентів

ReasonerAdapter - являє собою інтерфейс для взаємодії з бібліотекою. Діаграма інтерфейсу ReasonerAdapter представлено на рис. 3.10.

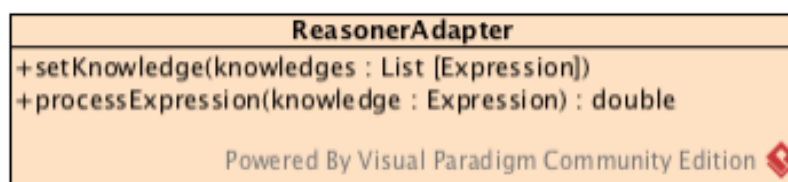


Рисунок 3.10 - Інтерфейс ReasonerAdapter

У таблиці 3.5 наведено опис методів інтерфейсу.

Таблиця 3.5 - Опис методів інтерфейсу ReasonerAdapter

Метод	Опис
setKnowledge (knowledges: List <Expression>	Передає знання з бази знань в бібліотеку
processExpression (knowledge: Expression): double	Повертає правдоподібність переданого виразу

У попередніх підрозділах розглянуто:

- механізми міркувань;
- модель даних бібліотеки;
- реалізація бібліотеки і модуля взаємодії TU і бібліотеки;
- основний потік дій системи.

Також представлені діаграми в форматі UML 2.0. На діаграмах представлені основні потоки роботи як для всієї системи, так і для кожного компонента. На основі цієї архітектури була виконана програмна реалізація з використанням функціональної мови програмування Scala.

3.4 Експериментальні дані

В якості експериментальних даних були складені різні запитання. Для початкового навчання в системі було закладено дві базові концепції:

- Object - об'єкт, який є базовою концепцією для всіх об'єктів;
- Action - дія, яка є базовою концепцією для всіх дій.

У таблиці 3.6 представлений список основних тренувальних даних.

На основі тестових даних були сформульовані тестові запитання, представлені в таблиці 3.7. Набір таких запитань достатньо якісно відображає процес навчання системи.

Таблиця 3.6 - Список тренувальних даних

Вхідний запит	Опис навчального запиту
Bird is an object	Створює зв'язок між концепцією bird і object
Swimmer is an object	Створює зв'язок між концепцією swimmer і object
Bird is a type of swimmer	Створює зв'язок між концепцією bird і swimmer
Bird is a type of animal	Створює зв'язок між концепцією bird і animal
Sport is an object	Створює зв'язок між концепцією sport і object
Competition is an object	Створює зв'язок між концепцією competition і object
Sport is a type of competition	Створює зв'язок між концепцією sport і competition
Swan is a type of swimmer	Створює зв'язок між концепцією swan і swimmer
Robin is a type of bird	Створює зв'язок між концепцією Robin і bird

Таблиця 3.7 - Список тестових запитів і результатів

Запит	Результат виконання запиту
Is bird a type of swimmer?	60%
Is bird a type of animal?	91%
Is sport a type of competition?	87%
Is swan a type of swimmer?	89%
Is Robin a type of bird?	75%

За підсумками проведеного експерименту видно, що система видає не сто відсотків в якості результату виконання запиту на навчених даних.

Така ситуація виникає через недостатність навчання системи, для усунення проблеми, котра виникла, необхідно проведення більш поглибленого навчання.

Таким чином розглянуто експериментальні дані, які були використані для верифікації бібліотеки, також дається обґрунтування отриманих результатів.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Навчання працюючих і інструктажі з охорони праці

Однією із складових ефективної роботи з профілактики виробничого травматизму є належна підготовка, навчання та підвищення кваліфікації працівників з питань охорони праці. Загальний порядок проведення навчання з питань охорони праці встановлений Законом України «Про охорону праці» (ст. 18. «Навчання з питань охорони праці»).

Виконання вимог Закону України «Про охорону праці» в частині проведення навчання та перевірки знань з питань охорони праці здійснюється відповідно до Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держкомітету України з нагляду за охороною праці 26 січня 2005 р. № 15 (далі — Типове положення).

Нагляд за дотриманням вимог Типового положення здійснюють органи державного нагляду за охороною праці, а координацію та методичний супровід — Головний навчально-методичний центр та навчальні підрозділи експертно-технічних центрів Держгірпромнагляду. Вивчення предмета «Охорона праці» при підготовці, перепідготовці та підвищенні кваліфікації працівників, які залучаються до виконання робіт з підвищеною небезпекою, на підприємстві регламентується п. 2.3 Типового положення. На підприємствах згідно з п. 1.1 Додатку 3 Типового положення навчання та перевірку знань з питань охорони праці повинні проходити керівники, заступники керівників, головні спеціалісти, керівники основних виробничих та технічних служб, безпосередньо пов'язані з організацією безпечного ведення робіт. Крім цього, згідно з п. 5 Додатку 3, навчання та перевірку знань з питань охорони праці мають проходити керівники, спеціалісти служб охорони праці, члени комісій з перевірки знань з питань охорони праці.

Типове положення встановлює порядок та місце проведення навчання та перевірки знань з питань охорони праці посадових осіб (п. 5.2 та п. 5.3). Посадові

особи, перелік яких наведено в п. 5.2, проходять навчання у Головному навчально-методичному центрі Держнаглядохоронпраці. Перевірка знань цієї категорії посадових осіб проводиться комісією, створеною наказом Держгірпромнагляду.

Організацію навчання та перевірки знань з питань охорони праці працівників на підприємстві здійснюють працівники служби кадрів або інші спеціалісти, яким роботодавець доручив організацію цієї роботи. Навчання та перевірка знань з питань охорони праці працівників (виконавців і посадових осіб), які не залучаються до виконання робіт підвищеної небезпеки, проводиться не рідше ніж один раз на три роки. Посадові особи та працівники, які виконують роботи підвищеної небезпеки, проходять спеціальне навчання та перевірку знань відповідних нормативно-правових актів з охорони праці не рідше одного разу на рік. Посадові особи малих підприємств (п. 5.4), які не мають можливості створити власні комісії з перевірки знань з питань охорони праці та провести навчання з питань охорони праці, проходять навчання та перевірку знань в навчальних закладах, які мають відповідний дозвіл. Спеціальне навчання з питань охорони праці може проводитись безпосередньо на підприємстві або навчальним закладом, який має відповідний дозвіл.

Періодичність інструктажів, навчання та перевірки знань з питань охорони праці залежить від видів виконуваних робіт та встановлюється Типовим положенням. Перевірка знань з питань охорони праці після проведення спеціального навчання проводиться комісією підприємства. Якщо на підприємстві неможливо створити комісію з перевірки знань з питань охорони праці (п. 4.4 Типового положення), перевірка знань проводиться комісією спорідненого підприємства або Теруправління Держгірпромнагляду.

Всі працівники та посадові особи підприємства, включаючи посадових осіб, відповідальних за виконання робіт підвищеної небезпеки (крім зазначених в п. 5.2 та п. 5.3 Типового положення), проходять навчання та перевірку знань з питань охорони праці на підприємстві. Типове положення не зобов'язує, але й не забороняє проводити навчання всіх виконавців та посадових осіб (особливо тих,

що виконують роботи підвищеної небезпеки) в навчальних закладах. У нашій країні є багато підприємств, де таке навчання проводиться, і це має позитивні наслідки. Ті витрати, які при цьому несуть підприємства, перекриваються створенням більш безпечних умов праці і в результаті збереженням життя та здоров'я працівників. Також в навчальних закладах проходять навчання та перевірку знань із загальних питань охорони праці всі посадові особи та фахівці, які проводять інструктажі або навчання підлеглих працівників з питань охорони праці, виконують роботи з проектування об'єктів, а також інші працівники, незалежно від того, передбачено таке навчання Типовим положенням чи ні.

4.2 Вимоги ергономіки до організації робочого місця оператора ПК

Робоче місце — це зона простору, що оснащена необхідним устаткуванням, де відбувається трудова діяльність одного працівника чи групи працівників [26]. Раціональне планування робочого місця має забезпечувати: найкраще розміщення знарядь і предметів праці, не допускати загального дискомфорту, зменшувати втомлюваність працівника, підвищувати його продуктивність праці. Площа робочого місця має бути такою, щоб працівник не робив зайвих рухів і не відчував незручності під час виконання роботи. Важливо мати також можливість змінити робочу позу. Проте доцільно виключати або мінімізувати всі фізіологічно неприродні і незручні положення тіла. Проведені дослідження показують, що при раціональній організації робочих місць продуктивність праці зростає на 15–25% [27].

Організація робочого місця користувача ПК має відповідати ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги», ДСан ПіН 3.3.2.007-98, характеру та особливостям трудової діяльності. Площа одного робочого місця користувача ПК повинна складати не менше 6 м², а об'єм — не менше 20 м³. Конструкція робочого місця користувача ПК повинна відповідати сучасним вимогам ергономіки, характеру виконуваної роботи і забезпечити оптимальне

розміщення на робочій поверхні документів та обладнання ПК (монітора, системного блоку, клавіатури, мишки та інших периферійних пристроїв. Монітор на робочому місці встановлюється так, щоб верхній край екрана знаходився на рівні очей.

Розташування монітора ПК має забезпечувати: безпечність роботи в цілому; зручність та ефективність зорової роботи з екраном в вертикальній площині під кутом 30° від лінії зору, площа екрана при цьому має бути перпендикулярною нормальній лінії зору користувача.

Клавіатура розміщується на поверхні столу або висувній полиці на відстані 100-300мм від краю, ближчого до користувача. Кут нахилу клавіатури має бути в межах 5-15°. Поверхня клавіатури повинна бути матовою з коефіцієнтом відбиття 0,4. Клавіші клавіатури мають бути зручними в роботі і м'якими при натисканні (хід всіх клавіш має бути однаковим з мінімальним опором натискання 0,25Н та максимальним – не більше 1,5Н) [28].

При розміщенні робочих місць з ПК слід дотримуватися вимог, зазначених в ДНАОП 0.00-1.31-99: робочі місця розміщуються на відстані не менше 1м від стін з світловими прорізами; відстань між бічними поверхнями моніторів ПК має бути не менше 1,2м; відстань між тильною поверхнею монітора одного ПК та екраном монітора іншого ПК має бути не меншою 2,5м. Вимоги двох останніх пунктів враховуються також при розміщенні робочих місць з ПК в суміжних приміщеннях з урахуванням конструктивних особливостей стін та перегородок.

Загальні принципи організації робочого місця:

- на робочому місці не повинно бути нічого зайвого. Усі необхідні для роботи предмети мають бути поряд із працівником, але не заважати йому;
- ті предмети, якими користуються частіше, розташовуються ближче, ніж ті предмети, якими користуються рідше;
- предмети, які беруть лівою рукою, повинні бути зліва, а ті предмети, які беруть правою рукою – справа;
- якщо використовують обидві руки, то місце розташування пристосувань вибирається з урахуванням зручності захоплення його двома руками;

- робоче місце не повинно бути зашарашене;
- організація робочого місця повинна забезпечувати необхідну оглядовість.

Статичні напруження працівника в процесі праці пов'язані з підтриманням у нерухомому стані предметів і знарядь праці, а також підтриманням робочої пози. Робоча поза – це основне положення працівника у просторі: зручна робоча поза має забезпечувати стійкість положення корпусу, ніг, рук, голови працівника під час роботи, мінімальні затрати енергії та максимальну результативність праці. Неправильна сидюча поза може викликати застій крові в ногах, а якщо виконується великий обсяг роботи для пальців рук – запалення суглобів.

Організація робочого місця користувача комп'ютера повинна забезпечувати відповідність усіх елементів робочого місця та їх взаємного розташування ергономічним вимогам (рисунок 4.1).

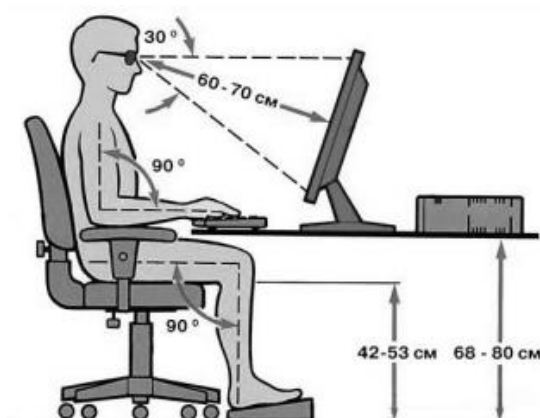


Рисунок 4.1 – Робоче місце і робоча поза користувача ПК

Найпоширенішими у процесі праці є пози сидючи і стоячи. Проектуючи робоче місце, потрібно враховувати, що при виконанні роботи з фізичним навантаженням бажана поза стоячи, а при малих зусиллях – сидючи. Робоча поза стоячи втомлює людину більше, ніж сидюча. Вона вимагає на 10% більше енергії, спричиняє підвищення артеріального і венозного тиску крові, розширення вен на ногах, пошкодження ступень, викривлення хребта [28].

ВИСНОВКИ

Однією з головних вимог, що пред'являються до ІС є здатність вести міркування. Така здатність має на увазі вміння узагальнення рішень конкретної проблеми і поширення висновків, зроблених за результатами проведеного аналізу, для вирішення інших завдань застосовуючи індукцію, дедукцію, аналогію і інші методи логічного мислення.

В кваліфікаційній роботі була розглянута ІС підвищення ефективності ІТ-служби підприємства, заснована на платформі TU з використанням однойменної моделі мислення.

В рамках роботи над проектом були вирішені поставлені завдання і отримані такі результати:

- проаналізовано платформи для побудови інтелектуальних інформаційних систем;
- проведено аналіз існуючих реалізацій механізмів міркувань;
- розроблена допоміжна бібліотека, що дозволяє обробляти запити і повертати результат на підставі наявних знань;
- проведена інтеграція розробленої бібліотеки в інтелектуальну систему реєстрації та аналізу проблемних ситуацій, побудовану на платформі TU.

Таким чином, розроблена бібліотека є універсальним рішенням і може бути використана в будь-яких інших інформаційних системах, де необхідний механізм логічного виводу або механізм міркувань.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Коптелов А. Аутсорсинг ИТ — ИТ-служба передается на аутсорсинг [Електронний ресурс]. — Режим доступу: <https://http://koptelov.info/publikatsii/outsorsing-it/> — (дата звертання: 01.05.2021).
2. Artificial intelligence [Електронний ресурс]. — Режим доступу: <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp> (дата звертання: 02.05.2021).
3. Tutubalina, E. Target-based topic model for problem phrase extraction (Conference Paper) [Text] / E. Tutubalina // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). — 2015. — Vol. 9022. — P. 271 – 277.
4. Bello-Orgaz, G.A. Social big data: Recent achievements and new challenges (Article) [Text] / G.A Bello-Orgaz, J.J. Jung, D.A. Camacho // Information Fusion. — 2015. — Vol. 28. — P. 45 – 59.
5. Baddoura, R. This Robot is Sociable: Close-up on the Gestures and Measured Motion of a Human Responding to a Proactive Robot (Article) [Text] / R Baddoura, G. Venture // International Journal of Social Robotics. — 2015. — Vol. 7. — P. 489 – 496.
6. Goetzl, B. OpenCog [Електронний ресурс]. — Режим доступу: <http://opencog.org/> (дата звертання: 02.05.2021).
7. Wang, P. Non-Axiomatic Logic A Model of Intelligent Reasoning. [Text] / P. Wang. — California, USA : World Scientific Publishing Company, 2013. — 276 p.
8. Toshev, A. Thinking model and machine understanding in automated user request processing [Text] / A. Toshev // CEUR Workshop Proceedings. — 2014. — Nov. — Vol. 1297. — P. 224–226.
9. HPE ONEVIEW. [Електронний ресурс]. - Режим доступу: <https://www.hpe.com/us/en/integrated-systems/software.html> - (дата звертання: 02.05.2021).

10. Let's put smart to work. [Электронный ресурс]. - Режим доступа: <https://www.ibm.com/thought-leadership/smart/> - (дата звертання: 02.05.2021).

11. Workflows for the modern enterprise. [Электронный ресурс]. - Режим доступа: <https://www.servicenow.com/> - (дата звертання: 02.05.2021).

12. Тощев А. С. Интеллектуальная система повышения эффективности ит-службы предприятия: диссертация ... кандидата технических наук: 05.13.11 / «Казанский (Приволжский) федеральный университет», 2017.- 121 с.

13. Toshchev, A. Thinking-Understanding approach in IT maintenance domain automation [Text] / A. Toshchev, M. Talanov, A. Krehov // Global Journal on 41 Technology: 3rd World Conference on Information Technology (WCIT-2012). — 2013. — Vol. 3. — P. 879 – 894.

14. Тощев, А.С. Архитектура и реализация интеллектуального агента для автоматической обработки входящих заявок с помощью искусственного интеллекта и семантических сетей [Текст] / А.С. Тощев, М.О. Таланов // Ученые записки Института социально-гуманитарных знаний. — 2014. — Т. No 1 (12), Ч. II. — С. 288 – 292.

15. Тощев, А.С. Модель мышления и понимания в автоматической обработке запросов пользователя [Текст] / А.С. Тощев // Труды 16-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции». — 2014. — С. 425 – 427.

16. Minsky, M. The Emotion Machine [Text] / M. Minsky. — New York, USA : Si- mon & Schuster, 2007. — 400 p.

17. NoSQL [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/NoSQL> – (дата звертання: 12.05.2021).

18. Minsky, M. The Society of Mind [Text] / M. Minsky. — NY, USA : Simon & Schuster, 2007. — 336 p.

19. An aggregated technique for optimization of SOAP performance in communication in Web services [Text] / K. Senagi, G. Okeyo, W. Cheruiyot, M. Kimwele // Service Oriented Computing and Applications. — 2015. — P. 6 – 7.

20. White, D. Akka Concurrency [Text] / D. White ; Ed. by K. Rolland. — [S. l.] : Artima, 2013. — 521 p.
21. Deductive reasoning [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Deductive_reasoning – (дата звертання: 11.05.2021).
22. Inductive reasoning [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Inductive_reasoning – (дата звертання: 13.05.2021).
23. Abductive reasoning [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Abductive_reasoning – (дата звертання: 13.05.2021).
24. Goertzel, B. Probabilistic Logic Networks [Text] / B. Goertzel, M. Iklé, A. Heljakka // Electronic Library. — 2008. — Vol. 24. — P. 832 – 846.
25. The OpenNARS implementation of the Non-Axiomatic Reasoning System [Text] / Patrick Hammer, Tony Lofthouse, Pei Wang [et al.] // 2015 IEEE 29th International Conference on Artificial General Intelligence. — Vol. 2016-April. — [S. l.] : IEEE, 2015. — mar. — P. 587–592.
26. Толок А.О. Крюковська О.А. Безпека життєдіяльності: Навч. посібник. — 2011. — 215 с.
27. Яремко З. М. Безпека життєдіяльності: Навч. посіб. — Львів., 2005. — 301 с.
28. Желібо Є. П. Заверуха Н.М., Зацарний В.В. Безпека життєдіяльності. Навчальний посібник. / Є. Желібо Є.П., Н.М. Заверуха П., В.В. Зацарний. — К.; Каравела, 2004. -328 с.