

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кваліфікаційна наукова праця
на правах рукопису

КАРПОВ ІГОР АНДРІЙОВИЧ

УДК 004.02

ДИСЕРТАЦІЯ
ПОБУДОВА СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ
ОНТОЛОГІЧНИХ МЕРЕЖ

124 – Системний аналіз

Технічні науки

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Карпов І. А.

Науковий керівник – Буров Євген Вікторович
д.т.н., професор

Львів – 2023

АНОТАЦІЯ

Карпов І. А. Побудова системи підтримки прийняття рішень на основі онтологічних мереж. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 124 «Системний аналіз» – Національний університет «Львівська політехніка», Міністерство освіти і науки України, Львів, 2023.

Зміст анотації. Дисертація присвячена розробці методів та побудові системи підтримки прийняття рішень на основі онтологічних мереж в умовах багатозначності.

Зростання обсягів інформації, яку необхідно враховувати при прийнятті рішень, визначає актуальність розробки інтелектуальних систем підтримки прийняття рішень. Передумовою коректного прийняття рішення є побудова концептуальної моделі проблемної ситуації, яка враховує всі релевантні фактори. Концептуалізація проблемної ситуації здійснюється за допомогою онтології цієї ситуації. Для формування онтології проблемної ситуації доцільно використовувати знання з існуючих онтологій, однак такий підхід може призвести до багатозначності вибору елементів онтологій, які найточніше відповідають ситуації.

Відмінність онтологій від експертних систем полягає в тому, що вони побудовані як узгоджений та широко розділений опис деякого узагальненого домену знань, зазвичай спрямованого на розв'язання конкретної проблеми. Таким чином, проблема створення та використання онтологій як опису знань у предметних областях є важливим завданням, що сприяє розвитку комп'ютерної інженерії, інформатики, штучного інтелекту та інших наукових дисциплін. Це відкриває нові можливості для застосування передових інтелектуальних інформаційних технологій у різних предметних галузях, включаючи ті, де раніше ці технології не застосовувалися.

Наукова новизна отриманих результатів полягає у:

1. Вперше розроблено метод багаторівневого та поетапного узгодження змісту онтології предметної області задачі прийняття рішення, що дозволяє зменшити кількість помилок концептуалізації. Цей підхід відрізняється від традиційних методів побудови онтологій, тим, що забезпечує більш точну та структуровану концептуалізацію предметної області, що сприяє покращенню якості прийнятих рішень.
2. Вперше розроблено метод, що базується на модифікованому методі TOPSIS для вибору онтології для підтримки прийняття рішення на основі аналізу подібності функцій та практичного використання онтологій інтелектуальним агентом, що дало змогу врахувати неявні складові концептуалізації та покращити обґрунтованість вибору онтології.
3. Вперше запропоновано модель подання контекстно залежної структури інтерпретацій концептів онтології на основі теорії прототипів, що дозволяє обирати інтерпретацію концепту залежно від конкретної ситуації прийняття рішення. Ця модель, на відміну від існуючих, дає змогу не тільки зберігати контекстно-залежні інтерпретації концептів, але й надає можливість агенту будувати нові інтерпретації.

У **першому розділі** проведено аналіз систем підтримки прийняття рішень, онтологій та методів концептуалізації проблемних областей у них. Використання онтологій у системах підтримки прийняття рішень дозволяє зменшити розбіжності у розумінні термінів та концепцій, що використовуються у предметній області. Проте, проектування онтологій пов'язане зі складнощами, які розв'язуються з урахуванням особливостей предметної області. Результати аналізу показують, що інтерпретація змісту концептів онтології залежить від конкретної ситуації прийняття рішення та повинна обиратися динамічно після ідентифікації ситуації. Застосування онтологічних мереж надає можливість вибору варіанту інтерпретації концепту

залежно від ситуації, що є важливим фактором у підвищенні ефективності системи підтримки прийняття рішень.

Аналіз існуючих підходів до побудови систем підтримки прийняття рішень показав, що необхідно удосконалювати процеси концептуалізації проблемних областей з огляду на контекстну залежність та середовище функціонування інтелектуального агента - децидента. Це дозволить забезпечити ефективне прийняття рішень у процесі роботи інтелектуального агента. Наприклад, врахування контекстуальних факторів може допомогти уникнути помилкових рішень, а адаптація до середовища дозволить інтелектуальному агенту ефективніше взаємодіяти зі своїм оточенням. У цілому, аналіз підходів до побудови систем підтримки прийняття рішень демонструє, що вдосконалення процесів концептуалізації проблемних областей є ключовим етапом для забезпечення ефективної роботи інтелектуального агента - децидента у процесі прийняття рішення.

У **другому розділі** досліджено застосування онтологічного моделювання у задачах підтримки прийняття рішень з метою визначення особливостей моделювання предметних областей в умовах багатозначності та контекстної залежності. Онтологічне моделювання є ефективним інструментом для підтримки прийняття рішень в умовах багатозначності та контекстної залежності. Цей підхід допомагає визначити особливості предметної області, збирати та структурувати знання і використовувати їх для прийняття рішень.

Аналіз методів застосування онтологічного моделювання показав, що однією з головних переваг онтологічного моделювання є можливість точного та контекстно-орієнтованого подання моделей предметних областей у процесі прийняття рішення. Однак, при використанні онтологічного моделювання для концептуалізації проблемних ситуацій необхідно мати на увазі, що створення онтології є складним процесом, який вимагає багато ресурсів та часу. Крім того, у

разі якщо ми маємо справу з складною предметною областю, можуть виникнути проблеми з узгодженням онтологій, що може призвести до неточностей у визначенні понять та зв'язків між ними. Тому, для досягнення точності та контекстно-орієнтованості у використанні онтологічного моделювання для концептуалізації проблемних ситуацій, необхідно проводити багатократне та багаторівневе узгодження онтологій. Це допоможе забезпечити точне та повне подання моделей предметних областей, що, у свою чергу, дозволить прийняти більш обґрунтовані та ефективні рішення в проблемних ситуаціях.

У **третьому розділі** презентовано концептуальну модель формування онтології проблемної ситуації, яка базується на декількох вихідних онтологіях. У контексті цієї моделі, наведено формальні визначення онтології проблемної ситуації, мережі онтологій, відповідності між елементами онтології та правил відповідності.

Подано структуру системи підтримки прийняття рішень на основі онтологічних мереж, а також визначено процес підтримки прийняття рішень у випадку застосування мереж онтологій. Центральним елементом системи є база знань, яка містить моделі ситуації та посилання на зовнішні онтології з мережі для кожної такої моделі. Посилання на зовнішні онтології забезпечують отримання актуальних знань з пов'язаних онтологій.

У рамках даної моделі, правила відповідності визначені як відображення між підмножинами концептів, відношень та функцій інтерпретації двох онтологій. При настанні проблемної ситуації, онтологія ситуації формується динамічно з урахуванням наявного контексту ситуації, що дозволяє використовувати актуальні знання з пов'язаних онтологій. Розглянута модель забезпечує ефективне формування та підтримку онтології проблемної ситуації з використанням знань з пов'язаних онтологій.

Запропоновані підходи до конструювання систем підтримки прийняття рішень, що базуються на використанні мереж онтологій, надають можливість динамічного

вибору концептів та відношень, що відповідають поточному контексту ситуації. Результати можна ефективно використовувати для створення систем підтримки прийняття рішень, які потребують інтеграції даних з різних предметних областей та в умовах багатозначності.

У **четвертому розділі** було розроблено прототип програмного забезпечення, за допомогою таких інтерфейсів, програм та мов програмування, як: API Google Search, Azure Cognitive Services, Protégé, ML.NET та Windows Forms.

Модуль збору даних збирає дані з веб-сторінок за допомогою API Google Search, це ефективний спосіб отримання великої кількості даних. Описано фактори, які допоможуть зробити цей процес ефективним і забезпечити якість даних.

Модуль обробки даних обробляє ці дані використовуючи методи обробки природної мови, щоб перетворити неструктуровану інформацію в структурований формат, для цього було використано Azure Cognitive Services, що включає: розпізнавання іменованих сутностей, визначення залежностей між словами, кластеризацію та тематичний аналіз.

Модуль побудови онтологій використовує програмне забезпечення Protégé. Protégé є безкоштовною платформою, що надає можливість створювати, редагувати та розширювати онтології, а також визначати правила логіки та відносин між об'єктами в онтології. Кожент пошуковий запит створює свою відповідну онтологію. Після створення онтології виконується перевірка та оцінка її якості.

Модуль побудови моделі будує модель з використанням ML.NET. ML.NET - це бібліотека, яка дозволяє розробляти моделі машинного навчання для аналізу даних на .NET-платформі.

Модуль підтримки прийняття рішень це програмний інтерфейс розроблений з використанням Windows Forms. Інтерфейс складається з двох компонент: поле вводу і блок виводу, для того щоб задати питання і отримати відповідь.

Наведено приклади роботи програмного забезпечення, одержано основний функціонал необхідний для системи підтримки прийняття рішень, який дозволяє задавати питання та отримувати на них відповіді. Відповіді зачастию включають в себе рекомендації. Також описано варіанти для майбутнього розширення.

Ключові слова: система підтримки прийняття рішень, онтологія, онтологічна модель, онтологічні мережі, багатозначність, інтелектуальні системи, інтелектуальні агенти.

ABSTRACT

Karpov I. A. Construction of a decision support system based on ontological networks. – Qualifying scientific work on the rights of a manuscript.

Dissertation for the degree of Doctor of Philosophy in the specialty 124 "Systems Analysis" – National University "Lviv Polytechnic", Ministry of Education and Science of Ukraine, Lviv, 2023.

Abstract content. The dissertation is devoted to the development of methods and construction of a decision support system based on ontological networks in conditions of ambiguity.

The growing amount of information that must be taken into account when making decisions determines the relevance of developing intelligent decision support systems. A prerequisite for correct decision-making is the construction of a conceptual model of a problem situation that takes into account all relevant factors. The conceptualization of a problem situation is carried out using the ontology of this situation. To form an ontology of a problem situation, it is advisable to use knowledge from existing ontologies, but this approach can lead to ambiguity in the choice of ontology elements that most accurately match the situation.

The difference between ontologies and expert systems is that they are built as a coherent and widely shared description of some generalized domain of knowledge, usually aimed at solving a specific problem. Thus, the problem of creating and using ontologies as a description of knowledge in subject areas is an important task that contributes to the development of computer engineering, computer science, artificial intelligence, and other scientific disciplines. This opens up new opportunities for the application of advanced intelligent information technologies in various subject areas, including those where these technologies have not been used before.

The scientific novelty of the obtained results is as follows:

1. For the first time, a method of multi-level and step-by-step harmonization of the content of the ontology of the subject area of the decision-making task has been developed, which reduces the number of conceptualization errors. This approach differs from traditional methods of building ontologies in that it provides a more accurate and structured conceptualization of the subject area, which helps to improve the quality of decisions.
2. For the first time, a method based on the modified TOPSIS method for selecting an ontology to support decision-making based on the analysis of similarity of functions and practical use of ontologies by an intelligent agent has been developed, which made it possible to take into account the implicit components of conceptualization and improve the validity of ontology selection.
3. For the first time, a model for representing the context-dependent structure of interpretations of ontology concepts based on the theory of prototypes is proposed, which allows choosing the interpretation of a concept depending on a specific decision-making situation. This model, unlike the existing ones, allows not only storing context-dependent interpretations of concepts, but also enables the agent to build new interpretations.

The **first section** analyzes decision support systems, ontologies and methods of conceptualizing problem areas in them. The use of ontologies in decision support systems helps to reduce the differences in understanding of terms and concepts used in the subject area. However, designing ontologies is associated with difficulties that are solved by taking into account the peculiarities of the subject area. The results of the analysis show that the interpretation of the content of ontology concepts depends on a specific decision-making situation and should be selected dynamically after the situation is identified. The use of ontological networks makes it possible to choose a variant of the concept interpretation depending on the situation, which is an important factor in improving the efficiency of the decision support system.

An analysis of existing approaches to building decision support systems has shown that it is necessary to improve the processes of conceptualizing problem areas, taking into account the contextual dependence and environment of the functioning of an intelligent agent - a decedent. This will ensure effective decision-making in the process of the intelligent agent's work. For example, taking into account contextual factors can help to avoid erroneous decisions, and adapting to the environment will allow an intelligent agent to interact more effectively with its environment. In general, the analysis of approaches to building decision support systems demonstrates that improving the processes of conceptualizing problem areas is a key step to ensure the effective operation of an intelligent agent - a decisional agent in the decision-making process.

The **second section** investigates the application of ontological modeling in decision support tasks in order to determine the features of modeling subject areas under conditions of ambiguity and contextual dependence. Ontological modeling is an effective tool for supporting decision-making in conditions of ambiguity and contextual dependence. This approach helps to identify the features of a subject area, collect and structure knowledge, and use it for decision-making.

The analysis of methods for applying ontological modeling has shown that one of the main advantages of ontological modeling is the ability to accurately and contextually represent models of subject areas in the decision-making process. However, when using ontological modeling to conceptualize problem situations, it should be borne in mind that creating an ontology is a complex process that requires a lot of resources and time. In addition, if we are dealing with a complex subject area, there may be problems with ontology coordination, which can lead to inaccuracies in the definition of concepts and relationships between them. Therefore, in order to achieve accuracy and context-awareness in the use of ontological modeling for conceptualizing problem situations, it is necessary to perform multiple and multi-level ontology alignment. This

will help to ensure accurate and complete representation of subject matter models, which, in turn, will allow for more informed and effective decision-making in problem situations.

The **third section** presents a conceptual model for the formation of a problem situation ontology based on several source ontologies. In the context of this model, formal definitions of the problem situation ontology, ontology network, correspondence between ontology elements and correspondence rules are given.

The structure of a decision support system based on ontology networks is presented, and the process of decision support in the case of using ontology networks is defined. The central element of the system is the knowledge base, which contains situation models and links to external ontologies from the network for each such model. Links to external ontologies provide up-to-date knowledge from related ontologies.

Within this model, the matching rules are defined as a mapping between subsets of concepts, relations, and interpretation functions of two ontologies. When a problem situation occurs, the ontology of the situation is formed dynamically taking into account the existing context of the situation, which allows the use of relevant knowledge from related ontologies. The considered model provides effective formation and maintenance of the ontology of a problem situation using knowledge from related ontologies.

The proposed approaches to the design of decision support systems based on the use of ontology networks provide an opportunity to dynamically select concepts and relations that correspond to the current context of the situation. The results can be effectively used to create decision support systems that require the integration of data from different subject areas and in conditions of ambiguity.

In the **fourth chapter**, a prototype software was developed using interfaces, programs, and programming languages such as: Google Search API, Azure Cognitive Services, Protégé, ML.NET, and Windows Forms.

The data collection module collects data from web pages using the Google Search API, which is an effective way to obtain a large amount of data. The factors that help make this process efficient and ensure data quality are described.

The data processing module processes this data using natural language processing techniques to transform unstructured information into a structured format, using Azure Cognitive Services, which includes: named entity recognition, word dependency detection, clustering, and thematic analysis.

The ontology building module uses Protégé software. Protégé is a free platform that allows you to create, edit, and extend ontologies, as well as define the rules of logic and relationships between objects in an ontology. Each search query creates its own ontology. After the ontology is created, its quality is checked and evaluated.

The model building module builds a model using ML.NET. ML.NET is a library that allows you to develop machine learning models for data analysis on the .NET platform.

The decision support module is a software interface developed using Windows Forms. The interface consists of two components: an input field and an output block to ask a question and get an answer.

Examples of software operation are given, and the basic functionality required for a decision support system, which allows you to ask questions and receive answers, is obtained. The answers often include recommendations. Options for future expansion are also described.

Keywords: decision support system, ontology, ontological model, ontological networks, ambiguity, intelligent systems, intelligent agents.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні результати дисертації:

1. Карпов І., Буров Є. *Використання онтологічних мереж у системах підтримки прийняття рішень в умовах неоднозначності*. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2020. Вип. 7. С. 8–15.
2. Burov Y., Mykich K., Karpov I. *Building a versatile knowledge-based system based on reasoning services and ontology representation transformations*. International Scientific and Technical Conference "Computer Sciences and Information Technologies". 2020. 15th IEEE Intern. sci. and techn. conf. on computer sciences and information technologies CSIT 2020, Lviv-Zbarazh, Ukraine, 23-26 Sept. 2020. Vol. 2. P. 255–260.
3. Карпов І., Буров Є. *Онтології у процесі прийняття рішень*. Вісник Хмельницького національного університету. Серія: «Технічні науки». 2023. № 2, т. 1. С. 149–153.
4. Burov Y., Karpov I. *Contextual concept meaning alignment based on prototype theory*. CEUR Workshop Proceedings. 2023. Vol. 3403 : Computational linguistics and intelligent systems 2023 : proc. of the 7th Intern. conf. on computational linguistics and intelligent systems. Vol. III: Intelligent systems workshop. Kharkiv, Ukraine, Apr. 20-21, 2023. P. 137–146.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Буров Є., Карпов І. *Вибір концептуалізації проблемної ситуації інтелектуальним агентом в задачах прийняття рішення*. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2023. Вип. 13.
2. Карпов І., Буров Є. *Підхід до побудови системи підтримки прийняття рішень на основі онтологій*. Scientific Collection «InterConf». 2022. № 99 : Proc. of the

- 4th Intern. sci. and pract. conf. "International scientific discussion: problems, tasks and prospects", Brighton, Great Britain, Febr. 19-20, 2022. С. 849–855.
3. Карпов І., Буров Є. *Особливості застосування онтологій в системах підтримки прийняття рішень: теоретичний та практичний аспект*. Грааль науки. 2022. № 12-13. Proc. of the III Correspondence intern. sci. and pract. conf. «An integrated approach to science modernization: methods, models and multidisciplinary», Apr. 29th, 2022. С. 307–310.
 4. Карпов І., Буров Є. *Застосування онтологій для прикладних задач*. III CISP Conference «Globalization of scientific knowledge: international cooperation and integration of sciences» III Міжнародна науково-практична конференція. 2021. Вінниця - Відень. С. 344–345.
 5. Карпов І., Буров Є. *Онтологічне моделювання: цілі та засоби*. Scientific Collection «InterConf», (107): with the Proceedings of the 12th International Scientific and Practical Conference «Science and Practice: Implementation to Modern Society». 2022. Manchester, Great Britain. С. 517–519.
 6. Карпов І., Буров Є. *Особливості використання онтологій у системах підтримки прийняття рішень: закономірності та особливості*. I International Scientific and Theoretical Conference «Technologies and strategies for the implementation of scientific achievements» I Міжнародна науково-теоретична конференція. European Scientific Platform. 2022. Стокгольм, SWE. С. 76–77.
 7. Карпов І., Буров Є. *Проблеми онтологічного підходу у віртуальному навчальному середовищі*. III International Scientific and Practical Conference «Education and science of today: intersectoral issues and development of sciences». III Міжнародна науково-практична конференція. 2022. Кембридж, GBR. С. 187–188.
 8. Karpov I., Burov Y. *Use of ontological networks in decision support systems under ambiguity*. Debats scientifiques et orientations prospectives du developpement

scientifique : materiales de la conferencia internacional científica y práctica -
"ΛΟΓΟΣ". Paris. Vol. 3. 2021. C. 97–103.

References

Scientific papers in which the main results of the dissertation are published:

1. Karpov I., Burov E. *The use of ontological networks in decision support systems under conditions of ambiguity*. Bulletin of Lviv Polytechnic National University. Series: Information systems and networks. 2020. Issue 7. C. 8-15.
2. Burov Y., Mykich K., Karpov I. *Building a versatile knowledge-based system based on reasoning services and ontology representation transformations*. International Scientific and Technical Conference "Computer Sciences and Information Technologies". 2020. 15th IEEE Intern. sci. and techn. conf. on computer sciences and information technologies CSIT 2020, Lviv-Zbarazh, Ukraine, 23-26 Sept. 2020. Vol. 2. P. 255–260.
3. Karpov I., Burov E. *Ontologies in the decision-making process*. Bulletin of Khmelnytsky National University. Series: "Technical Sciences". 2023. № 2, т. 1. C. 149-153.
4. Burov Y., Karpov I. *Contextual concept meaning alignment based on prototype theory*. CEUR Workshop Proceedings. 2023. Vol. 3403 : Computational linguistics and intelligent systems 2023 : proc. of the 7th Intern. conf. on computational linguistics and intelligent systems. Vol. III: Intelligent systems workshop. Kharkiv, Ukraine, Apr. 20-21, 2023. P. 137–146.

Scientific papers certifying the approbation of the dissertation materials:

1. Burov Y., Karpov I. *Choosing the conceptualization of a problem situation by an intelligent agent in decision-making tasks*. Bulletin of the National University "Lviv Polytechnic". Series: Information systems and networks. 2023. Issue 13.
2. Karpov I., Burov Y. *Approach to building a decision support system based on ontologies*. Scientific Collection "InterConf". 2022. № 99 : Proc. of the 4th Intern. sci. and pract. conf. "International scientific discussion: problems, tasks and prospects", Brighton, Great Britain, Febr. 19-20, 2022. C. 849-855.

3. Karpov I., Burov Y. *Features of the use of ontologies in decision support systems: theoretical and practical aspects*. The Grail of Science. 2022. № 12-13. Proc. of the III Correspondence intern. sci. and pract. conf. "An integrated approach to science modernization: methods, models and multidisciplinary", Apr. 29th, 2022. C. 307-310.
4. Karpov I., Burov Y. *Application of relativities for applied problems*. III CISP Conference "Globalization of scientific knowledge: international cooperation and integration of sciences" III International Scientific and Practical Conference. 2021. Vinnytsia - Vienna. C. 344-345.
5. Karpov I., Burov Y. *Ontological modeling: goals and means*. Scientific collection "InterConf", (107): based on the materials of the 12th International Scientific and Practical Conference "Science and Practice: Implementation in modern society". 2022. Manchester, UK. C. 517-519.
6. Karpov I., Burov Y. *Features of the use of ontologies in decision support systems: regularities and features*. I International Scientific and Theoretical Conference "Technologies and strategies for the implementation of scientific achievements" I International Scientific and Theoretical Conference. European Scientific Platform. 2022. Stockholm, SWE. C. 76-77.
7. Karpov I., Burov Y. *Problems of the ontological approach in the virtual learning environment*. III International Scientific and Practical Conference "Education and science of today: intersectoral issues and development of sciences". III International Scientific and Practical Conference. 2022. Cambridge, GBR. C. 187-188.
8. Karpov I., Burov Y. *Use of ontological networks in decision support systems under ambiguity*. Debats scientifiques et orientations prospectives du developpement scientifique : materiales de la conferencia internacional científica y práctica - "ΛΟΓΟΣ". Paris. Vol. 3. 2021. C. 97–103.

ЗМІСТ

ВСТУП	20
РОЗДІЛ 1. Система підтримки прийняття рішень та онтології: теоретичний та практичний аспект	26
1.1. Основні поняття системи підтримки прийняття рішень	26
1.2. Основні поняття, побудова та розвиток онтології	36
1.3. Узгодження онтологій та операції з онтологіями	52
Висновки до розділу 1	58
РОЗДІЛ 2. Застосування онтологій для пошуку, опрацювання та аналізу інформації у процесі прийняття рішень	59
2.1. Застосування онтологій для систем підтримки прийняття рішень	59
2.2. Узгодження онтологій у процесі прийняття рішень	69
2.3. Аналіз проблем використання онтологій в системах підтримки прийняття рішень	78
Висновки до розділу 2	84
РОЗДІЛ 3. Методи та алгоритми побудови системи підтримки прийняття рішень на основі онтологічних мереж	85
3.1. Метод багаторівневого узгодження структур на основі онтологічних мереж	85
3.2. Метод вибору провайдера знань на основі модифікованого методу TOPSIS	111
3.3. Модель подання контекстно залежного змісту концептів онтології на основі теорії прототипів	118
Висновки до розділу 3	123
РОЗДІЛ 4. Практична реалізація прототипу системи підтримки прийняття рішень на основі онтологічних мереж	124
4.1. Опис та структура програмного забезпечення	124
4.2. Вхідні дані та очікуваний результат експерименту	133
4.3. Аналіз результатів та застосовності розробленого програмного забезпечення	140
Висновки до розділу 4	154
ВИСНОВКИ	155
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	158
ДОДАТОК А	168
ДОДАТОК Б	172

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

СППР – система підтримки прийняття рішень.

ІАС – інформаційно-аналітична система.

ІСР – інтегроване середовище розробки.

БД – база даних.

СУБД – система управління базами даних.

СОД – система обробки даних.

ІАД – інтелектуальний аналіз даних.

ШІ – штучний інтелект.

ПЗ – програмне забезпечення.

ВСТУП

Тема дослідження: Побудова системи підтримки прийняття рішень на основі онтологічних мереж.

Актуальність дослідження: Розвиток інтелектуальних систем підтримки прийняття рішень є актуальним завданням у зв'язку зі зростанням обсягу інформації, що впливає на процес прийняття рішень. Для забезпечення коректності рішень необхідно побудувати концептуальну модель проблемної ситуації, яка враховує всі важливі фактори. Одним із способів концептуалізації є використання онтологій, які описують ситуацію з точки зору знань. Однак, формування онтології може стикатися з проблемою багатозначності вибору елементів, які найкраще відповідають ситуації. Ця багатозначність становить виклик при створенні систем прийняття рішень, оскільки різні ситуації можуть тлумачити інформацію по-різному.

Застосування онтологій у системах підтримки прийняття рішень є активним напрямом досліджень. У сучасній літературі найвідомішими авторами, що працюють у галузі онтологій, є Т. Грубер, Н. Гуаріно, Б. Сміт, Дж. Сова, А. Гомес-Перес, Р. Лауріні, а також Ф. Фонсека, Н. Каза та Х. Шиверс, які активно досліджують цю тему в останні роки.

Незважаючи на наявність спільної інформаційної компоненти, існуюче програмне та інформаційне забезпечення часто має проблеми з коректним зв'язком між базами даних та програмними додатками. Недоліки в наукових знаннях ускладнюють їх використання, а модернізація залежить від доопрацювання, що часто є дорогим та витратним заходом. Вирішення питань появи нових об'єктивних знань про предметну область та проблем, що виникають під час розробки інформаційних систем, потребує виявлення, формалізації, опису та фіксації нових вимог. Уніфікація підходів до вирішення складних задач потребує наявності інтелектуальної компоненти, зокрема з використанням знань орієнтованих систем. Таке прагнення спонукає до необхідності створення загальноновживаної термінології,

категоризації понять та уніфікації їх сенсу, щоб мати загальний та доступний опис явищ та проблем різних предметних областей. Створення комп'ютерних онтологій, що входять до категорії систем, ґрунтованих на знаннях, служить цієї меті. Відмінність онтологій від експертних систем полягає в тому, що вони побудовані як узгоджений та широко розділений опис деякого узагальненого домену знань, зазвичай спрямованого на розв'язання конкретної проблеми. Таким чином, проблема створення та використання онтологій як опису знань у предметних областях є важливим завданням, що сприяє розвитку комп'ютерної інженерії, інформатики, штучного інтелекту та інших наукових дисциплін. Це відкриває нові можливості для застосування передових інтелектуальних інформаційних технологій у різних предметних галузях, включаючи ті, де раніше ці технології не застосовувалися.

Мета дослідження: Розробка методів та побудова системи підтримки прийняття рішень на основі онтологічних мереж в умовах багатозначності.

Об'єкт дослідження: Процеси концептуалізації предметної області у системах підтримки прийняття рішення в умовах багатозначності.

Предмет дослідження: Методи та засоби концептуалізації предметної області в умовах багатозначності на основі онтологічних мереж.

Завдання дослідження:

1. Проведення аналізу систем підтримки прийняття рішень та методів концептуалізації проблемних областей у них.

Завдання полягає у дослідженні систем підтримки прийняття рішень та методів концептуалізації проблемних областей, з метою визначення їх ефективності та можливостей застосування. В процесі аналізу доцільно проаналізувати різноманітні системи та методи, їх переваги та недоліки, можливості застосування в різних областях, а також розглянути проблемні області, які можуть бути вирішені за допомогою цих систем та методів. Результатом проведення аналізу буде формування часткової картини про

можливості та обмеження систем підтримки прийняття рішень та методів концептуалізації проблемних областей, що дозволить зробити висновки про їх доцільність та ефективність в конкретних умовах.

2. Дослідити застосування онтологічного моделювання у задачах підтримки прийняття рішень з метою визначення особливостей моделювання предметних областей в умовах багатозначності та контекстної залежності.

Завдання полягає у дослідженні застосування онтологічного моделювання для підтримки прийняття рішень в умовах, коли предметна область має багатозначний та контекстуальний характер. Онтологічне моделювання допомагає у визначенні особливостей предметної області, збиранні та структуруванні знань, а також у використанні цих знань при прийнятті рішень. Дослідження має на меті проведення аналізу ефективності використання онтологічного моделювання для підтримки прийняття рішень у багатозначних та контекстуальних умовах, а також рекомендації щодо його застосування.

3. Розробити методи онтологічного моделювання предметних областей у системах підтримки прийняття рішень в умовах багатозначності.

Завдання полягає в розробці методів моделювання предметних областей з використанням онтологій в системах, що підтримують процес прийняття рішень. Важливо, щоб ці методи були ефективними в умовах багатозначності, коли інформація не є повною або точною, або коли існує декілька можливих варіантів рішень. Метою такого моделювання є покращення якості прийнятих рішень, зменшення ризику помилкових рішень та забезпечення більш ефективного використання доступної інформації.

4. Реалізувати та апробувати розроблені методи на практиці.

Завдання полягає в тому, щоб створити прототип програмного забезпечення, яке буде включати в себе попередньо розроблені методи.

Методи дослідження:

1. У процесі аналізу систем підтримки прийняття рішень застосовано методи системного аналізу та методи аналізу бізнес-процесів.

Методи системного аналізу використовуються для дослідження складних систем, з метою визначення структури та функціонування системи, що дозволяє отримати більш повне розуміння процесів, які відбуваються всередині системи. Методи аналізу бізнес-процесів, з іншого боку, використовуються для визначення потреб користувачів системи та для з'ясування того, як краще забезпечити їх потреби. Використання цих двох методів дозволяє побудувати більш ефективні та корисні системи підтримки прийняття рішень.

2. Для дослідження особливостей використання онтологій у системах прийняття рішень застосовано методи онтологічного моделювання, зокрема методи узгодження онтологій, нечіткі, контекстні та розподілені онтології.

Метод узгодження онтологій дозволяє вирішити проблему несумісності між онтологіями, що використовуються в різних системах. Нечітка онтологія - це модель знань, яка дозволяє виражати нечіткі, або неоднозначні поняття, які можуть мати різні рівні приналежності до класів або категорій. Контекстна онтологія - це модель знань, яка враховує контекст, в якому використовується поняття. Розподілена онтологія - це модель знань, яка розглядає знання як сукупність елементів, розподілених по всій системі, і не прив'язує їх до конкретних вузлів або концептів. Використання нечітких, контекстних та розподілених онтологій дозволяють описувати складні та неоднозначні факти, а також залежності в інформаційних системах. Методи є корисними для створення більш точних та ефективних систем прийняття рішень на основі онтологічного моделювання.

3. При розробці методів онтологічного моделювання проблемних областей в умовах багатозначності використано методи теорії прототипів, теорії

концептів, моделювання бізнес-функцій.

Метод теорії прототипів використовується при онтологічному моделюванні для визначення типових представників об'єктів, що належать до певної категорії. Цей метод дозволяє встановити суттєві характеристики, які є спільними для всіх екземплярів категорії. Метод теорії концептів використовується для визначення властивостей, що відокремлюють одну категорію від іншої. Цей метод дозволяє встановити відношення між різними категоріями і зв'язки між ними. Метод моделювання бізнес-функцій використовується для визначення функцій, що повинні виконувати система, яка будується на основі онтологій. Цей метод дозволяє встановити, які процеси повинні бути автоматизовані та які ресурси потрібні для їх реалізації.

Наукова новизна одержаних результатів:

4. Вперше розроблено метод багаторівневого та поетапного узгодження змісту онтології предметної області задачі прийняття рішення, що дозволяє зменшити кількість помилок концептуалізації. Цей підхід відрізняється від традиційних методів побудови онтологій, тим, що забезпечує більш точну та структуровану концептуалізацію предметної області, що сприяє покращенню якості прийнятих рішень.
5. Вперше розроблено метод, що базується на модифікованому методі TOPSIS для вибору онтології для підтримки прийняття рішення на основі аналізу подібності функцій та практичного використання онтологій інтелектуальним агентом, що дало змогу врахувати неявні складові концептуалізації та покращити обґрунтованість вибору онтології.
1. Вперше запропоновано модель подання контекстно залежної структури інтерпретацій концептів онтології на основі теорії прототипів, що дозволяє обирати інтерпретацію концепту залежно від конкретної ситуації прийняття рішення. Ця модель, на відміну від існуючих, дає змогу не тільки зберігати

контекстно-залежні інтерпретації концептів, але й надає можливість агенту будувати нові інтерпретації.

Практичне значення одержаних результатів:

Розроблено прототип програмного забезпечення, яке використовує навчену з використанням онтологічних мереж модель штучного інтелекту для аналізу підприємства. Це досягнуто завдяки використанню онтологій для тренування моделі, що забезпечило покращення ефективності та точності роботи системи, а також дозволило зменшити необхідну кількість навчальних даних.

Розроблене ПЗ може забезпечити підприємствам інформацію про нові тенденції на ринку та допомогти налаштувати бізнес-стратегії з метою підвищення ефективності та точності роботи. Це може призвести до покращення задоволеності клієнтів та збільшення конкурентоспроможності.

РОЗДІЛ 1. Система підтримки прийняття рішень та онтології: теоретичний та практичний аспект

1.1. Основні поняття системи підтримки прийняття рішень

Тією чи іншою мірою системи підтримки прийняття рішень (СППР) присутні у будь-якій інформаційно-аналітичній системі (ІАС). У міру розвитку будь-якої сфери людської життєдіяльності, упорядкування структури організації та налагодження міжкорпоративних зв'язків проблема розробки та впровадження СППР стає особливо актуальною. Сфера освіти не є винятком. Одним із підходів до створення ІАС стало використання сховищ даних (Data Warehouse) та алгоритмів інтелектуального аналізу даних (Data Mining).

СППР можна, залежно від даних, з якими вони працюють, поділити на оперативні (Executive Information Systems), призначені для негайного реагування на поточну ситуацію, та стратегічні (Decision Support System) - засновані на аналізі великої кількості інформації з різних джерел залученням відомостей, що містяться в системах, що акумулюють досвід вирішення проблем [13, 14].

СППР першого типу дістали назву інформаційних систем управління. По суті, вони є кінцевими наборами звітів, побудовані на підставі даних з транзакційної інформаційної системи підприємства або OLTP (Online Transaction Processing) – системи, що в ідеалі адекватно відображає в режимі реального часу всі аспекти виробничого циклу підприємства. Для ІСР (інтегроване середовище розробки) характерні такі основні риси [15]:

- Звіти, як правило, базуються на стандартних для організації запитах; число останніх невелике;
- ІСР представляє звіти у максимально зручному вигляді, що включає, поряд з таблицями, ділову графіку, мультимедійні можливості тощо;
- Як правило, ІСР орієнтовані на конкретний вертикальний ринок, наприклад, управління ресурсами, фінанси.

СППР другого типу передбачають значне оброблення даних, спеціально перетворених для зручного використання під час прийняття рішень. Невід'ємною складовою СППР цього рівня є правила прийняття рішень, які на основі агрегованих даних надають менеджерському складу висновки та наділені властивостями штучного інтелекту. Такі системи розробляються лише в тому випадку, якщо структура організації вже достатньо визначена та є основи для узагальнення та аналізу як даних, так і процесів їх обробки. Якщо інформаційна система ресурсного планування є просто розвитком системи оперативного управління виробничими процесами, то СППР у сучасному розумінні - це механізм розвитку, який включає у себе певну частину управляючої інформаційної системи, велику систему зовнішніх зв'язків підприємства та навіть технологічні процеси його розвитку.

Якщо подивитися на СППР з функціонального боку, можна виділити такі компоненти: сервер сховища даних; інструментарій OLAP; інструментарій Data Mining [16].

Сховища даних – це архітектура побудови інформаційних систем, що отримала розвиток внаслідок бажання кінцевих користувачів мати безпосередній одноманітний доступ до необхідних даних, джерела походження яких організаційно і територіально розподілені, а аналіз яких може сприяти прийняттю рішень. Білл Інмон (W. Inmon), автор концепції сховищ даних, визначив їх як "предметно-орієнтовані, інтегровані, незмінні, що підтримують хронологію набори даних, організовані з метою підтримки управління", покликані виступати в ролі "єдиного джерела істини", що забезпечує менеджерів та аналітиків достовірною інформацією, необхідною для оперативного аналізу та прийняття рішень [17, 18]. Річард Хакаторн (R. Hackthorn), інший основоположник цієї концепції, писав, що мета сховищ даних – забезпечити для організації "єдиний образ існуючої реальності" [19].

Такий образ реальності має бути представлений єдиною інформаційною моделлю. Проблематика моделювання даних пов'язана з таким поданням інформації, яке найбільш природно відображає реальний світ і водночас може підтримуватися комп'ютерними засобами. Отже, модель, з одного боку, повинна описувати складну структуру даних та їх взаємозв'язок, а з іншого - бути досить просто організованою, щоб забезпечити швидкий та надійний доступ до даних, що зберігаються. Протягом усієї історії розвитку інформаційних систем робилися різні спроби вирішення цієї суперечності. Так виникли "класичні" моделі даних – ієрархічна, мережева та реляційна, реалізовані різними СУБД. Семантичні характеристики кожної з цих моделей мають свої переваги та недоліки, проте сьогодні чільне становище займає реляційна модель, оскільки її концепція найкраще формалізується традиційними механізмами математичної логіки. Подання даних у реляційній формі має достатню гнучкість, але не забезпечує належної інтуїтивності сприйняття описуваних об'єктів з погляду кінцевого користувача інформаційних систем.

Метою створення сховища даних є інтеграція, актуалізація та узгодження оперативних даних із різноманітних джерел з метою формування єдиного несуперечливого погляду на об'єкт управління в цілому. За основу концепції сховища даних прийнято визнання необхідності розділення наборів даних, що використовуються для транзакційної обробки, та наборів даних, що використовуються в системах підтримки прийняття рішень [20].

Незважаючи на відмінності у підходах та реалізаціях, усім сховищам даних властиві спільні риси.

Предметна орієнтованість є характерною особливістю організації даних у сховищі, де інформація структурована відповідно до основних аспектів діяльності, що відрізняє його від оперативних баз даних, де дані організовані згідно з процесами. Завдяки такій організації даних в сховищі вдається спростити процес аналізу та підвищити швидкість виконання аналітичних запитів. При цьому

використовуються схеми організації даних, які відрізняються від схем оперативних систем. Наприклад, у випадку зберігання даних у реляційних СУБД можуть бути використані схеми "зірки" (star) або "сніжинки" (snowflake), а також дані можуть бути збережені в багатовимірних СУБД у n-мірних кубах.

Інтегрованість, прив'язка до часу та незмінність є важливими аспектами сховища даних, яке забезпечує зручний та ефективний аналіз великих обсягів даних. Вихідні дані отримуються з оперативних баз даних та піддаються обробці, щоб надати їм єдиний формат та забезпечити їхню агрегацію. Завантаження цих даних у сховище відбувається з прив'язкою до певного періоду часу, що дозволяє аналізувати динаміку розвитку організації. Незмінність даних є ключовим аспектом сховища. Після того, як дані були завантажені до сховища, вони більше не можуть бути змінені. Це відрізняє сховище від оперативних баз даних, де дані постійно змінюються. Така стійкість даних у сховищі полегшує їхній аналіз, оскільки забезпечує стабільність результатів.

Концепція сховища даних включає не тільки єдиний логічний погляд на організацію даних, а й реалізацію єдиного інтегрованого джерела даних. У зв'язку з цим, альтернативним підходом до формування єдиного погляду на корпоративні дані є створення віртуального джерела, яке базується на розподілених базах даних різних систем обробки даних. При цьому кожен запит до такого джерела динамічно транслюється у запити до вихідних баз даних, а отримані результати на льоту узгоджуються, зв'язуються, агрегуються і повертаються до користувача. Однак, цей підхід має ряд суттєвих недоліків, які потрібно враховувати [21].

Час обробки запитів до розподіленого сховища значно перевищує відповідні показники централізованого сховища. Крім того, структури баз даних СОД (системи обробки даних), розраховані на інтенсивне оновлення одиночних записів, високою мірою нормалізовані, тому в аналітичному запиті до них потрібно об'єднання великої кількості таблиць, що також призводить до зниження швидкодії.

Інтегрований погляд на розподілене сховище можливий лише при виконанні вимог постійного зв'язку всіх джерел даних у мережі. Таким чином, тимчасова недоступність хоча б одного з джерел може зробити роботу ІАС неможливою, або призвести до помилкових результатів.

Виконання складних аналітичних запитів над таблицями СОД споживає великий обсяг ресурсів сервера БД і призводить до зниження швидкодії СОД, що є неприпустимим, оскільки час виконання операцій на СОД часто дуже критично.

Різні СОД можуть підтримувати різні формати, дані можуть бути неузгоджені. Дуже часто на те саме питання може бути отримано кілька варіантів відповіді, що може бути пов'язано з несинхронністю моментів оновлення даних, відмінностями в трактуванні окремих подій, понять і даних, помилками при введенні і т. д. У такому випадку мета - формування єдиного несуперечливого погляду на об'єкт управління - може бути не досягнутою [22].

Головним же недоліком слід визнати практичну неможливість огляду тривалих історичних послідовностей, бо за фізичної відсутності центрального сховища доступні ті дані, які у момент запиту є у реальних БД пов'язаних СОД. Основне призначення СОД - оперативна обробка даних, тому вони не можуть дозволити собі розкіш зберігати дані за тривалий (більше кількох місяців) період; старі дані вивантажуються в архів і видаляються з транзакційної БД. Що стосується аналітичної обробки, для неї найцікавіший погляд на об'єкт управління в історичній ретроспективі.

Отримання даних з різних джерел та їх подальша обробка є важливою задачею управління даними. Сховище даних є засобом для збору та зберігання цих даних відповідно до встановленого регламенту. За цим регламентом до сховища збираються дані з баз даних різних систем оперативної обробки. В сховищі забезпечується хронологічний порядок даних: разом з поточними даними зберігається час, до якого вони відносяться. Це дозволяє отримати доступ до

необхідних даних про об'єкт управління в одному місці та в єдиному форматі, що сприяє їх узгодженню та агрегації до мінімально необхідного рівня узагальнення.

Полегшеним варіантом сховища даних можуть бути вітрини даних (Data Mart), тобто тематичні БД, які містять інформацію, що належить до окремих аспектів діяльності організації. Концепція вітрин даних була запропонована Forrester Research в 1991. При цьому головна ідея полягала в тому, що вітрини даних містять тематичні підмножини заздалегідь агрегованих даних, що за розмірами набагато менші, ніж загальнокорпоративне сховище даних, і, отже, потребують менш продуктивної техніки підтримки. У 1994 році Марк Демарест (М. Demarest) запропонував об'єднати дві концепції та використовувати сховище даних як єдине інтегроване джерело для численних вітрин даних [23].

У найпростішому варіанті для сховищ даних використовується та модель даних, яка є основою транзакційної системи. Якщо транзакційна система функціонує на реляційній СУБД (Access, MySQL, PostgreSQL і тощо), найскладнішим завданням стає виконання запитів ad-hoc, оскільки неможливо заздалегідь оптимізувати структуру БД так, щоб усі запити працювали ефективно.

Однак при створенні СППР було виявлено, що існує залежність між частотою запитів та ступенем агрегованості даних, з якими запити оперують, а саме чим агрегованішими є дані, тим частіше запит виконується. Іншими словами, коло користувачів, що працюють із узагальненими даними, ширше, ніж те, для якого потрібні детальні дані. Це спостереження лягло основою для підходу до пошуку та вибірці даних, названого оперативною аналітичною обробкою (Online Analytical Processing, OLAP).

OLAP-системи базуються на двох основних принципах [24]. По-перше, всі дані, що необхідні для прийняття рішень, агрегуються заздалегідь на всіх відповідних рівнях і організовуються таким чином, щоб забезпечити максимально швидкий доступ до них. По-друге, мова маніпулювання даними заснована на

використанні бізнес-понять. В основі OLAP лежить поняття гіперкуба, або багатовимірного куба даних. У комірках цього куба зберігаються аналізовані числові дані, наприклад, обсяги продажів. Вимірювання складається з набору значень інших даних, таких як найменування дисциплін та назв семестрів року. У найпростішому випадку двовимірного куба (квадрату) ми можемо отримати таблицю, яка показує значення успішності студентів з дисциплін та семестрів. Подальше ускладнення моделі даних може йти за кількома напрямками:

- Збільшення кількості вимірів - дані про продаж не тільки за семестрами та дисциплінами, а й за групами;
- Ускладнення вмісту осередку – наприклад, нас може цікавити не лише загальна успішність, але й, скажімо, успішність з дисциплін різних циклів. І тут у осередку буде кілька значень;
- Введення ієрархії в межах одного виміру - загальне поняття час природно пов'язане з ієрархією значень: рік складається з семестрів, семестр з місяців і т.д.

В даний момент обговорюється логічна модель даних, а не їх фізична структура зберігання. Зокрема, уточнюється інтерфейс моделі даних, який визначає базові операції, такі як поворот, проекція, розкриття (Drill-Down), згортка (roll-up/drill-up) та перетин (slice-and-dice). При проекції значення в осередках, які розташовані на осі проекції, підсумовуються за певним законом. Під час розкриття одне зі значень виміру замінюється сукупністю значень з наступного рівня ієрархії виміру, що відповідно замінює значення в осередках гіперкубу. Операція згортки є зворотною до операції розкриття.

Існує три способи зберігання даних в OLAP-системах або три архітектури OLAP-серверів [25]:

- MOLAP (Multidimensional OLAP);
- ROLAP (Relational OLAP);

- HOLAP (Hybrid OLAP).

У випадку MOLAP, вихідні та багатовимірні дані зберігаються в багатовимірній БД або багатомірному локальному кубі. Такий спосіб зберігання забезпечує високу швидкість виконання операцій OLAP. Але багатовимірна база в цьому випадку найчастіше буде надмірною. Куб, побудований на її основі, сильно залежатиме від кількості вимірювань. При збільшенні кількості вимірювань обсяг куба буде експонентно зростати. Іноді це може призвести до "лавиноподібного" зростання обсягу даних, що паралізує в результаті запити користувачів.

Для систем ROLAP гіперкуб є інтерфейсом користувача, який емулюється на звичайній реляційній СУБД (системі управління базами даних). Ця структура дозволяє зберігати значні обсяги даних, але її недолік полягає в низькій та неоднаковій ефективності операцій OLAP. Досвід експлуатації продуктів ROLAP показав, що вони більше підходять на роль інтелектуальних генераторів звітів, ніж дійсно оперативних засобів аналізу. Вони застосовуються у таких областях, як телекомунікації та фінанси, де кількість даних є значною, а висока ефективність запитів не є першочерговим завданням.

Що стосується використання гібридної архітектури, тобто, у HOLAP-продуктах, вихідні дані залишаються в реляційній базі, а агрегати розміщуються у багатовимірній. Побудова OLAP-куба виконується на запит, а OLAP-засоби на основі реляційних та багатовимірних даних. Такий підхід дозволяє уникнути вибухового зростання даних. У цьому можна досягти оптимального часу виконання клієнтських запитів.

Багато сучасних OLAP-серверів підтримують всі три способи зберігання даних: MOLAP, ROLAP та HOLAP. Одним з найпоширенішим в даний час серверним рішенням є OLAP-сервер корпорації Microsoft.

За допомогою OLAP-сервера може бути організовано фізичне зберігання обробленої багатовимірної інформації, що дозволяє швидко видавати відповіді на

запити користувача. Крім того, передбачається перетворення даних з реляційних та інших баз на багатовимірні структури в режимі реального часу.

При побудові програмно-технологічної архітектури сховища слід мати на увазі, що система прийняття рішення, на які б візуальні засоби подання вона не спиралася, повинна надати користувачеві можливість деталізації інформації.

Інтелектуальний аналіз даних (ІАД) зазвичай визначають як метод підтримки прийняття рішень, що базується на аналізі залежностей між даними. У межах такої загальної системи, формулювання простого аналізу звітів, побудованих за базою даних, також можна розглядати як різновид ІАД. Щоб перейти до розгляду більш сучасних технологій ІАД, подивимося, як можна автоматизувати пошук залежностей між даними [16, 26].

Існує два підходи до аналізу даних. У першому підході користувач самостійно формулює гіпотези про можливі залежності між даними. Традиційні методи аналізу використовували цей підхід, де гіпотези приводили до побудови звіту, аналіз звіту дозволяв висувати нові гіпотези тощо. У разі використання розвинених інструментів аналізу даних, таких як OLAP, користувач повністю контролює процес пошуку залежностей між даними. В багатьох системах інтелектуального аналізу даних (ІАД) автоматизовано перевірку достовірності гіпотез, що дозволяє оцінити ймовірність певних залежностей в базі даних. Наприклад, можна зробити висновок, що ймовірність того, що зростання продажів продукту А обумовлене зростанням продажів продукту В, становить 0.25.

Другий підхід полягає в тому, що залежність між даними шукаються автоматично. Кількість продуктів, що виконують автоматичний пошук залежностей, говорить про зростаючий інтерес виробників та споживачів до систем саме такого типу.

ІАД розподіляється на три основні категорії: пошук залежностей (відкриття), прогнозування (моделювання) та аналіз аномалій (дослідження). Пошук

залежностей полягає у автоматичному виявленні залежностей, шляхом перегляду бази даних. Проте, головною складністю є відбір значущих залежностей з величезної кількості наявних у базі даних. Прогнозування передбачає можливість користувача запитати систему, щоб отримати значення, що відсутні в записі з певними пустими полями. Система, аналізуючи вміст бази даних, надає правдоподібні передбачення щодо цих значень. Аналіз аномалій полягає у пошуку підозрілих даних, які значно відрізняються від стійких залежностей.

Використання технологій ІАД сприяє вирішенню різноманітних задач з використанням методів математичної статистики, теорії ймовірності та штучного інтелекту. Найбільш поширеним є застосування методів класифікації, регресії, кластеризації, пошуку асоціативних правил, прогнозування послідовностей та аналізу відхилень [27].

Класифікація дозволяє віднести об'єкт до одного з відомих класів в залежності від його характеристик. Регресія допомагає прогнозувати значення вихідного параметру об'єкта за допомогою вхідних параметрів. Кластеризація групує об'єкти на кластери за їх характеристиками, що дозволяє краще зрозуміти дані. Пошук асоціативних правил виявляє закономірності між пов'язаними об'єктами і може використовуватися для прогнозування подій. Прогнозування послідовностей дозволяє знайти залежності між подіями в формі правил, аналіз відхилень забезпечує аналіз даних на наявність нехарактерних шаблонів.

Незважаючи на те, що для кожної прикладної задачі аналізу даних підходить переважно один спосіб вирішення, в цілому розв'язання задач є найчастіше комплексом із методів інтелектуального аналізу даних, що застосовуються.

Необхідно також згадати про можливості інтеграції ІАД до інформаційних систем. Багато методів ІАД виникли із завдань експертного аналізу, тому вхідними даними їм традиційно служать "плоскі" файли даних. З використанням ІАД в СППР

нерідко доводиться спочатку отримувати дані зі сховища, перетворювати в файли необхідних форматів і потім переходити власне до інтелектуального аналізу [28].

1.2. Основні поняття, побудова та розвиток онтології

Онтологія - підрозділ метафізики, що займається буттям, існуванням [29].

Поняття "онтології" в інформатиці чимось спирається на філософію, але є дуже важливі відмінності. В інформатиці онтологія — це спосіб цілісної формалізації знань про якусь предметну галузь, що цілком узгоджується з уявленнями Тома Грубера, родоначальника використання терміна "онтологія" в інформатиці: "онтологія це явна (імпліцитна) специфікація концептуалізації". Ми дозволяємо собі розглядати в онтологіях не тільки поняття (концепти) та відносини між ними, але й інші потрібні об'єкти, наприклад алгоритми, а також вживаємо термін формалізація як розширення ідеї специфікації, оскільки формалізація – це не лише формальний опис, а й можливість щось обчислювати [30].

Онтології пройшли великий шлях розвитку. На перших етапах онтології використовувалися як семантичні мережі та інші способи представлення знань. Особливо важливо, що онтології можна висловити у графічній формі.

Підхід до побудови онтологій не єдиний і не перший — людство винайшло і успішно застосовує різні способи представлення областей знань.

Таксономія — вчення про принципи і практику класифікації та систематизації складноорганізованих ієрархічно співвідносних сутностей.

Таксономії використовують поняття та відносини наслідування між ними. Можна сприймати це так — якийсь клас має атрибути, всі його дочірні класи успадковують ці атрибути, крім цього, можуть мати ще якісь свої. Мовою дискретної математики можна сказати, що математичною моделлю таксономії є ліс орієнтованих корневих дерев.

Таксономію використовують у біології, медицині, інформаційному пошуку. Таксономію також можуть використовувати в об'єктно-орієнтованому програмуванні, оскільки ця парадигма оперує класами, які можуть успадковувати один одного — тобто існує ієрархія.

Однак таксономія має свої суттєві обмеження. Розглянемо концептуальну модель торгівлі. У ній є поняття: "Магазин", "Товар", "Постачальник". Ці три поняття є узагальненнями одне одного, але при цьому між ними є залежність. Цю залежність не може показати таксономія — треба щось більш спільне [31].

Реляційна модель даних — добре відомий спосіб вирішити проблему. У реляційній моделі існують класи, атрибути у класів, і навіть будь-які відносини між класами (крім цього, для відносин можна зазначати, скільки екземплярів кожного класу бере участь у відношенні).

Такий підхід має явні плюси — можна висловити інформацію про будь-які відносини між поняттями. Але при цьому є проблема – немає майже жодних обмежень. Можливо, для завдань навчання нам не потрібні абсолютно будь-які відносини в онтологіях, а достатньо буде всього кількох. Через відсутність обмежень на відносини, уявлення може бути менш наочним. Крім цього, немає перевірки на коректність (якщо немає обмежень, нічого не можна порушити) [32, 33].

Тому для конкретних завдань найчастіше потрібно знайти компроміс між виразністю та обмеженнями — потрібно вирішити, які стосунки залишити, а які прибрати, які властивості мають бути у відносин, якими можуть бути об'єкти, тощо. Для цілей опису онтологій розроблено цілу серію спеціальних мов [34].

Мова RDF (Resource Description Framework). Найпростіша з усіх мов, винайдена у 1990-х роках. Вся інформація подається у вигляді набору триплетів "суб'єкт - предикат - об'єкт".

Суб'єкти та об'єкти – це сутності, які можна візуалізувати вершинами деякого графа, а предикат – це бінарне відношення між сутностями, яке можна показати

дугою. Таким чином, всю онтологію, тобто безліч триплетів RDF, можна показати як діаграми орієнтованого графа.

RDF — надто примітивна мова, в якій немає типізації відносин, всі дуги однакові — це не дуже добре з погляду виразності при побудові онтології.

Мова RDFS (RDF Schema). Розширення мови RDF. У мові RDF всі можливості відображати відносини між поняттями вичерпуються іменованими предикатами, причому про самі предикати нічого конкретного сказати не можна. У мові RDFS з'явилися нові можливості. Можна зазначити, що один клас є підкласом іншого, одна властивість (предикат) — окремий випадок іншої властивості. Таким чином, RDFS має можливість специфікувати узагальнення як сутностей, так і відносин. Також для властивості можна вказувати область визначення та значення.

Мова OWL (Web Ontology Language). Ця мова включає всі функції RDFS. Крім цього, можна накладати ще дуже багато обмежень і на властивості (відносини), і на класи (сутності). Можна, наприклад, сказати, що ставлення має бути транзитивним, рефлексивним, симетричним тощо.

У мові OWL також є конструктори класів — можна створювати нові класи об'єднанням, перетином, доповненням старих. Можна задавати класи через обмеження властивостей. Мабуть, OWL — найбагатша мова для опису онтологій із спеціалізованих мов на сьогоднішній день.

Уніфікована мова моделювання UML (Unified Modeling Language). Для опису онтологій підходить уніфікована мова моделювання UML. Діаграма класів UML включає всі можливості RDF, RDFS і OWL, хоча деякі конструкції в UML виглядають більш заплутаними, а тому рідше використовуються.

У діаграмах класів UML є класи з атрибутами, є різні відносини між класами: асоціація, узагальнення, реалізація, залежність, агрегація, композиція. Узагальнення — те саме, що й підкласи з RDFS та OWL, реалізація, як і асоціація, — також є у RDFS та OWL. Агрегації/композиції (ставлення ціле-частина) в OWL і RDFS немає,

хоча це досить корисне ставлення. Також наявність такого вбудованого типу відносин як залежності в UML видається дуже корисною. Можна показати, що якийсь клас виражається через деякі інші, але на відміну від конструктора класів OWL, цей вислів не обов'язково наводити явно. І це набагато корисніше — суворий висновок залежного класу з деяких незалежних часто може бути дуже складним, тому нам важливіша інформація, що він у принципі можливий, ніж те, як саме цей висновок робиться. Тому ми вважаємо, що наявність відношення "залежність" в UML для онтологій набагато корисніша за конструктор класів у OWL. Таким чином, агрегація, композиція та залежність - відносини, які часто застосовні при описі онтологій, при цьому властиві вони тільки UML.

В даний час загальновизнано, що онтології в інформатиці повинні бути машиночитаними і мають бути спрямовані на досягнення певної мети. Для прикладу, метою є використання онтології для навчання. Онтології можуть бути корисними в освітньому процесі. Їх можна використовувати як візуальний матеріал поряд з презентаціями для лекцій.

Онтології дозволяють [35]:

1. Винести всі вивчені поняття в одну схему - так учням простіше зрозуміти, що взагалі було пройдено;
2. Показати ієрархію понять — які у сутностей, що розглядаються, є спеціалізації та узагальнення;
3. Відобразити, як одні математичні об'єкти можуть перетворюватися на інші, тобто, які існують алгоритми.

Застосування обчислювальної техніки для вирішення завдань, пов'язаних з семантичною обробкою інформації, почалося в 60-ті роки XX століття. У рамках проекту зі штучного інтелекту (ШІ), робилося кілька спроб створити інформаційні семантичні системи. Основним досягненням робіт у галузі семантичної обробки інформації були розробка математичного та категоріального апарату для подальшого

розвитку цього напрямку. Зі зростанням продуктивності обчислювальних машин та розвитку глобальних мереж цей напрямок у дослідженнях дозволить у майбутньому суттєво просунутися у напрямі інтелектуалізації різних інформаційних систем. На даному етапі розвитку людства, глобальна мережа інтернет є широке інформаційне поле, величезна база знань, що містить докладні відомості наукового, історичного, політичного та повсякденного характеру. Інтернет можна порівняти з величезним словником, що описує наш всесвіт, нашу планету і всі процеси, що супроводжують розвиток людської цивілізації, які відбувалися, відбуваються і можуть статися у майбутньому. Однак завдання користувачів вимагають більш систематизованого та настроюваного механізму інтеграції розподілених та різнорідних знань у цілісну картину предметної галузі. Слід зазначити, що оригінальна специфікація WWW розроблялася саме на вирішення завдання інтеграції наукових матеріалів.

Сучасний інтернет розроблявся як інформаційний простір, корисний як для комунікації людей, але й як простір, у якому ефективно сприяння можуть надавати й машини. Однак згодом релевантний пошук у мережі стало дедалі складніше здійснювати через експоненційне зростання обсягу розміщеної там інформації. Цей аспект ставить нові завдання перед розробниками пошукових машин та інтелектуальних інформаційних систем. Головним завданням є забезпечення релевантного пошуку інформації за семантичною ознакою. Поруч зарубіжних дослідників у галузі інформаційних технологій було розроблено інструментальні засоби та механізми, представлені в рамках єдиної концепції так званої "семантичної павутини".

Розглянемо сучасний стан робіт у галузі онтологій. Концепція "семантичної павутини" (Semantic Web) базується на таких основних компонентах: активне використання метаданих; метамову XML, онтологічному підході та моделі Resource Description Framework (RDF). Вони дозволяють здійснювати обмін даними та їх багаторазове використання. У рамках цього проекту найбільшого поширення набула

модель представлення метаданих Resource Description Framework (RDF). RDF надає засоби для запису триплетів, трійок даних: суб'єкт-предикат-об'єкт. Об'єкт і суб'єкт відповідають вузлам графа, а предикат або властивість спрямованій дузі графа. Дуга спрямована від суб'єкта до об'єкта. Кожен із елементів триплету називають RDF ресурсом та ідентифікують за допомогою URI ідентифікаторів. Платформа RDF активно використовується для представлення різних даних, зокрема RSS 3.0, агрегатори новин збирають інформацію у форматі RDF. Однак просте використання RDF не дозволяє реалізовувати завдання семантичного павутиння [36].

Найбільш загальний спосіб уявлення знань, у якому предметна область сприймається як сукупність об'єктів і зв'язуючих їх відносин, реалізований у мережевій моделі знань. Одним із головних досягнень консорціуму з всесвітнього павутиння (W3C) є розробка стандарту опису онтологій. Будь-яка онтологія передбачає виділення її трьох взаємопов'язаних компонентів: таксономії термінів, описів сенсу термінів, а також правил їх використання та обробки. Онтологія використовується для структурування інформації. Модель онтології може бути статичною чи динамічною. Для опису відносин в онтологіях використовується весь спектр математичного апарату, розробленого в галузі штучного інтелекту - обчислення предикатів, системи продукції, семантичні мережі, фрейми тощо. Таким чином, модний сьогодні термін "онтологія" виявився близьким за значенням до терміну "штучний інтелект", а термін "онтологічний інжиніринг" став синонімом терміна "інженерія знань". До основних завдань, розв'язуваних з допомогою онтології, ставляться створення та використання баз знань, організація семантичного пошуку, реалізація механізмів міркувань, уявлення сенсу метаданих про інформаційний ресурс та інших.

Для опису онтологій, доступних через веб, створені мови RDFS (RDF Schema - Схеми RDF) і OWL (Ontology Web Language - Мова Мережеских Онтологій). Як базові елементи дані мови використовують RDF ресурси. RDFS використовується

для запису словників, а OWL – онтологій. Мережеві онтології надають виразніші можливості порівняно з RDF словниками, наприклад логічні операції над класами та логічні обмеження властивостей.

На даний момент одним із перспективних напрямів є розробка алгоритмічних та програмних засобів побудови онтологій та роботи з ними. Незважаючи на наявність засобів і механізмів, орієнтованих спеціально на використання онтологій для широкого кола завдань, пов'язаних з інтелектуальною обробкою інформації, залишаються проблеми з труднощами проектування та побудови самих онтологій, автоматизацією побудови онтологій за масивами даних природною мовою та широкого застосування онтологій.

Проаналізуємо побудову онтологій [34]. Інформаційні системи різного призначення на основі онтологій на практиці показали свою ефективність, проте побудова онтології потребує експертних знань у досліджуваній предметній галузі. На даному етапі розвитку онтологічного інжинірингу процес розробки онтології займає суттєвий обсяг часу, оскільки ґрунтується на ручному створенні онтології експертом предметної галузі. На даний момент у світі існує кілька десятків систем для підтримки розробки онтологій. Такий технологічний процес на даному етапі розвитку ІТ не може вважатися ефективним, тому актуальним завданням є автоматизація процесу побудови онтології. На даний момент немає повноцінних інструментів автоматичної побудови предметних онтологій, хоча ведуться роботи в цьому напрямку. Для початкового вилучення інформації з різних джерел нині користуються підходи, засновані на автоматизації отримання знань з даних, текстів та веб-ресурсів (data-mining, text-mining, web-mining). Проте отримані дані зазвичай все ще вимагають їх обробки експертом предметної області. Будь-яка онтологія для її успішного використання повинна мати ряд властивостей, виділених Томом Грубером [37]:

- Прозорість - онтологія повинна ефективно передавати певне значення певного терміну;
- Зв'язність - онтологія має бути зв'язною, тобто, вона має дозволяти робити висновки, які узгоджуються з вихідними визначеннями понять. Принаймні, аксіоми, що визначаються, повинні бути логічно узгоджені між собою;
- Розширюваність - онтологія повинна бути розроблена з можливістю використання словника, що розділяється і поповнюється;
- Незалежність від синтаксису - концептуалізація має бути специфікована лише на рівні знання максимально незалежно від уявлення понять, лише на рівні символів;
- Мінімальний базис при високій виразності - онтологія повинна вводити мінімальний базовий набір понять, але їх має бути достатньо, щоб моделювати світ у необхідних цілях та створювати складні сцени.

У літературі зустрічається кілька підходів до представлення знань для завдань онтологічного інжинірингу: динамічні семантичні мережі (Dynamical Semantic Network), продукційні моделі, нейронні мережі та автомати, що самонавчаються. Однак усі ці підходи окремо не дають очікуваного ефекту. Останнім часом застосовуються комбіновані схеми, що поєднують переваги явного та непрямого уявлення знань. Ці підходи реалізовані в таких продуктах як інструмент візуального конструювання знань CAKE та агентна система Agent Builder компанії Reticular Systems, які використовують фреймо-продукційну модель.

Існує безліч пропозицій щодо методик розробки онтології. У межах таких методик зазвичай виділяються такі основні завдання [34].

1. Аналіз цілей створення та сфери застосування створюваної онтології;
2. Побудова онтології. Збір та фіксація знань про предметну область, що включають: визначення основних понять та їх взаємовідносин у обраній предметній галузі; створення точних несуперечливих визначень для кожного

основного поняття та відносини; визначення термінів, які пов'язані з основними поняттями та відносинами; узгодження перелічених компонентів онтології;

3. Кодування, що включає: розбиття сукупності основних термінів, що використовуються в онтології, на класи; вибір чи розробку спеціальної мови для представлення знань; формування концептуалізації у межах обраної мови уявлення знань.

Наведений алгоритм є загальним і залежно від постановки конкретних завдань може змінюватись.

Розглянемо динамічну модель онтології. Про необхідність привнесення динаміки до онтології говорить також досвід практичного застосування поширених статичних онтологій. Так, у [33] розглядаються кілька експериментів, що мають на меті підвищення точності пошуку з використанням семантики лінгвістичної онтології WordNet. Метою експериментів була спроба виконати пошук документів на основі не окремих слів, а значень WordNet. Ефективність використання векторів синсетів порівнювалася з ефективністю інформаційного пошуку на основі стандартного вектора слів. Оцінки ефективності інформаційного пошуку на основі показника середньої точності показали серйозне погіршення ефективності векторів, що включають синсети (від 6.2% до 42.3%). Основна причина такого погіршення ефективності полягає в тому, що процедура вирішення багатозначності для слова у запиті може вибрати одне значення, а для того ж слова у документі інше значення. Інша група експериментів щодо використання WordNet в інформаційному пошуку досліджувала можливість розширення запиту синонімами або іншими словами, пов'язаними зі словами запиту відносинами, описаними в WordNet. У таких експериментах немає необхідності вибору єдиного значення слова, що у разі помилки призводить до серйозного погіршення результатів пошуку. Максимальне

покращення, яке вдалося отримати - 0.7% середньої точності, що не є статистично значущою величиною.

Отже, статичний підхід до побудови онтологій не забезпечує сам собою істотного поліпшення результатів інформаційного пошуку. Понад те, виникає потреба у складній логіці інтелектуального агента, покликаної підвищити релевантність запитів. Але навіть використання подібної логіки не приносить відчутного результату, а часом, як бачимо, навіть призводить до зниження ефективності пошуку порівняно з широко використовуваними несемантичними методами.

Інший підхід до вирішення проблеми ефективного використання онтологій запропонований авторами мови семантичної розмітки SHOE (Simple HTML Ontology Extensions). SHOE є заснованою на онтологіях мовою представлення знань, що вбудовується в існуючі веб-сторінки. Він дозволяє привнести до розмітки сторінки логіку (наприклад, логічні оператори IF-THEN-ELSE), створювати аліази властивостей та термінів онтології, агрегувати кілька онтологій. Мова SHOE містить також механізм підтримки версій. Але, як і у випадку зі сторонньою програмною компонентою, що застосовувалася в ході експериментів з онтологією WordNet, мова багато в чому покладається на інтелектуальних агентів, пропонуючи лише гнучкішу і потужнішу розмітку і досить примітивну логіку, вбудовану безпосередньо в нього.

Найсуттєвішим мінусом вищеописаних підходів і те, що це зусилля для поліпшення результатів пошуку даних переважно чи повністю перекладаються на інтелектуальних агентів, клієнтські веб-додатки. При цьому сама онтологія не сприяє збільшенню релевантності даних, а лише надає структуровану (нехай навіть докладно) семантику конкретної предметної галузі.

Проаналізуємо модульну структуру онтології.

Ключову роль інструмента еволюціонування онтології грає ядро онтології. Ядро – високоінтелектуальна система. Воно містить у собі механізми роботи з

семантикою, що містяться у статичній онтології; здійснення еволюційних змін в онтології, а також повністю бере на себе функції виконання пошукового запиту даних та взаємодії всіх учасників інформаційного процесу. У процесі пошуку даних ядро може активно взаємодіяти з різними онтологіями, доступними в інтернеті, навіть якщо вони є статичними.

Розглянемо механізм пошукового запиту із використанням модульної динамічної еволюційної онтології. Ядро онтології отримує запит природною мовою від інтелектуального агента, а той, своєю чергою, від користувача інформаційної системи. Потім, ядро перетворює запит на деяке внутрішнє уявлення. На цьому етапі відбувається активний обмін даними з онтологією та, якщо потрібно, з іншими онтологіями, представленими в інтернеті. Таким чином, у запиті виділяються найбільш значущі ключові конструкції, за якими можна проводити ефективний пошук серед семантично розмічених сторінок інформаційної системи, що існує в рамках інтернету. Система, отримавши безліч документів від ресурсів мережі, намагається визначити їхню релевантність та шукає конфлікти, багатозначності трактування шуканих ключових слів та словосполучень у знайдених документах. У разі виявлення подібних ситуацій, ядро готує запит на уточнення та надсилає його інтелектуальному агенту, а агент - користувачеві. Далі агент надсилає відповідь, що дозволяє уточнити специфіку та семантику запиту. Потім ядро заново або частково заново посилає запит у мережу, а також фільтрує документи, знайдені на попередньому етапі пошуку. Коли неясних термінів не залишається, ядро повертає набір релевантних документів інтелектуальному агенту, а той у свою чергу - користувачеві.

На кожному етапі запиту ядро здійснює процес зміни онтології, у результаті здійснюється оптимізація її внутрішньої структури, що сприяє підвищенню релевантності семантичного пошуку. Використання цього підходу дозволить точніше здійснювати пошук даних за природним запитом користувача, згодом.

Істотним плюсом модульного підходу є можливість роботи єдиного ядра з кількома еволюційними онтологіями водночас і зв'язування їхніх стосунків лише у запитах, коли це буде потрібно. Так само вдається уникнути необхідності повністю підтримувати кілька окремих версій "великих" онтологій, для несуттєво відмінних клієнтських систем. У разі використання модульного підходу ядро буде єдиним. Крім того, впровадження інтелектуального ядра, здатного до процесу еволюції, дозволить серйозно підвищити ефективність структури зберігання даних предметної області всередині статичної онтології.

Розробники інтелектуального агента, у свою чергу, замість того щоб реалізовувати логіку підвищення ефективності онтологічного пошуку як такого, зможуть зосередитися на більш продуктивній роботі. Наприклад, реалізації обробки окремих випадків і нестандартних правил предметної області. Крім того, інтелектуальний агент, виконавши попередню обробку складного запиту, сам побічно допомагатиме ядру онтології ефективніше обробити ці дані та змінити структуру онтології [38].

Розглянемо докладніше класи завдань, які вирішуються ядром онтології. Ядро онтології є складною системою, на яку покладається широкий спектр різного роду функцій:

- *Запуск та управління процесом пошуку даних запиту.* Має на увазі здійснення механізму поетапної обробки запиту як складного процесу взаємодії семантичних систем. Навіть один досить простий запит клієнта запускає всередині ядра багаторівневий процес, що стосується практично всіх модулів ядра і більшу частину його функціональних можливостей. Зрештою, клієнт отримує релевантні запиту документи, а система еволюціонує, модифікуючи онтологію з результатів пошуку;
- *Управління покроковим пошуком даних.* Має на увазі, що спілкування користувача з системою може відбуватися не тільки в режимі, коли користувач

вводить текст запиту і, відправивши його, відразу отримує відповідь. У складних ситуаціях ядро може знадобитися уточнення запиту. Виходить певна подібність діалогу експертної системи. Такий підхід дозволяє зняти велику проблему - коли виникають ситуації неоднозначного трактування семантики і система не може ефективно вирішити їх самостійно. Кінцевий користувач виграє в тому, що витративши лише трохи часу на уточнення свого запиту, відразу ж отримує релевантні дані;

- *Виконання запитів даних до онтології, джерел в інтернеті, інших онтологій.* Документи можуть бути отримані з мережі на основі семантики предметної галузі, що зберігається як і у внутрішній онтології, так і у зовнішніх джерелах;
- *Перетворення вхідних та проміжних даних, фільтрація одержуваних документів.* Дані запиту природною мовою можуть бути перетворені на векторний або булівський запит, або в іншу структуру, яка дозволить отримати більш точний набір документів з мережі. Також ядро в змозі самостійно відфільтрувати проміжні дані, які явно не відповідають запиту або сильно відрізняються від решти вибірки;
- *Підтримка зв'язків з іншими статичними та динамічними онтологіями.* Якщо семантика запиту неясна чи частина запиту відповідає предметній області тезауруса, то логічно уточнити її семантику з інших зовнішніх джерел. При цьому вважатимемо, що в ядро вшиті посилання на певний набір існуючих у мережі онтологій та правил у якому разі і до якої із зовнішніх онтологій звертатися;
- *Еволюція структури онтології у відповідність до потреб запитів клієнтських систем.* Ядро проводить постійний аналіз запитів документів та семантичних, мовних конструкцій. В результаті може доповнитися список зв'язків між термінами внутрішньої онтології, виявлення нових синонімів, винятків, тощо;

- *Вирішення конфліктів і помилок у процесі еволюціонування.* Ядро постійно аналізує стан онтології, має у своєму сховищі приклади винятків, помилкових зв'язків та класифікацій. Компонент самонавчається та намагається виявити наявність подібних проблем у внутрішній онтології та виправляє їх, або формує звіт для експерта онтологій у складних та неочевидних ситуаціях;
- *Обмеження надмірного зростання онтології.* У процесі еволюціонування онтологія зростає, збільшується кількість термінів та зв'язків, що зберігаються в ній. Але онтологія не повинна вбирати в себе дані, що не торкаються її предметної галузі, а натомість містити лише вивірену семантику свого інформаційного поля. Тому ядро бере частину даних із зовнішніх баз знань і не допускає їх дублювання у своїй онтології;
- *Контроль версій онтології в процесі еволюціонування.* Ядро створює безліч версій (або ревізій) онтології в процесі її розвитку, таким чином, експерт, який обслуговує онтологію, завжди зможе втрутитися і повернути онтологію на кілька кроків тому, якщо щось піде не так. Скажімо, якщо буде виявлено безліч семантичних помилок всередині онтології.

Однак на шляху впровадження онтологічного підходу є багато проблем. Одна з найважливіших відсутність загального стандарту на опис та подання онтологій, хоча рекомендації W3C на мову OWL розроблені вже давно. Відсутність цих стандартів гальмує розробку універсальних інструментів для створення та підтримки онтологій, та бібліотек готових рішень. Тут виникають проблеми інтеграції гетерогенних онтологій. Іншим важливим фактором є те, що існуючі інструменти "ручної" розробки онтологій призначені скоріше для програмістів, а не на експертів предметних галузей.

Онтологічні мережі - це сукупність онтологій які пов'язані відношеннями відображення. У такій мережі об'єкти представлені як поняття, які пов'язані між собою за допомогою відношень. Кожен об'єкт може мати багато різних відношень з

іншими об'єктами, що дозволяє моделювати складні взаємозв'язки між ними.

Онтологічні мережі дозволяють представляти знання про об'єкти та їх взаємодії у вигляді концептуальної структури, що робить їх доступним для комп'ютерного аналізу. Це дозволяє використовувати знання для розв'язання різноманітних задач, таких як пошук інформації, класифікація об'єктів, прогнозування подій та інші.

Онтологічні мережі складаються з трьох основних елементів:

1. Поняття або концепти - це сутності, які потрібно описати в онтологічній мережі. Кожне поняття має ім'я та визначення, яке описує його характеристики та відношення з іншими поняттями.
2. Відношення - це зв'язки між поняттями в онтологічній мережі. Відношення можуть бути різними видами, наприклад, частини, місця, причини, наслідки, взаємодії тощо. Відношення можуть бути орієнтованими або не орієнтованими та мати різні ступені семантичної близькості.
3. Аксіоми - це правила, які визначають, як взаємодіяти з поняттями та відношеннями в онтологічній мережі. Аксіоми використовуються для формалізації знань та використовуються в алгоритмах, що розв'язують різні задачі на основі цієї мережі.

Онтологічні мережі можуть бути створені вручну, але це дуже трудомістка та складна робота. Тому для автоматизації процесу створення онтологій використовуються різні алгоритми та інструменти, які допомагають створювати онтології з використанням даних.

Онтологічні мережі використовуються в багатьох галузях, наприклад:

1. Пошук інформації - знання, що зберігається в онтологічній мережі, може бути використане для пошуку інформації за ключовими словами та іншими параметрами. Це допомагає забезпечити більш точний та релевантний результат.

2. Класифікація - онтологічні мережі можуть використовуватися для класифікації об'єктів та понять за певними критеріями. Наприклад, у медицині онтологічні мережі використовуються для класифікації захворювань за їх симптомами та лікуванням.
3. Розумні системи - онтологічні мережі можуть бути використані для створення розумних систем, які можуть розуміти та аналізувати знання. Наприклад, розумні системи можуть використовувати онтології для аналізу текстової інформації та виявлення залежностей між об'єктами та поняттями.

Використання мереж онтологій у процесі прийняття рішень відноситься до загальної проблематики відображення або ж інтеграції онтологій [39]. Причиною використання мереж онтологій є те, що значення концептів залежать від контексту, від сфери використання, від досвіду експерта, що створював онтологію. При цьому онтології можуть створюватися на різних мовах. При вирішенні практичної задачі прийняття рішення важливо використати концепти та відношення, які найточніше відповідають ситуації прийняття рішення.

Для інтеграції онтологій існують такі підходи:

1. *Лінгвістичний*. В різних онтологіях порівнюють значення термінів відповідно до їх словникового визначення. Це дуже зручно, коли відомо назву концепта в одній онтології та треба підібрати подібні концепти з інших онтологій. Для цього використовують метрики семантичної відстані між термінами.
2. *На основі порівняння сфер призначення двох онтологій*. В основі визначення онтологій лежить множина моделей застосування знань (intended models). Що також відображається у питаннях щодо компетенції онтологій. (можна також використати порівняння функціональних вимог або use-cases для систем в яких використовуються ці онтології. Такий підхід пов'язано з проблематикою повторного використання онтологій. [40, 41])

3. *Структурна відповідність (порівняння структур двох онтологій).* Існує думка, що значення концепту визначається не стільки як ми його називаємо, а сукупністю відношень з іншими концептами. У цьому підході ми визначаємо подібність концептів у двох онтологіях за наявністю (або подібністю) тих самих відношень, які пов'язують концепти у обох онтологіях з іншими, відомими (і також подібними) концептами у цих онтологіях.
4. *Порівняння подібності онтологічних патернів або ж груп патернів.* Доменні онтології сьогодні створюються шляхом опису патернів (груп патернів, мов патернів) [42]. Таким чином, при визначенні подібності двох онтологій можна також порівнювати патерни, або ж групи взаємопов'язаних патернів. Такий підхід, зокрема відображено у [43].

Онтологічні мережі - це потужний інструмент, який допомагає структурувати та формалізувати знання в різних галузях. Використання онтологічних мереж допомагає покращити ефективність та точність обробки даних, забезпечує більш точний та релевантний пошук, а також дозволяє розумніше аналізувати та використовувати знання. Отже, онтологічні мережі є важливим інструментом у розвитку інформаційних технологій та забезпеченні прогресу в різних галузях. Із зростанням обсягу даних, які генеруються в різних галузях, використання онтологій стає ще більш актуальним.

1.3. Узгодження онтологій та операції з онтологіями

Інтелектуальний агент створює та використовує концептуалізацію предметної області, керуючись своїми цілями та намірами. Кожен агент має свою власну систему знань (онтологію), яку він розбудовує на основі своєї діяльності, цілей та спілкування з іншими агентами. Оскільки середовище, в якому діє агент, та його цілі постійно змінюються, стан його знань також змінюється завдяки процесам навчання. Таким чином, концептуалізація середовища є динамічною та унікальною для

кожного агента. Системи знань та концептуалізації частково узгоджуються в процесі спілкування агентів та використання фрагментів онтологій інших агентів (навчання в інших агентів) та вбудовуються у систему знань агента. Практичні задачі, які розв'язує агент, тестують та оновлюють системи знань.

Такий підхід, на відміну від використання фіксованих, розроблених поза агентом онтологій дає змогу відображати різні, навіть несумісні, перспективи у розумінні та концептуалізації проблем, з метою побудови гіпотез та методів їх перевірки.

З філософської точки зору такий підхід ґрунтується на ідеях Фалібілізму (Fallibilism) [44], епістемологічного положення, відповідно до якого не може бути абсолютної впевненості і ніколи не можна виключити всі альтернативні шляхи у вирішенні задач. Обґрунтування або пояснення, що ставить метою дати остаточне обґрунтування, та остаточно вирішити проблему є невірним і не може привести до успіху. Тому залишається тільки завжди перевіряти переконання, думки чи гіпотези на помилки і, за можливістю, замінити їх на кращі.

Подібний методологічний підхід, наприклад, відображений і в онтології верхнього рівня General formal ontology (GFO) [45], який бачить онтологію як систему часткового порядку логічних теорій, деякі з яких можуть бути у логічному протиріччі з іншими теоріями, які не розташовані на тому ж шляху впорядкування.

Основні визначення:

1. *Система, що базується на знаннях (Knowledge based system – expert system).*
Комп'ютерна програма яка використовує базу знань та засіб логічного виведення для вирішення задач які зазвичай вимагають використання експертних знань.
2. *База знань.* Це технологія що використовується для збереження складних структурованих і неструктурованих даних в комп'ютерній системі.

3. *Система знань (Knowledge system)*. Це структура та процес який створює та подає контент, компоненти, класи знань які: є специфічними та релевантні у домені використання що визначено користувачами; з логічними зв'язками що з'єднують контент знань з їх значимістю (корисністю); мають множину ітеративних процесів що підтримують еволюційний розвиток, оновлення, адаптацію та створення нових знань; оцінюються за критеріями релевантності, надійності та якості.

Автор Olive [46] визначає поняття інформаційної бази (information base) як подання сутностей та відношень предметної області та їх класифікацію за типами сутностей та відношень. Віднесення конкретної сутності або відношення до визначеного типу називається фактом.

Розв'язання проблеми узгодження онтологій потрібне для забезпечення семантичної інтероперабельності між інтелектуальними агентами. Це означає здатність комп'ютерних систем обмінюватися даними, які мають однакову інтерпретацію їх змісту [47]. Для досягнення такої інтероперабельності потрібна не лише семантична, але й синтаксична узгодженість, що передбачає використання узгоджених форматів (мов) подання інформації.

У статті [48] зазначено класифікацію причин семантичної неузгодженості онтологій. Згідно з цією класифікацією всі причини можна розділити на дві групи в залежності від місця їх виникнення - обумовлені різницями у мовах подання та різними підходами до моделювання.

У першому випадку причини семантичної неузгодженості можуть бути наступними:

- Різне синтаксичне подання.
- Різні припущення на мета-рівні.
- Різні подання логіки.
- Різна семантика мовних примітивів.

- Різна експресивність мовних засобів.

Причини, що виникають на рівні моделей у свою чергу можуть бути викликані різними підходами до моделювання, різницями у концептуалізаціях та розумінні понять. Так, різниці концептуалізації можуть містити:

- Різні концепти та відношення в онтологіях.
- Однакові концепти, але різний зміст.
- Покривати іншу частину області застосування або покривати область неповно.
- Різну гранулярність (використання більших або менших елементів).

У випадку різного розуміння виникають:

- Різниці у визначеннях понять.
- В термінології – за рахунок використання синонімів та омонімів.
- Різниці в одиницях вимірювання (напр. метри та фути).
- Різниці через використання різних модельних парадигм.

У тій же роботі [48] наведено такі форми операцій з узгодження онтологій:

- *Комбінування (Combining)*. Використання двох або більше різних онтологій для виконання завдання в якому важливим є залежність між ними.
- *Інтеграція, злиття (Merging, integration)*. Створення нової онтології на основі існуючих, що частково перекриваються.
- *Узгодження (Aligning)*. Створення взаємної узгодженості та цілісності між онтологіями.
- *Відображення (Mapping)*. Поєднання подібних (за визначеними метриками) концептів та відношень з різних джерел відношенням еквівалентності.
- *Артикуляція (Articulation)*. Відношення між елементами двох узгоджуваних онтологій, специфікація операцій узгодження.
- *Переклад (Translation)*. Зміна форми подання онтології з збереженням семантики.

- *Трансформація (Transforming)*. Зміна семантичного змісту (можливо, і форми) онтології для її використання для інших, ніж попередньо передбачалося задач.
- *Версія (Version)*. Результат зміни онтології, який існує разом з оригіналом.
- *Версіонування (Versioning)*. Метод для підтримки версійності онтологій.

Узгодження онтологій передбачає можливість проведення операцій та перетворень як з цілими онтологіями, так і з їх частинами в процесі узгодження. Розглянемо які операції було запропоновано.

1. Відношення відповідності.

У роботі [49] дані основні формальні визначення щодо відношень відповідності між онтологіями у мережі онтологій. Так, якщо є дві онтології O та O' , з відповідними мовами подання $Ent(O)$, $Ent(O')$ та множина відношень відповідності R , відповідність між їх елементами визначено як трійку (e, e', r) таку що $e \in Ent(O)$, $e' \in Ent(O')$. Іншими словами, відповідність, це твердження, що між концептами e та e' існує відношення відповідності r . Відповідність між онтологіями у цілому визначено як множину усіх відповідностей між елементами [50].

2. Правила відповідності (articulation rules).

Частиною відношення відповідності є правило відповідності $ArtRu$, яке встановлює яким чином з елементу (або фрагменту) онтології O отримати елемент онтології O' . У найпростішому випадку це просто відображення між елементами двох онтологій: $ArtRu: e \rightarrow e'$. У загальному випадку правило відповідності це відображення підмножини концептів, відношень та функцій інтерпретації вихідної онтології в підмножину концептів, відношень та функцій інтерпретації результуючої онтології:

$$ArtRu: (Sub(T), Sub(R), Sub(F)) \rightarrow (Sub(T'), Sub(R'), Sub(F')),$$

де $Sub(T)$, $Sub(R)$, $Sub(F)$ це підмножини концептів, відношень та функцій

інтерпретації вихідної онтології, а $Sub(T')$, $Sub(R')$, $Sub(F')$ – підмножини концептів, відношень та функцій інтерпретації для результуючої онтології. Правила відповідності використовують для безпосереднього перетворення та інтерпретації елементів однієї онтології в елементи іншої онтології.

3. Сигнатури онтологій та редукти.

Онтологію можна подати як двійку: $O = (S, A)$, де S - це сигнатура онтології (містить визначення усіх її елементів), а A - набір аксіом. У випадку якщо ми розглядаємо концептуальну модель M яка побудована на основі онтології $O_M \subseteq O$, у цьому випадку O_M є редуктом онтології O по $\sigma_M : O_M = Red(O, \sigma_M)$.

4. Повторне використання онтологій.

Деякі автори, зокрема [40] розглядають задачу узгодження онтологій як задачу їх повторного використання. При цьому автор з [40] використовує положення, що в основі онтологічної концептуалізації лежить множина потрібних моделей (intended models) яке вперше було сформульовано у роботах Ніколи Гуаріно [51]. При цьому множина потрібних моделей визначена як вибірка з можливих моделей, відповідно до намірів та потреб розробників онтології [40].

5. Виокремлення онтологій (інші підходи).

Операція виокремлення онтологій розглянута у багатьох роботах. Зокрема, у [52] розглянуто виокремлення онтологій на основі вимог до системи. Детальний аналіз операції виокремлення онтології подано у [53]. У [54] розглянуто як коректно поділяти онтологію на частини у процесі декомпозиції. Головною вимогою щодо результату виконання операції виокремлення є збереження семантичної повноти у результаті [52] – результуюча онтологія мусить бути цілісною, так що її можна буде використовувати незалежно від інших.

У [53] розглянуто алгоритм виокремлення онтології, що складається з кроків

вибору концептів з первинної онтології, вибору атрибутів, та трансформацію відношень. Також ставиться вимога, щодо зв'язності графу, що відображає структуру онтології.

Висновки до розділу 1

Для побудови моделей предметних областей у системах підтримки прийняття рішень використовуються онтології, які дозволяють формалізувати знання про предметну область та її елементи. Використання онтологій у системах підтримки прийняття рішень дозволяє зменшити розбіжності у розумінні термінів та концепцій, що використовуються у предметній області. Також це дозволяє скоротити час на пошук і формалізацію даних. Проте, проектування онтологій пов'язане зі складнощами, які розв'язуються з урахуванням особливостей предметної області. Результати аналізу показують, що інтерпретація змісту концептів онтології залежить від конкретної ситуації прийняття рішення та повинна обиратися динамічно після ідентифікації ситуації. Застосування онтологічних мереж надає можливість вибору варіанту інтерпретації концепту залежно від ситуації, що є важливим фактором у підвищенні ефективності системи підтримки прийняття рішень.

РОЗДІЛ 2. Застосування онтологій для пошуку, опрацювання та аналізу інформації у процесі прийняття рішень

2.1. Застосування онтологій для систем підтримки прийняття рішень

Прийняття рішень є важливою складовою управління будь-якою організацією, і для підтримки цього процесу використовуються різні методи, включаючи багатокритеріальне прийняття рішень, особливо в тих випадках, коли рішення стосується кількох зацікавлених сторін і критеріїв. Ці методи, що базуються на оцінці кожної альтернативи за набором критеріїв, зазвичай мають суб'єктивний характер. Проте, незалежно від цього, важливо, щоб особи, які приймають рішення, мали максимально можливі знання про альтернативи, включаючи розуміння всіх наслідків кожної альтернативи та їх вплив на діяльність організації. Це вимагає глибокого розуміння галузі, в якій приймається рішення. Для досягнення такого розуміння використовують організаційну онтологію, яка забезпечує створення загального словникового запасу для дослідників і фахівців, що працюють у даній галузі та потребують обміну інформацією. У різних галузях, включаючи штучний інтелект, використовують онтології. Мета штучного інтелекту полягає у тому, щоб зробити машини достатньо складними для інтегрування змісту інформації. Для досягнення цієї мети дуже важливим є крок організації знань, який породив інженерію знань та покладається на онтології як засіб представлення та організації знань. Прийняття рішень вимагає виявлення наслідків для кожної альтернативи, включаючи непрямі наслідки. Один із способів визначення наслідків - вивчення всіх бізнес-процесів, на які вплине кожна альтернатива, і детальний аналіз цих процесів для визначення повного ступеня їх впливу. Для ефективного аналізу потрібно проаналізувати предметну область, для чого необхідна структура знань, яка полегшує цей аналіз.

Стрімкий розвиток комп'ютерних технологій обумовлює все більший останніми роками інтерес до інтелектуальної обробки великих обсягів інформації. У

той же час найчастіше фахівці перевантажені обсягом даних, що швидко зростає: масиви інформації стало складніше обробляти, складніше шукати потрібне, зросла потреба в оптимізації пошуку необхідного фактичного матеріалу. У зв'язку з цим дедалі більшої популярності набуває створення впорядкованих інформаційних баз у багатьох сферах виробничої діяльності, при цьому збільшується і значущість онтологій в обміні та структуруванні знань. Актуальність створення онтологій у геології обумовлена можливостями, що відкриваються при застосуванні сучасних цифрових інструментів, таких як машинне навчання, комп'ютерний зір, розробка баз знань, обробка природної мови та інше, для вирішення прикладних завдань. Онтологічне моделювання має велике значення, оскільки з огляду на різноманітність існуючих класифікацій та нестачі чіткої структури взаємозв'язків між поняттями існують певні складнощі із систематизацією знань. У зв'язку з цим ключовим аспектом створення онтологій є не просто збирання та аналіз всього наявного геологічного матеріалу, а впорядкування інформації відповідно до вирішуваного завдання.

Онтологія як термін використовується у філософії вже близько 600 років і означає філософську науку про суще і про буття. Однак у контексті сучасного світу вона означає модель опису будь-якої системи, що складається з набору типів, властивостей та взаємозв'язків.

Основоположником застосування поняття онтології в інженерії є американський вчений Т. Грубер, який розглянув різні аспекти взаємовідносин інтелектуальних систем між собою та людиною. Грубер, зокрема, охарактеризував онтологію як опис (специфікацію) концептів та взаємозв'язків (концептуалізацію) [37]. Згодом було сформульовано безліч різних визначень онтології. Одне з найпопулярніших дав італійський програміст Н. Гуаріно: "Онтологія - це формальна теорія, що обмежує можливі концептуалізації світу" [51]. Онтологія - це специфікація предметної області або формальне її уявлення, яке включає словник

показників на терміни предметної області та логічні висловлювання, які описують, що ці терміни означають, як співвідносяться один з одним і як вони можуть або не можуть бути пов'язані між собою. Іншими словами, онтології потрібні для того, щоб описувати основні положення в предметній галузі, їх властивості та взаємовідносини між ними, забезпечуючи обмін та систематизацію знань.

Найпростішим і дуже поширеним прикладом онтології є ієрархічна система класифікації живих організмів, запропонована К. Ліннеєм, в основі якої лежать категорії роду та виду. Ще один популярний приклад онтології – періодична система хімічних елементів, запропонована Д. І. Менделєєвим. Онтологічні моделі тією чи іншою мірою присутні у всіх галузях знань, але ми не завжди знаємо про їхнє застосування. Онтології визначають загальну термінологію для вчених, які спільно використовують інформацію в предметній галузі. Цілі створення онтологій можуть бути різними. Т. Грубер розглядав онтології як спосіб взаємодії інтелектуальних систем як між собою, так і з людиною. Крім того, онтології створюються для забезпечення можливості повторного використання знань у предметній галузі. Якщо цікава онтологічна модель вже була описана, нехай і для інших цілей, її повторне використання можливе і в інших галузях знань. Перевагою онтологій є те, що їх можна легко редагувати та змінювати.

Зазвичай, створення онтологічної моделі сама собою є метою. Розробка онтології зводиться до визначення набору даних та їх структури для використання при вирішенні прикладних завдань, зокрема при створенні програмних продуктів. Т. Грубер вважав, що онтологія включає п'ять складових частин: класи (поняття, концепти, сутності, категорії), відносини, функції, аксіоми, екземпляри (приклади). Класи використовуються в широкому сенсі, ними може бути будь-яка сутність, про яку може бути дано певну інформацію. Як правило, класи організовані у таксономії. Відносини показують тип взаємодії між класами цієї області. Функції - це особливий випадок відносин, у якому обраний елемент безпосередньо залежить від усіх

попередніх елементів. Аксиоми використовуються для констатації висловлювань, які завжди будуть істинними. Вони можуть бути включені в онтологію в різних цілях, наприклад визначення загальних обмежень на значення атрибутів, для перевірки коректності даних, представлених в онтології, або для виведення нової інформації. Примірники – це конкретні елементи класів.

Щоб дозволити інтелектуальним системам обмінюватися один з одним знаннями, необхідно зробити їх опис, зрозумілий як системі, так і людині, тому Т. Грубер запропонував описувати знання двома способами [37]:

- У заздальгідь передбаченій, канонічній формі: формальна мова з чітко визначеним синтаксисом та семантикою. Під семантикою розуміється наділення інформації сенсом для її обробки. Зокрема, цей термін означає, що програми мають отримати можливості обробки даних з урахуванням змісту та зв'язку між об'єктами;
- У формі онтології, що складається з безлічі термінів (класів, функцій, відносин, тощо), пов'язаних між собою визначеннями.

Сьогодні мови, у яких описуються онтології, можуть поєднувати ці форми. Виходячи з цього зараз під онтологією можна розуміти опис декларативних знань, зроблений формальною мовою що має ієрархію об'єктів.

Для початку розробки онтології слід визначити її область та масштаб. Для цього Н. Ной та Д. Макгіннес [55] пропонують відповісти на кілька основних питань, серед яких такі:

1. Яку область охоплюватиме онтологія;
2. Навіщо ми збираємося її використовувати;
3. На які типи питань має давати відповіді інформація в онтології;
4. Хто використовуватиме і підтримуватиме створену онтологічну модель.

Питання можуть змінюватися у міру створення онтологічної моделі, але у будь-який момент вони допомагають обмежити масштаб цієї моделі.

Для створення та оцінки онтологій необхідні об'єктивні критерії. Т. Грубер запропонував ряд принципів для конструювання онтологій, метою яких є обмін знаннями та взаємодія між програмами [37]:

1. Ясність - онтологія має ефективно та об'єктивно передавати значення кожного терміна;
2. Узгодженість - онтологія має бути послідовною, тобто має підтримувати умовиводи, що узгоджуються з визначеннями;
3. Розширюваність - онтологія має бути розроблена таким чином, щоб можна було, ґрунтуючись на наявному словнику, визначати нові терміни, не переглядаючи та не виправляючи вже існуючі терміни;
4. Мінімальна похибка кодування (мінімум впливу кодування) - вибір способу реалізації онтології не повинен бути заснований на міркуваннях зручності у конкретному випадку, тому що може виникнути проблема обміну знаннями з іншими системами;
5. Мінімальне онтологічне зобов'язання - для онтології має бути набраний мінімальний обсяг словника, достатній для опису знань, які передбачається спільно використовувати.

Питання оцінки якості онтологій є відкритим, оскільки критерії оцінки носять суб'єктивний характер. Безсумнівно, онтологія має бути представницькою та добре візуалізованою. І тому можна назвати ряд параметрів, що описують якість цього представлення.

1. Поняття одного рівня ієрархії пов'язуються з батьківським концептом одним і тим самим типом відносини (наприклад, "клас - підклас" або "частина-ціле");
2. В онтології має бути приблизно рівна глибина гілок;
3. Онтологія має бути симетричною;
4. Не повинно бути перехресних посилань;

5. Глибина гілок і максимальна кількість класів одного рівня ієрархії не повинна бути більшою за число Інґве-Міллера (7 ± 2).

Прийняття рішень включає вибір між різними альтернативами. Люди, які приймають рішення, зазвичай аналізують цілі, цінності та критерії, які є важливими для оцінки кожної альтернативи. Зазвичай існує кілька суперечливих критеріїв, які потрібно враховувати при прийнятті рішення. Для ефективного вибору необхідно ретельно зважити кожен критерій та розуміти наслідки кожної альтернативи. Хоча інтуїція може допомогти в більш простих випадках, для важливих рішень на рівні організації важливо структурувати проблему та забезпечити чітку оцінку кожної альтернативи з точки зору критеріїв.

Дослідники зосереджуються на розробці інструментів щоб допомогти людям приймати рішення. Існують різні методи та інструменти, наприклад, багатокритеріальне прийняття рішень і ціннісно-орієнтоване мислення. Процес прийняття рішень складається з чотирьох фаз [56]:

1. Фаза інтелекту, на якій визначається проблема або ситуація, що вимагає прийняття рішення.
2. Фаза проектування, на якій аналізуються можливі варіанти дій.
3. Фаза вибору, на якій обирається курс дій.
4. Фаза впровадження, на якій розроблений план дій виконується.

Інші моделі прийняття рішень включають наступні етапи для вирішення проблеми:

1. Конкретизувати цілі, цінності або завдання та відсортувати їх за важливістю.
2. Перерахувати всі можливі стратегії для досягнення цих цілей, цінностей або завдань.
3. Оцінити наслідки кожної стратегії.
4. Порівняти наслідки кожної стратегії з цілями, цінностями або завданнями.
5. Вибрати стратегію, яка найкраще відповідає цілям, цінностям або завданням.

Загалом, прийняття рішень є складним процесом, але може бути спрощеним за допомогою методів та інструментів, таких як багатокритеріальне прийняття рішень та ціннісно-орієнтоване мислення. Цей процес може бути розділений на чотири фази. Інші моделі раціонального прийняття рішень додатково вимагають уточнення цілей та цінностей, переліку можливих стратегій, оцінювання наслідків та порівняння їх з цілями, цінностями та завданнями, щоб обрати найкращий варіант. Ці методи допомагають звернути увагу на деталі та зробити кращу оцінку можливих наслідків.

Прийняття рішень можна трактувати як процес вибору однієї з можливих альтернатив, або як складний процес, що включає кілька етапів, таких як визначення проблеми, пошук альтернатив, вибір оптимальної альтернативи, прийняття рішення та оцінка його наслідків. При цьому можуть виникати складнощі і багатозначність ситуації, а також конфліктні цілі.

У багатьох дослідженнях передбачається, що людина, яка приймає рішення, є [57]:

1. Повністю поінформована.
2. Нескінченно чутлива.
3. Раціональна.
4. Здатна впорядковувати ситуації за певним критерієм, який має бути максимальним.

Нас цікавить, як люди насправді приймають рішення. Фактично, людина не має повного розуміння свого контексту та можливих кінцевих станів, які можуть виникнути в результаті її дій. Тому властивості економічної людини (1) і (2) не є реалістичними. Однак ми припускаємо, що властивості (3) і (4) все ще зберігаються. Щодо (3), важливо розуміти, що бути раціональним не означає бути емоційно нечутливим, але здатним аналізувати помилки, що базуються на емоціях і цінностях. Нарешті, ми вважаємо, що ситуації з різними можливими результатами можуть бути

впорядковані залежно від значень, які приймаюча рішення особа надає їм відповідно до своїх переконань.

Прийняття рішень є процесом обдуманого вибору з доступних варіантів на основі логіки. Коли такий вибір базується на ключових факторах, які допомагають досягнути успіху конкретної стратегії, то це можна вважати прийняттям стратегічного рішення. Важливо, щоб стратегічні рішення були прийняті, оскільки це дає можливість планувати дії відповідно до стратегій організації.

Сьогодні методи прийняття рішень використовуються в кількох сферах, таких як управління бізнесом та розробка програмного забезпечення. Існують різні інструменти, що допомагають приймати рішення, такі як системи підтримки прийняття рішень та стратегічні інформаційні панелі для перегляду стратегічних показників. Використання таких інструментів може сприяти більш обдуманому та ефективному процесу прийняття рішень.

Метод забезпечення загальної базової мови, який об'єднує концепції для управління рішеннями та обробки стратегічних індикаторів полягає в створенні онтології, що дозволяє структурувати знання для полегшення їх розуміння та комунікації. Онтологія може бути використана як основа для створення інструментів, що допомагають приймати стратегічні рішення. Крім того, ця онтологія може бути використана в майбутньому для забезпечення можливостей міркування, таких як пропозиції на вимогу.

У прийнятті рішень часто виникають труднощі через багатозначність та/або недостатнє знання проблемної ситуації та наявних ресурсів, слабку структуру завдання та багатокритеріальний вибір. Консультант-аналітик формалізує прикладну задачу, вибирає процедуру вирішення, організовує роботу особи, що приймає рішення, і експертів. Впровадження систем підтримки прийняття рішень допомагає знизити рівень суб'єктивності шляхом вирішення проблем недостатності та багатозначності вихідної інформації.

Онтології є дуже важливими у багатьох галузях, зокрема в штучному інтелекті. Їх розроблення повинно враховувати конкретні проблеми, які вони мають вирішувати. Наприклад, якщо ми створюємо онтологію для обслуговування комп'ютера, вона повинна бути здатна відповідати на будь-які питання, пов'язані з діагностикою помилок комп'ютера на основі даних, які надає користувач.

Загальне визначення онтології - це “явна та формальна специфікація спільної концептуалізації”. Елементи цього визначення слід розуміти так:

- Концептуалізація: абстракція явища, отримана шляхом визначення понять, відповідних цьому явищу;
- Формальний: вказує на те, що онтології інтерпретуються машиною;
- Явна специфікація: це означає, що поняття онтології та обмеження, пов'язані з їх використанням, визначені декларативно;
- Спільний: стосується того факту, що онтологія фіксує консенсусне знання.

Розглянемо загальний погляд на основні типи онтологій:

- Онтологія високого рівня: цей тип онтології представляє концепції незалежно від контексту чи ситуації. При цьому, оскільки концепції поза контекстом, вони повинні бути загальноприйнятими людьми. За допомогою цього підходу можна представити все або майже все.
- Онтологія домену: як впливає з назви, цей тип онтології задуманий у певному контексті. Це система знань, пов'язаних із галуззю, якою можуть бути сільське господарство, тваринництво, цивільне будівництво, тощо.
- Онтологія завдань: цей тип онтології використовується в контексті вирішення проблем для представлення концепцій, пов'язаних із конкретними завданнями, які виконуються. Це можуть бути завдання, пов'язані з дизайном, або ті, що пов'язані з хірургічною операцією.
- Онтологія програми: онтологія програми є досить специфічною, оскільки вона пов'язує дві конкретні онтології, одна з яких є онтологією домену, а інша -

онтологією завдання. Іншими словами, це концептуалізація завдань, які виконують актори поля, коли вони працюють.

Процес прийняття рішень можна поділити на кілька етапів. Спочатку необхідно визначити проблему та контекст, в якому вона виникла, а також обмеження та критерії, які необхідно врахувати при прийнятті рішення. Після цього настає фаза колективного розуміння проблеми, під час якої всі учасники обговорюють загальні погляди на проблему та шукають спільне рішення. Наступна фаза - створення альтернативних ідей для вирішення проблеми. Після цього відбувається обговорення та конфронтація різних поглядів з метою знайти оптимальне рішення. На останньому етапі приймається рішення на основі попередньо визначених критеріїв та спільної згоди учасників. Не менш важливим етапом є моніторинг всього процесу прийняття рішень з метою вчасно виявити будь-які проблеми та забезпечити їх виконання. На заключному етапі генерується документ, який стане основою для подальшої роботи.

Роботи, виконані Hernandez i Mothe i Lechchine [58], дозволяють виділити вісім кроків побудови онтології. Перший крок - визначення сфери знань, типу, цілей та майбутнього використання онтології, а також обсягу та методів для вилучення концепції. Другий крок - вибір корпусу документів для витягнення концепцій. Третій крок - лінгвістичне дослідження корпусу для виділення термінів та їх зв'язків. Четвертий крок - нормалізація результатів третього кроку, що полягає в ідентифікації та визначенні понять і семантичних зв'язків між ними. П'ятий крок - моделювання онтології за допомогою мови онтології для формального представлення семантичної мережі. Шостий крок - побудова структури онтології, що складається з ієрархії концепцій без надмірностей і неоднозначності та можливих нових зв'язків. Сьомий крок - перевірка онтології фахівцями предметної області. І останній крок - оновлення онтології з урахуванням змін та еволюції предметної області.

Після встановлення процесу необхідно визначити п'ять основних параметрів, які включають мету рішення, мету та обсяг прийняття рішення, кількість осіб, які приймають рішення, тривалість процесу та анонімність. Ці параметри визначають питання, на які повинна відповісти онтологія, включаючи багатокритеріальність або однокритеріальність прийняття рішень, використання моделей співпраці, учасників процесу прийняття рішень, достатню кількість осіб, які приймають рішення, тривалість процесу та необхідність анонімності. Всі ці запитання дозволяють зрозуміти необхідність розробки онтології для розрізнення знань про поле та оперативних знань.

Прийняття рішень є складним процесом, який розпочинається з постановки запитання та може призвести до знаходження відповіді на початкову проблему. Необхідно відзначити, що процес прийняття рішення відрізняється від самого рішення, яке є результатом процесу. Зазвичай цей процес складається з багатьох дій, процесів та подій, пов'язаних зі збором, обробкою та перевіркою інформації. В деяких випадках, прийняття рішення може бути складним процесом, що включає багато “підрішень”, які передують знаходженню вихідного рішення проблеми. Важливо зазначити, що існує значна онтологічна відмінність між процесом прийняття рішення та його результатом - рішенням. Крім того, при наданні більш еластичних меж для прийняття рішень, може статися так, що одне прийняте рішення матиме більше одного наслідку.

2.2. Узгодження онтологій у процесі прийняття рішень

Інтелектуальні агенти формують власні концептуалізації, які вони постійно доповнюють та коректують за результатами виконання завдань. Таким чином, концептуалізація бази знань агента відображає його досвід та сформовані знання щодо способів діяльності та виконання завдань. Ці концептуалізації відображені формально в онтології агента. Водночас, комплекс знань агента формує цілісну

систему знань, в якій деякі знання існують у неявній формі. Процес прийняття рішень інтелектуальним агентом поділяється на етапи [59, 60]:

- Виявлення та описання проблемної ситуації.
- Постановки задачі та побудови моделі вибору, яка містить мету, критерії, альтернативи і метод вибору.
- Вибору з альтернатив, оцінки наслідків вибору.
- Реалізації обраної альтернативи.

Важливим етапом цього процесу є виявлення та описання проблемної ситуації. Від правильності розуміння та подання моделі ситуації залежить коректність результатів процесу прийняття рішення загалом. Результатом опису проблемної ситуації є побудова її концептуальної моделі Md_{sit} . Така концептуальна модель створюється на основі онтології агента On_a .

На практиці, агенти рідко потрапляють у повністю нові проблемні ситуації. Знання про попередні ситуації зберігаються у вигляді концептуальних моделей-шаблонів у базі знань агента. При виявленні та ідентифікації проблемної ситуації агент вибирає ситуацію зі своєї бази знань яка найбільш подібна до виявленої ситуації та вносить необхідні корекції та оновлення у цю модель при необхідності. Агент оновлює модель ситуації за результатом виконання прийнятого рішення.

У багатьох випадках, для формулювання повної моделі ситуації необхідно отримати інформацію з баз знань інших агентів. Ці бази знань використовують інші онтології та, відповідно і інші концептуалізації предметної області, що відображають досвід діяльності цих агентів. Таким чином, на етапі формування моделі проблемної ситуації агент-децидент вирішує завдання:

- Вибору агента — власника знань (кореспондента), які необхідні для формування моделі ситуації.

- Узгодження онтологій обох агентів у контексті проблемної ситуації.

У загальному випадку агент, що приймає рішення може використати знання декількох агентів, щоб сформувати та доповнити свою модель проблемної ситуації.

На практиці агент, який потребує доступу до знань інших агентів, має можливість вибору агента з декількох альтернатив. При цьому має значення не тільки наявність в агента-кореспондента потрібної інформації, але й спосіб концептуалізації, який він використовує та який має відповідати концептуалізації проблемній ситуації агентом-децидентом.

Концептуалізація знань кожного агента відображає його характеристики та середовище в якому він працює, задачі які він вирішує. Врахування фактору середовища формування онтології при виборі агента для отримання інформації є важливим завданням, яке рідко враховують при виборі онтології. Більшість підходів до повторного використання та інтеграції онтологій обмежуються порівнянням змісту онтологій [61].

Деякі автори [62] на основі досвіду виконання проектів з інтеграції онтології прийшли до висновку, що для успішної інтеграції онтології потрібно врахувати не тільки зміст понять та відношень поданих в онтології, але й яким чином ці елементи онтології будуть використовуватися, тобто практичні аспекти використання онтології у певному середовищі. При вирішенні задачі вибору агента-кореспондента агент-децидент має врахувати задачі, що вирішує кандидат, сферу його діяльності, досвід, репутацію, рівень компетентності, юридичні та економічні аспекти перспективної взаємодії.

Задача вибору агента-кореспондента на основі інформації про його діяльність таким чином є багатокритеріальною задачею вибору, що може бути вирішена, наприклад методом аналітичної ієрархії [63]. Множина критеріїв визначає критерії вибору агента залежно від його середовища, функцій, ситуації прийняття рішення та

інших особливостей. Наприклад, можна визначити такі критерії вибору агента-кореспондента:

- Сфера діяльності, ступінь її близькості до потреб запиту.
- Кількість років роботи, досвід.
- Репутація, оцінки та відгуки клієнтів.
- Економічні аспекти (оцінка вартості послуг).
- Ризики та юридичні аспекти.

Важливість критеріїв (а також попарна важливість, для застосування методу аналітичної ієрархії) залежить від конкретної ситуації та зберігається агентом разом з концептуальними моделями типових ситуацій. Вирішення задачі вибору агента-кореспондента, що має найбільш подібний профіль діяльності Інтелектуальний агент часто вирішує задачу вибору серед набору подібних баз знань та відповідних онтологій такої, яка найбільш точно відображає його власну діяльність, потреби та перспективу бачення предметної області. Можна зробити припущення, що інші агенти, що працюють у подібному середовищі та вирішують подібні задачі сформували концептуалізацію, яка потребуватиме меншого узгодження, та яка накопичила корисний досвід, який можна буде використати.

Важливою частиною процесу прийняття рішень є ідентифікація та концептуалізація проблемної ситуації. Помилки або неповнота такої концептуалізації призводять до хибного розуміння проблеми та неправильних рішень на всіх наступних етапах процесу прийняття рішення.

Припускаємо, що кожен агент Ag виконує певну цілеспрямовану діяльність у своєму середовищі. Ця діяльність визначається цільовим компонентом (компонентом намірів, Intentional component) In_a його системи знань, якій містить:

- Цілі SGL – визначення станів середовища, які агент має сформувати у результаті своїх дій.

- Плани *SPl* – знання про те, що треба зробити (послідовності дій, завдань, методи та алгоритми) щоб досягти певної цілі залежно від стану середовища.
- Критерії, цінності, тести - *SCr* які дозволяють оцінити успіх (неуспіх) у досягненні цілі, ступінь досягнення цілі, вплинути на вибір дій при формуванні планів.

Цільовий компонент знань агента подається як набір концептуальних моделей (патернів) сформульованих з використанням онтології агента On_a .

Виконання завдань, визначених у цільовому компоненті є важливим джерелом проблемних ситуацій, які виникають під час виконання. Інше джерело проблемних ситуацій для інтелектуального агента – це непередбачені зміни у середовищі, а також результати взаємодії з іншими агентами. Такі ситуації виявляють за набором ознак, комбінації яких вказують на певні типові ситуації та відповідні концептуальні моделі. При цьому агент порівнює стан середовища з типовими концептуальними моделями ситуацій, які зберігаються у репозиторії моделей.

Моделі ситуацій у репозиторії – це шаблони, які релевантні певним класам подібних ситуацій та які можна налаштовувати приймаючи до уваги цілі агента та стан середовища. Вибір моделі здійснюється за подібністю біжучого стану середовища та цілей до моделі – шаблону. Вибір подібної моделі можна розглядати як задачу розпізнавання образів використовуючи функцію подібності $F_{sim}(Cm_i, In_a, Cm_j)$.

Таким чином, задача вибору моделі ситуації з репозиторію сформулюємо як задачу пошуку концептуальної моделі ситуації Cm_{sit}^k що мінімізує функцію відстані, з урахуванням обмежень, які накладаються цільовою компонентою: $F_{sim}(Cm_i, Cm_j, In_a) \rightarrow min$. Інтелектуальний агент опрацьовує інформацію використовуючи робочий набір концептуальних моделей, які постійно оновлюються.

Контекстна модель Cm_{con} відображає об'єкти середовища та цільовий компонент, який сприймається агентом в даний момент. Вона відображає залежності між об'єктами, їх властивості та надає інформації необхідну для прийняття рішення щодо наступних дій. Контекстна модель оновлюється якщо виникають зміни у середовищі або намірах агента.

Моделі ситуацій $SCm_{sit} = \{Cm_{sit}\}$ специфікують завдання, що виконує агент, а також проблеми, що виявлені у середовищі та які потребують реагування. Порядок опрацювання моделей використовує пріоритети з метою своєчасного реагування та зменшення використання ресурсів.

Модель наявної ситуації Cm_{sit}^{cur} містить елементи з контекстної моделі Cm_{con} (визначені через відображення з обраної моделі Cm_{sit}^k з репозиторію) а також інші компоненти, які додані моделлю Cm_{sit}^k .

Усі концептуальні моделі формують з використанням елементів онтології On_a агента. Водночас, кожна модель містить тільки підмножину релевантних до ситуації елементів онтології. Можна сказати, що існують контекстна онтологія та онтологія ситуації, які є цілісними логічними міні-теоріями отриманими шляхом виокремлення з первинної онтології агента On_a .

Розглянемо концептуальну модель проблемної ситуації детальніше. Цільовий компонент In_a містить структуру взаємозалежних цілей $SrGl$, множину планів SPl , та множину критеріїв SCr : $In_a = (SrGl, SPl, SCr)$.

Структура $SrGl$ містить множину цілей $\{Gl_i\}$ та залежності між ними $DpGl$: $SrGl = (\{Gl_i, DpGl\})$.

Кожна ціль визначена концептуальною моделлю кінцевого стану Cm_{st} та множиною тверджень (тестів, пост-умов) $\{Asrt^j\}$ які повинні бути виконані у середовищі з досягнутою ціллю: $Gl_i = (Cm_{st}^i, \{Asrt^j\}^i)$.

Плани подаються як множина концептуальних патернів (моделей), що визначають процедурні знання, релевантні у контексті наявного стану середовища та цілей: $SPl = \{Pl_i\}$, $Pl_i = (Cn_i, Pt_i, Ts_i)$, де Cn_i – це контексти застосування знань, що залежать від наявного стану середовища та цілей, Pt_i – це структура завдань (дій, які треба виконати) подана графом. Цей граф також містить посилання на проміжні стани, процедури перевірки їх успішного досягнення та специфікації додаткових дій у випадку збоїв, Ts_i – це постумови, що перевіряються для визначення успіху у досягненні цілі.

Незважаючи на наявність тестів та перевірок успішності виконання завдань та досягнення цілей, доцільно також мати множину критеріїв які визначають переваги у процесі прийняття рішень, наприклад, надання переваги простішим та дешевшим рішенням, дешевизні перед продуктивністю.

Будемо розрізняти проблемні ситуації, що виникають у процесі досягнення цілей системою та непередбачені ситуації, виявлені у середовищі та такі, що потребують реагування. В обох випадках ми будемо моделювати їх як різницю (проміжок) між наявним станом та цільовим станом середовища, разом з процедурними знаннями, критеріями які необхідні для визначення дій необхідних для подолання цієї різниці. Проблемна ситуація Ps_g , що виникла у процесі досягнення цілі Gl_i подається як: $Ps_g = (Cm^{cur}, Gl_i, In_a)$. А проблемна ситуація Ps_d , що виявлена у середовищі як: $Ps_d = (Cm_{sit}^{cur}, Cm_{sit}^{tar}, In_a)$, де Cm_{sit}^{cur} – це модель

ситуації, оновлена даними з моделі контексту, Cm_{sit}^{tar} концептуальна модель кінцевого стану, отримана з репозиторію моделей.

Також розглянемо комунікацію інтелектуальних агентів в процесі прийняття рішень. Агент-децидент, після побудови концептуальної моделі проблемної ситуації часто виявляє, що для прийняття рішення бракує інформації або знань. В цьому випадку можна отримати необхідну інформацію або знання з баз знань інших агентів. Однак, ці агенти використовують власні концептуалізації предметної області та онтології, що можуть відрізнятися від концептуалізації агента – децидента. Таким чином, для отримання і успішного використання інформації від іншого агента, необхідно попередньо узгодити онтології.

Успішна реалізація комунікації агентів передбачає вирішення таких задач:

- *Пошук та вибір агента – кореспондента.* На практиці потрібна інформація може пропонуватися декількома агентами. Агент-децидент використовує різні критерії для вирішення задачі вибору такі як якість та достовірність даних (знань), вартість, репутація, компетентність.
- *Узгодження інформації.* У цьому випадку агент-децидент потребує додаткової інформації про конкретний об'єкт що є екземпляром певного концепту онтології. Перед отриманням даних про цей об'єкт необхідно перевірити (та, при необхідності, узгодити) узгодити концептуалізації цього концепту в обох агентах у контексті проблемної ситуації.
- *Узгодження знань.* У цьому випадку агент-децидент потребує знань про те, яким чином вирішити певну задачу або виконати завдання. Знання про це передаються у формі патернів, структур, концептуальних моделей. Вони сформульовані з використанням елементів онтології агента-кореспондента. Під час комунікації потрібно узгодити не тільки зміст окремих використаних концептів та відношень, але й концептуальні моделі загалом.

- *Узгодження операцій.* Відбувається тоді, коли агент-децидент делегує виконання операції іншому агенту. При цьому він має перевірити, чи дійсно ця операція за змістом відповідає заявленій, а також дотримання інших вимог. При узгодженні операцій також виконують узгодження концептуальних моделей та онтологій обох агентів, що взаємодіють.

Перед отриманням необхідної інформації від агента-кореспондента агент-децидент має обрати його з множини подібних агентів, що надають подібні послуги. Критерії вибору включають крім вимоги щодо наявності потрібної інформації ще й оцінювання того, наскільки близька концептуалізація агента – кореспондента до концептуалізації агента-децидента. Кожна концептуалізація має як явну компоненту, подану онтологією, так і неявну компоненту, що знаходить своє відображення у початкових припущеннях при проектуванні онтології, виборі тих чи інших компонент онтології та їх зв'язків, обґрунтуванні аксіом та ін.

При оцінюванні ступеня подібності онтологій доцільно використати семантичні метрики, подані, наприклад, у роботах [64, 65].

Неявна компонента концептуалізації відображає властивості агента, його наміри, вартості та уподобання, середовище, в якому він працює. Якість концептуалізації яка залежить від досвіду агента та відображена у його репутації рідко коли враховують при виборі онтології. Наявні на сьогодні підходи головним чином при виборі онтологій враховують тільки її зміст.

Водночас, практичні аспекти використання онтології повинні бути враховані при її виборі. На приклад [62] на основі значного досвіду з вирішення задач по інтеграції онтологій стверджує, що для успішної інтеграції ми повинні враховувати не тільки зміст відношень та понять онтології, але й як ці елементи будуть використовуватися, практичні аспекти використання онтології у визначеному середовищі. Таким чином, у процесі вибору агента-кореспондента, агент-децидент

має прийняти до уваги завдання, які він виконує, рівень досвіду та репутацію, вартість, ризики та інші аспекти, що впливають на проведення взаємодії.

Агент-децидент формує запит додаткової інформації виходячи з аналізу концептуальної моделі проблемної ситуації Ps . Цей аналіз виявляє брак даних про певні властивості елементів концептуальної моделі, або відсутність процедурних знань про порядок виконання операції. У випадку участі оператора - людини за результатами аналізу формують питань на природній мові, що відносяться до потрібних даних чи знань. На наступному кроці ці питання перетворюють у запити на мові SPARQL до онтології.

У випадку чисто машинного аналізу, ці запити відразу формують на SPARQL відносно елементів концептуальної моделі ситуації Cm_{sit} . Ці запити групують тематично, відповідно до послуг, які надають інформаційні провайдери (потенційні агенти-кореспонденти).

У загальному випадку онтологія проблемної ситуації в агента – децидента формується в результаті поповнення елементами з онтологій інших агентів, та попереднього узгодження в контексті цієї проблемної ситуації.

2.3. Аналіз проблем використання онтологій в системах підтримки прийняття рішень

Проблеми використання онтологій в системах підтримки прийняття рішень можна поділити на дві категорії: технічні та концептуальні проблеми.

Технічні проблеми використання онтологій:

- Складність побудови онтологій: побудова онтології може бути дуже складною і трудомісткою задачею, яка потребує високого рівня експертизи з боку фахівців в області. Це може затримати розробку системи підтримки прийняття рішень.

- Складність підтримки та змінення онтологій: якщо потрібно додати нові поняття або змінити вже існуючі, то це може бути досить складно і трудомістко. Це може призвести до того, що онтологія стає застарілою або неактуальною, що знижує ефективність системи підтримки прийняття рішень.

Концептуальні проблеми використання онтологій:

- Стандартизація та зрозумілість онтологій: немає стандартів, які б допомогли використовувати онтології в різних системах та галузях знань. Крім того, може бути важко зрозуміти, яким чином визначені поняття та залежності між ними відображені в онтології.
- Незрозумілість метрик та критеріїв: використання онтологій може бути важко зрозуміти для користувачів, які не мають досвіду в даній галузі знань. Це може призвести до того, що користувачі можуть робити помилкові висновки або неправильно інтерпретувати результати системи підтримки прийняття рішень.

У цілому, використання онтологій в системах підтримки прийняття рішень має свої переваги і недоліки, і підходить не для всіх видів задач. Проте, при правильному використанні онтології можуть значно поліпшити якість прийнятих рішень та зменшити час, необхідний для їх прийняття.

Одна з можливостей вирішення концептуальних проблем полягає в розробці стандартів, що допоможуть зрозуміти, яким чином використовувати онтології в різних системах та галузях знань. Крім того, можна надати користувачам можливість переглядати та редагувати онтології для полегшення їх розуміння та забезпечення актуальності.

Щодо технічних проблем, можна використовувати автоматичні методи для створення та оновлення онтологій. Наприклад, машинне навчання може бути використано для автоматичного збору інформації з веб-джерел та розробки онтології на основі цих даних.

Крім того, можна використовувати методи зіставлення онтологій для вирішення проблем, пов'язаних з незгодою між різними онтологіями. Зіставлення може допомогти у визначенні спільних та відмінних елементів різних онтологій, що полегшить їх подальше використання та інтеграцію.

Отже, використання онтологій в системах підтримки прийняття рішень має свої виклики та складнощі, проте з відповідними рішеннями може бути дуже корисним для поліпшення якості прийнятих рішень та зменшення часу, необхідного для їх прийняття.

Застосування онтологій у системах підтримки прийняття рішень є активною галуззю досліджень. Найбільш відомими сьогодні авторами на тему онтологій є Т. Грубер, Н. Гуаріно, Б. Сміт, Дж. Сова, А. Гомес-Перес. У галузі просторових та геоінформаційних онтологій поряд з Б. Смітом та О. С. Гомес-Перес добре відомі також А. Марк, М. Егенхофер, Ж. Теллер, К. Руссо, Р. Лауріні, останніми роками – Ф. Фонсека, Н. Каза, Х. Шиверс.

Один із прикладів використання онтологій для покращення систем підтримки прийняття рішень пов'язаний з медичною сферою. У своєму дослідженні [66] “Ontology-Based Clinical Decision Support Systems: A Scoping Review and Classification of Uses” Анджело Антоніно Наполі та ін. оцінювали використання онтологій у медичних системах підтримки прийняття рішень та визначили, що використання онтологій може допомогти знизити час, необхідний для здійснення діагностики та прийняття рішень, а також поліпшити якість прийнятих рішень.

Інший приклад використання онтологій у системах підтримки прийняття рішень пов'язаний з галуззю екології. У дослідженні [67] “An ontology-driven web application for environmental decision-making: An example of pesticide exposure assessment” автори Елінора Гузеева та Денніс Росс використали онтологію для розробки веб-застосунку, що допомагає оцінити ризики впливу пестицидів на довкілля. Вони довели, що використання онтологій може полегшити процес

прийняття рішень, а також забезпечити збереження даних та розуміння взаємозв'язків між елементами.

Розроблене для вирішення цих завдань різне програмне та інформаційне забезпечення, незважаючи на наявність спільної інформаційної компоненти часто має проблеми коректного стикування баз даних і програмних додатків між собою. Прогалини в існуючому науковому знанні роблять його використання проблематичним, а подальшу модернізацію з урахуванням нового знання неможливою без суттєвого доопрацювання, ціна якої часто перевищує вартість нового програмного коду.

Однак невирішеність питання появи нового об'єктивного знання про предметну область та проблеми, з якими розробляються інформаційні системи, ставить завдання його виявлення, і навіть формалізації, опису і фіксації нових вимог.

З одного боку, при складних ситуаціях, що часто виникають, підтримки прийняття рішень слабоструктурованих проблем доцільною стала розробка різних експертних систем, заснованих на знаннях, отриманих від фахівців-експертів у відповідних галузях знання.

З іншого боку, з розвитком складності та загальної величини інформації, що виробляється у світі, загальнодоступних мереж її передачі (інтернет та інтранет), стали розроблятися знання орієнтовані на системи з метою класифікації та ефективного інформаційного пошуку віддалених даних та документів у корпоративних сховищах та всесвітній WWW-павутині [68].

Спроби уніфікації різних підходів під час вирішення складних завдань, вимагають наявності інтелектуальної компоненти, у тому числі з використанням знання орієнтованих систем, призводять до розуміння необхідності створення загальноновживаної термінології, категоризації понять та уніфікації їх сенсу, тобто, загального та доступного для багатьох розуміння та описи явищ та проблем різних предметних областей. Вказаною ціллю покликане служити створення комп'ютерних

онтологій, що входять до узагальненої категорії систем, що ґрунтуються на знаннях. На відміну від експертних систем, де формалізуються та використовуються знання окремих експертів, онтології будуються як узгоджений та широко розділений опис деякого узагальненого домену знань (як правило, проблемно орієнтованого). Таким чином, проблема створення та використання онтологій як опису знання предметних областей, є важливим науково та практичним завданням побудови систем, заснованих на знаннях, успіх вирішення якої сприяє розвитку комп'ютерної інженерії, інформатики, штучного інтелекту та інших фундаментальних та прикладних дисциплін, що розширює можливості прикладного застосування передових інтелектуальних інформаційних технологій у багатьох предметних галузях (зокрема і тих, де вони раніше не використовувалися).

Розробник онтологій змушений вирішувати безліч проблем як типових, що виникають під час проектування та мають відомі способи вирішення, так і специфічних, вирішення яких залежить від конкретної предметної галузі.

До типових проблем побудови онтологій належать такі [69]:

- Повнота та коректність інформації;
- Семантична структуризація;
- Сприймання людиною;
- Адаптація до розширення та реорганізації;
- Оцінка якості;
- Надмірність інформації.

Специфічні проблеми пов'язані з визначенням кордонів предметної області, що формалізується, і двовимірною структуризацією онтології.

Встановлено, що досвідчений консультант витрачає в середньому у 2.5 рази менше часу на вирішення проблеми, ніж недосвідчений консультант. При цьому останній, використовуючи прості засоби запису та отримання знань про рішення проблем, здатний наблизити власну ефективність до рівня досвідченого

спеціаліста-аналітика. Побудова в галузі ІТ-підтримки повноцінної інтелектуальної системи управління знаннями, може дозволити накопичувати, систематизувати, та ефективно використовувати досвід аналітиків із консультування співробітників організації.

У системі знань у галузі ІТ-підтримки використовуються міркування з урахуванням прецедентів. Кожен прецедент включає наступні елементи:

- Опис ситуації за допомогою атрибутів;
- Рішення, яке було ухвалено в цій ситуації;
- Обґрунтування (результат) застосування рішення.

Пропонується метод інтеграції прецедентів консультування з онтологіями предметної галузі ІТ-консультування на основі встановлення асоціативних зв'язків прецедентів із концептами онтології.

Кожен прецедент пов'язаний не з одним, а з кількома концептами онтології, що знаходяться на різних рівнях ієрархії, що дозволяє найбільш адекватним способом описати семантику того чи іншого прецеденту консультування користувачів засобами онтології. Для отримання прецедентів замість міри близькості за атрибутами використовується порівняння між поточною ситуацією та прецедентами, оцінюючи ступінь їхнього зв'язку з концептами онтології, тобто семантичну близькість пов'язаних із цими прецедентами концептів. Для цього на етапі створення бази знань при внесенні кожного нового прецеденту необхідно встановити смислові зв'язки нововведеного прецеденту з онтологією концептів.

Пропонований метод індексування прецедентів може бути використаний для подальшого видалення прецедентів, релевантних поточній ситуації, з використанням методу побудови нечітких лінгвістичних правил. Даний метод найбільш ефективний при накопиченні великої бази знань та витягу з неї найбільш семантично близького прецеденту.

У роботі [70] досліджується застосування онтологій у програмній інженерії з використанням гнучкого підходу, тобто постійної адаптації вимог в умовах мінливості факторів бізнес-середовища. Авторами реалізовано OWL-онтологію, яка виконує міркування про вимоги до програмного продукту. Самі вимоги подані у вигляді історій, написаних природною мовою. В свою чергу, користувальницькі історії включають роль користувача, досягнуту мету, цінність для користувача, критерій прийому.

Ця онтологія входить до складу архітектури системи підтримки прийняття рішень, що складається також з інтерфейсу користувача, програми на мові Python, машини логічного виводу, файлової бази даних. У процесі ухвалення рішення онтології дозволяють організувати зберігання знань у нереляційних базах даних.

Отже, дослідження показують, що використання онтологій може бути корисним для покращення систем підтримки прийняття рішень у різних галузях знань.

Висновки до розділу 2

Онтологічне моделювання є ефективним інструментом для підтримки прийняття рішень в умовах багатозначності та контекстної залежності. Цей підхід допомагає визначити особливості предметної області, збирати та структурувати знання і використовувати їх для прийняття рішень. Онтологічні мережі є важливим інструментом онтологічного моделювання. Онтологічні мережі відображають взаємозв'язки між поняттями предметної області та їхніми характеристиками. Вони допомагають у визначенні структури та взаємодії між елементами системи, збиранні та організації знань. Рекомендації щодо застосування онтологічного моделювання включають визначення якнайбільш точних та релевантних понять, що відображають предметну область, і створення структурованих моделей знань, які дозволяють легко здійснювати пошук та аналіз.

Результати розділу опубліковано у [1, 3, 6, 7, 8, 9, 10, 11].

РОЗДІЛ 3. Методи та алгоритми побудови системи підтримки прийняття рішень на основі онтологічних мереж

3.1. Метод багаторівневого узгодження структур на основі онтологічних мереж

У сучасному світі прийняття рішень є ключовим процесом у багатьох сферах діяльності, від бізнесу до науки та технологій. Одним з найбільш ефективних інструментів прийняття рішень є використання онтологій, які дозволяють формалізувати знання про предметну область задачі та використовувати ці знання для підтримки процесу прийняття рішень. Однак, створення онтологій може бути складним та непередбачуваним процесом, що може призвести до помилок концептуалізації та невірного використання онтологій у процесі прийняття рішень.

Одним з найважливіших етапів створення онтологій є узгодження змісту онтології з предметною областю задачі. Однак, цей етап може бути досить складним, оскільки він передбачає розуміння предметної області задачі, вибір та формалізацію концептів та їх відносин, та узгодження змісту концептів з експертами та користувачами.

Багаторівневий та поетапний підхід до узгодження змісту онтології предметної області задачі прийняття рішень являє собою процес узгодження онтологічних моделей на різних рівнях абстракції та змісту, включаючи більш загальні та абстрактні концепції, а також конкретні деталі та відношення між об'єктами.

На першому рівні, підходом багаторівневого узгодження є визначення загальних концепцій та термінів, які відображають базову логіку предметної області. Цей рівень є вихідним пунктом для подальшого уточнення концепцій та зв'язків між об'єктами на більш детальному рівні. На цьому етапі вирішуються основні питання щодо організації інформації та визначення мета-об'єктів, досягнення яких має допомогти розроблена онтологія.

На другому рівні, підходом поетапного узгодження є детальне визначення понять та відношень між об'єктами, які були визначені на першому рівні. Цей етап включає уточнення взаємодії між об'єктами та визначення їхніх властивостей та атрибутів. На цьому етапі вирішуються деталі, пов'язані з ієрархією концепцій та відношеннями між ними.

Тобто, починаємо з визначення загальних понять на багаторівневому рівні, а потім переходимо до поетапного визначення більш детальних понять та відношень між ними. Після цього потрібно повернутися до багаторівневого рівня, щоб переконатися, що концепції та зв'язки на більш детальному рівні відповідають загальним принципам та логіці предметної області.

Важливим аспектом розробки онтології є також забезпечення її взаємодії з іншими онтологіями та засобами обробки інформації. Для цього потрібне використання відкритих стандартів та протоколів, що дозволять обмінюватися інформацією та взаємодіяти з іншими онтологіями та засобами.

Узгодження змісту онтології може бути складним і тривалим процесом, який вимагає високої кваліфікації та досвіду в області предметної області та онтологічного моделювання. Однак, результатом успішного узгодження буде створення якісної та детальної онтології, яка допоможе вирішувати складні задачі прийняття рішень відповідно до встановлених мета-об'єктів.

Для ефективного прийняття рішень необхідно сформулювати коректну концептуальну модель проблемної ситуації, що містить концепти, відношення та обмеження, які найточніше відображають цю ситуацію. Побудована модель дає змогу концептуалізувати проблемну ситуацію, що формально представляється як онтологія. Використання онтологій проблемних ситуацій дозволяє зберігати та повторно використовувати знання про прийняття рішень та використовувати інтелектуальні програмні агенти для підтримки прийняття рішень.

Для побудови онтології проблемної ситуації рекомендується використовувати

знання з існуючих онтологій, оскільки це допомагає забезпечити її якість та стандартизацію. Однак, це є непростю задачею через наявність подібних за компетенцією онтологій. Принцип множинності онтологій проявляється як на рівні онтологій вищого рівня (SUMO, DOLCE, UFO, GFO), що використовують різні принципи побудови, так і на рівні галузевих онтологій.

Наприклад, в розробці GFO автори [45] використовують принцип множинності та розглядають GFO як набір різних онтологій, оскільки кожна з них відображає як спільне (common intentionality), так і специфічне розуміння авторів щодо предметної області. Ця проблема зустрічається на практиці в складності побудови єдиної схеми даних, навіть в межах одного підприємства, що послужило підставою розробки сервіс-орієнтованих архітектур [71]. У своїй роботі автори [72] аргументують переваги використання багатьох онтологій, в тому числі можливість використання різних уявлень та специфікацій концептів у різних онтологіях з метою вибору найкращого відповідника. Такі онтології утворюють мережі, в яких відношення відповідності між концептами та відношеннями різних онтологій встановлюються за допомогою правил відповідності (articulation rules). Під час побудови онтології проблемної ситуації виникає проблема багатозначності вибору, коли декілька онтологій мають подібні концепти, які можуть бути використані в онтології проблемної ситуації.

З урахуванням вище сказаного, перспективним рішенням є подача інформації у систему підтримки прийняття рішень з використанням онтологічних мереж. Кожна така онтологія містить інформацію про конкретну предметну область та відношення до інших мереж та функції, що дозволяють отримати необхідні дані під час процесу прийняття рішення.

Онтологію можна подати як:

$$O = \langle T, R, F \rangle, \quad (1)$$

де T – це терміни предметної області, R – відношення між ними, F – функції інтерпретації. Причому при формуванні системи підтримки прийняття рішень, множини R і F не можуть бути пустими. Тобто це є певна не пуста множина об'єктів, якій приписують такі властивості:

- Об'єкти формують деяку ієрархічну структуру;
- Параметри інтерпретуються відповідно до функцій;
- Визначенні параметри є аксіомами.

Така онтологія може бути представлення у вигляді певної інформаційної системи, яка складається з подібних параметрів та зв'язків між ними, а результатом такої системи буде онтологічна база знань.

Однак, при процесі прийняття рішень, є ще один важливий елемент, це критерій, за яким відрізняються альтернативи з відповідної множини. Саме критерій вносить значну роль у прийнятті рішення людиною.

Отже, онтологію проблемної ситуації для системи підтримки прийняття рішень можна представити як:

$$O' = \langle T, R, F, C \rangle, \quad (2)$$

де C – це критерії, які є у кожного елементу онтології та використовуються у задачі прийняття рішення. Таку задачу можна описати як, є множина альтернатив в якій у кожного елемента є спеціальні значення, що є критеріями цієї альтернативи. А розв'язком такої задачі є підмножина найкращих альтернатив профільтованих за важливістю значення критерія. Причому критерій може змінюватися на всіх етапах прийняття рішень.

У роботі [49] надані основні формальні визначення щодо відношень відповідності між онтологіями у мережі онтологій. Так, якщо є дві онтології O та O' , з відповідними мовами подання $Ent(O), Ent(O')$ та множина відношень відповідності R , відповідність між їх елементами визначено як трійку $\langle e, e', r \rangle$

таку що $e \in Ent(O)$, $e' \in Ent(O')$ та $r \in R$. Іншими словами, відповідність, це твердження, що між концептами e та e' існує відношення відповідності r .

Відповідність між онтологіями у цілому визначено як множину усіх відповістей між елементами.

Мережа онтологій визначена як двійка:

$$\langle M, V \rangle, \quad (3)$$

де M – це множина онтологій, а V – це відповідність між ними. Часто у позначенні відповідності вказують, між якими онтологіями вона встановлена, наприклад $V(O, O')$.

Частиною відношення відповідності є правило відповідності (*ArtRu* - articulation rule), яке встановлює яким чином з елементу онтології O отримати елемент онтології O' . У найпростішому випадку це просто відображення між елементами двох онтологій:

$$ArtRu: e \rightarrow e', \quad (4)$$

У загальному випадку правило відповідності це відображення підмножини концептів, відношень та функцій інтерпретації вихідної онтології в підмножину концептів, відношень та функцій інтерпретації результуючої онтології:

$$ArtRu: \langle Sub(T), Sub(R), Sub(F) \rangle \rightarrow \langle Sub(T'), Sub(R'), Sub(F') \rangle, \quad (5)$$

де $Sub(T), Sub(R), Sub(F)$ це підмножини концептів, відношень та функцій інтерпретації вихідної онтології, а $Sub(T'), Sub(R'), Sub(F')$ – підмножини концептів, відношень та функцій інтерпретації для результуючої онтології.

Онтологія проблемної ситуації O_s формується (рис. 3.1) на основі деякого шаблону – моделі ситуації розробленого експертом. На початку він містить елементи та відношення, які після виконання операції встановлення відповідності заміщуються елементами з інших онтологій. При цьому можуть використовуватися різні онтології для відображення різних частин онтології проблемної ситуації. На

рис. 3.1, як приклад, показано що онтологія проблемної ситуації С ініціалізується концептами та відношеннями з онтологій А, В та D. При цьому один з концептів онтології ситуації сформовано на основі концептів двох онтологій.

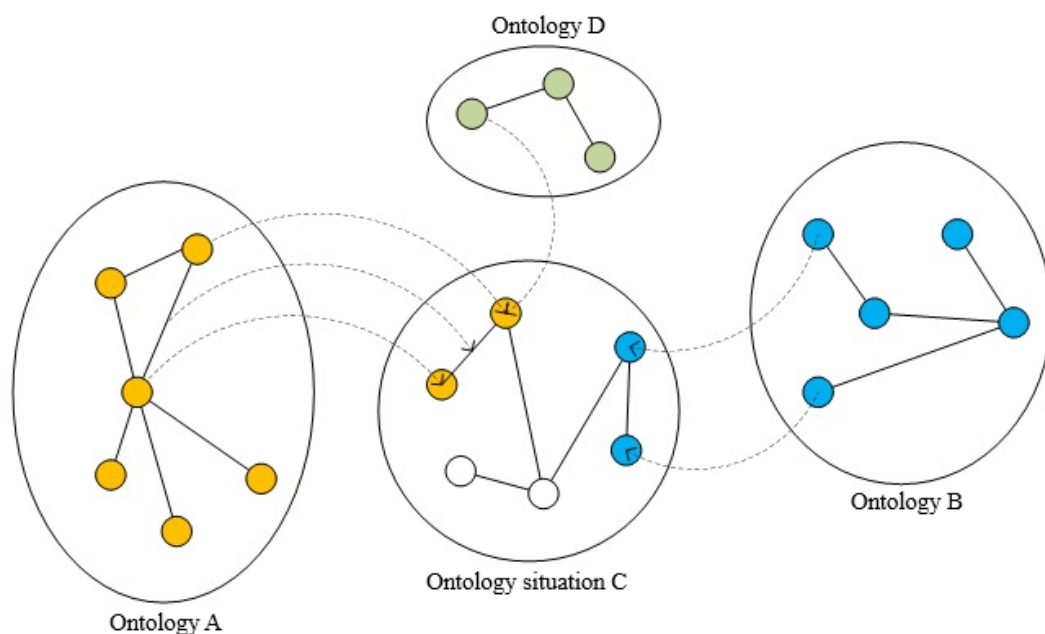


Рис. 3.1. Формування онтології проблемної ситуації у мережі онтологій.

На рис. 3.2 наведено структуру системи підтримки прийняття рішень з використанням онтологій. Під час процесу навчання системи, експерт формує її базу знань. Остання включає в себе онтологію задач, термінів та зв'язків.

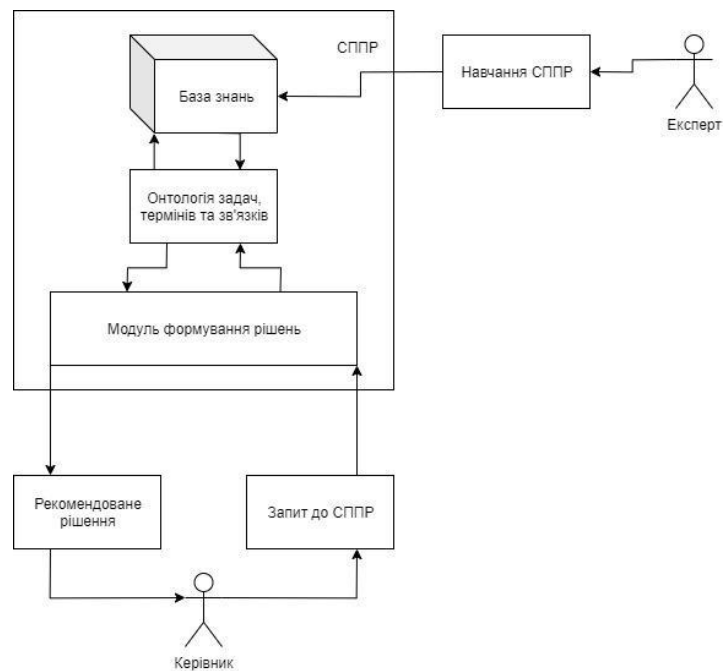


Рис. 3.2. Структура СПДР з використанням онтологій.

Загалом процес прийняття рішення проходить у кілька етапів:

1. Постановка питання.
2. Збирається вся доступна інформація на момент прийняття рішення, джерелом якої є онтологічні мережі компаній.
3. Формування критеріїв.
4. Побудова онтологічної мережі, тобто множини альтернатив.
5. Фільтрація альтернатив.
6. Розбір питання та формування критеріїв питання.
7. Фільтрація альтернатив.
8. Вибір найкращої альтернативи.

Деякі етапи процесу прийняття рішень можуть бути делеговані сучасним службам. У цій системі наявні два входи: один для експерта, чия задача полягає у поповненні/оновленні джерел знань, а інший для керівника з можливістю задавати запити до системи. Після цього слідує модуль джерела знань, який містить усі підключені онтології та зв'язки між ними. База знань формує один із модулів системи підтримки прийняття рішень, а сама база знань використовується ядром.

Модуль ядра відповідає за всі інші задачі, такі як: обробку запиту від керівника, завантаження необхідних знань з бази знань згідно з термінами, формування множини альтернатив, вибір альтернатив відповідно до запиту, вибір найкращої альтернативи та виведення рішення.

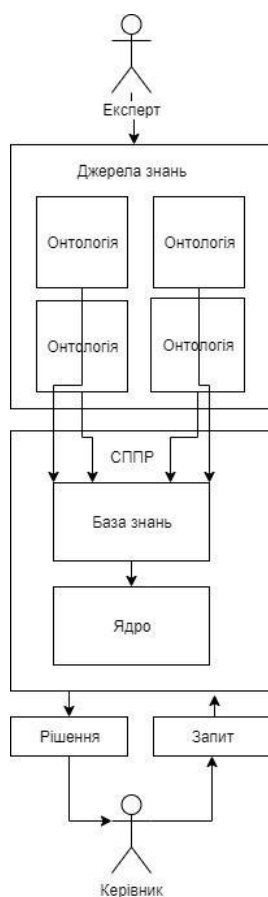


Рис. 3.3. Структура базової СППР з використанням онтологічних мереж.

Як показано на рис. 3.3, мережа онтологій використовується на всіх етапах роботи системи. В першу чергу, вона використовується для формування бази знань через використання наявних зв'язків та створення нових. Під час роботи ядра онтологія допомагає швидко знайти відповідні дані.

У кожної компанії є свої експерти які слідкують за ринком та вибудовують плани для розвитку компанії. Ці ж експерти, на основі спостережень за ринком та за внутрішніми процесами в компанії, збирають собі дані у якісь внутрішні бази, які можливо описати як онтології, а вже ці онтології складають онтологічну мережу цієї

компанії. Зазвичай частина з цих даних випускається публічно, хтось може подумати що у цьому немає смислу, однак це будує бренд компанії і саме бренд допомагає у залученні нових користувачів. Ці публічні дані та приватні дані самої компанії і будуть використовуватися для системи підтримки прийняття рішень.

Надалі приклади описуватимуться тільки компаніями які надають послуги для людей та бізнесів, однак сам функціонал може бути використаний для будь якої категорії компаній.

У кожної компанії є свої функції та параметри. Під функціями мається на увазі приклади як прибирання квартири чи допомога з переїздом, також це можуть бути написання та підтримка сайту, та багато інших. Далі, параметри, це те що характеризує компанію, базово це місцезнаходження, кількість працівників чи річний прибуток, таких параметрів може бути безліч.

В чому різниця між цілями, задачами та функціями? Задача - перелік дій, які треба зробити, щоб досягнути цілей. Якщо ціль це певний орієнтир на результат, то задачі це перелік конкретних дій. Функції - це перелік постійних, більш-менш довгострокових очікуваних результатів людини на певній позиції. Це певна зона відповідальності. Тобто функцією менеджера може бути, наприклад, координація відділу продажу, а задачею на рік - переструктурувати відділ згідно новій організаційній структурі компанії [73].

Модель бізнес-функції є ієрархічною декомпозицією функцій, необхідних для досягнення певних цілей у бізнесі. Взаємозв'язки між цими функціями утворюють бізнес-процеси. У моделі бізнес-функцій використовуються терміни, що зрозумілі в бізнесі, а не в корпоративній системі. На рівні варіантів бізнес-функцій можуть відбуватися зміни, залежно від обраної бізнес-моделі.

Модель бізнес-функції може бути використана для різних цілей, зокрема [74]:

- Як перший крок у процесі моделювання, для створення моделі бізнес-процесу на основі загальних цілей та функцій.

- Як допоміжний інструмент для радників та консультантів під час презентацій для керівництва, де демонструється модель підприємства для певної галузі промисловості.
- Як засіб для передачі галузевого "ноу-хау" та найкращих практик ведення бізнесу та показників діяльності.
- Як інструмент для накладання нейтральної бізнес-моделі, що відповідає термінам, зрозумілим у бізнесі, на специфічне для корпоративної системи рішення.

Для створення референтної моделі архіву узагальнених бізнес-функцій для всієї промисловості необхідна система класифікації. У системі динамічного моделювання (DEM) використовується ієрархічне дерево бізнес-функцій для класифікації відповідно до стандартної структури Ернста і Янга. Ієрархічне дерево складається з чотирьох рівнів [75]:

- Перший рівень - Мега-функції, який охоплює шість узагальнених бізнес-функцій, не пов'язаних зі спеціалізацією діяльності компанії, а саме: управління, стратегічне планування, розвиток і виконання процесів здійснення..
- Другий рівень - Головні функції, які декомпонують Мега-функції і більше пов'язані зі спеціалізацією бізнесу компанії.
- Третій рівень - Базові функції, які представляють головні процеси у процесній моделі.

Функції бізнесу пов'язані з виробництвом та обмінними операціями між підприємцем та іншими елементами господарського середовища. Архів головних функцій буде зростати зі збільшенням кількості спеціалізованих моделей бізнесу.

Бізнес-ціль — це кінцева точка, досягнення або ціль, яких організація хоче досягти в короткостроковій або довгостроковій перспективі. Бізнес-цілі можуть набувати різних форм і бути прагненням або мотивацією, наприклад, спрямовувати

організацію до певної мети, як-от покращення обслуговування клієнтів. Вони також можуть мати дуже конкретні цілі, такі як досягнення певної цілі доходу, чистого прибутку, норми прибутку, прибутку або іншої фінансової віхи [76].

Заяву про місію часто розглядають як визначення мети та причини існування організації, що є формою бізнес-цілі. Заява про бачення – це ще один поширений спосіб для організації сформулювати свої цілі, надаючи перспективу того, куди вона хоче рухатися.

Чому бізнес-цілі важливі?

Бізнес-цілі важливі з багатьох причин, які можуть вплинути на загальну діяльність і успіх організації.

Бізнес-цілі допомагають оцінити прогрес. Забезпечують віхи, які можуть допомогти організації виміряти її успіх або відсутність такого.

Бізнес-цілі визначають напрямок компанії. Дозволяють усім співробітникам мати чітке уявлення про те, куди компанія хоче рухатися і якою вона хоче бути.

Бізнес-цілі встановлюють підзвітність. Дозволяють керівництву взяти на себе відповідальність за свої успіхи чи невдачі.

Бізнес-цілі покращують процес прийняття рішень. Вирівнюють діяльність підприємства, щоб керівництво могло постійно оцінювати рішення, щоб переконатися, що бізнес рухається до своєї мети.

Які елементи бізнес-цілі?

Щоб бути ефективною, бізнес-ціль складається з кількох компонентів. На самому базовому рівні всі бізнес-цілі визначають, чого організація хоче досягти протягом певного періоду часу. Абревіатура SMART зазвичай використовується, щоб переконатися, що правильні елементи присутні в бізнес-цілі.

Специфічний. Необхідно визначити бізнес-ціль.

Вимірний. Має бути якийсь показник, наприклад відсоток, щоб вимірювати прогрес.

Досяжна. Має бути певне очікування, що мета досяжна, навіть якщо це складне завдання.

Реалістичний. Ціль має бути реалістичною для діяльності компанії та відповідати її бізнес-плану.

Прив'язаний до часу. Мета повинна бути досягнута за певний час.

Різні типи бізнес-цілей і приклади [75]:

Фінансові цілі. Ці цілі часто є конкретними фінансовими цілями, яких компанія хоче досягти, щоб збільшити дохід або прибутковість. Хорошим прикладом фінансової цілі є бізнес, який має на меті первинне публічне розміщення.

Цілі зростання. Цілі цього типу спрямовані на розширення або зростання частини бізнесу на певний відсоток. Наприклад, зростання може відбуватися за рахунок нових продуктів, клієнтів або місць розташування. Іншим прикладом може бути планування розширення діяльності в кількох країнах.

Цілі співробітника. Допомога працівникам у досягненні власних цілей щодо просування по службі, балансу життя та роботи або кар'єрного зростання є важливою категорією бізнес-цілей. Це особливо вірно в епоху великої відставки, коли працівники шукають самореалізації.

Цілі процесу. Кожна організація має власні процеси для різних видів діяльності. Встановлення цілей щодо вдосконалення цих процесів і робочих процесів може допомогти оптимізувати ефективність бізнесу. Наприклад, метою процесу буде збільшення кількості випусків програмного забезпечення з покращеними процесами DevOps.

Соціальні цілі. Соціальні цілі, такі як сприяння різноманітності або встановлення цілей сталого розвитку для проблем навколишнього середовища, стають все більш стратегічною та важливою категорією в процесі встановлення цілей. Скорочення викидів парникових газів на певний відсоток або зменшення вуглецевого сліду організації підійде до цієї категорії.

Цілі на основі часу. Час має бути атрибутом кожної з цілей, описаних вище. Цілі, орієнтовані на час, включають короткострокові, а також довгострокові цілі та завдання. Короткострокові цілі можуть бути досягнуті за дні, тижні або кілька місяців, тоді як довгострокові цілі мають дату завершення від багатьох місяців до років.

Основні функції стосуються діяльності, що приносить дохід, наприклад, виробництва кінцевих товарів або послуг. Ці функції зазвичай є основною діяльністю компанії. Однак вони також можуть включати інші види діяльності, якщо компанія вважає їх частиною своєї основної функції.

Допоміжні функції стосуються діяльності всередині організації, спрямованої на виконання основних функцій. Вони не виробляють продукцію для продажу чи отримання прибутку, а радше вони надають послуги підтримки основних функцій.

Організація та управління бізнес-функціями є важливими, оскільки вони взаємопов'язані та взаємозалежні одна від одної. Ваша компанія розподіляє завдання та роботу на кілька функцій або відділів, щоб забезпечити ефективність організації. Кожна функція відіграє важливу роль для інших. Якщо одна з функцій відсутня або не працює належним чином, операції не виконуються або перериваються.

Усі бізнес-рішення, прийняті в кожному відділі, мають поєднуватися й враховувати інші частини. Це вимагає хорошого спілкування, співпраці та тісних стосунків між кожною функцією. Таким чином, кожен відділ підтримує один одного в досягненні організаційних цілей, не лише займаючись своїми власними цілями.

Давайте розглянемо приклад того, чому рішення в одному відділі залежать від іншого. Наприклад, для операційної функції потрібна інформація від відділу маркетингу про специфікації продукту та обсяг виробництва. Подібним чином функція маркетингу не продає продукти без операційної функції.

Коли специфікації продукту, яких вимагає ринок, потрапляють у виробничий відділ, команда думає про те, як виробляти ефективно. Наприклад, чи потрібно їм

купувати нові машини чи покладатися на існуючі виробничі потужності? Якщо необхідно придбати нову машину, виробничий відділ складає проект бюджету для подання до фінансового відділу.

Потім фінансовий відділ розглядає, чи достатній наявний бюджет. Якщо цього недостатньо, команда думає про фінансування, чи потрібно компанії залучати капітал, наприклад, шляхом випуску боргових зобов'язань або акцій.

Фінансовий відділ приймає рішення про виділення грошей не лише для виробничого відділу, а й для інших бізнес-функцій, будь то маркетинг і кадри. Наприклад, відділу маркетингу потрібні кошти для просування продукції, або відділу кадрів для набору нових працівників.

Тоді для успішного виконання кожної функції вашій компанії потрібні кваліфіковані та продуктивні люди. У цьому випадку на перший план виходить управління людськими ресурсами. Ця функція набирає працівників, забезпечує навчання та розвиток, встановлює винагороду та мотивує їх надавати найкраще для вашої компанії.

Види бізнес-функцій. Чотири основні бізнес-функції компанії [77]:

- Операція;
- Людські ресурси;
- Маркетинг;
- Фінанси та бухгалтерський облік.

Операція. Функцію операцій іноді називають виробничою функцією. Ви, напевно, також знайомі з терміном виробничий відділ. Як випливає з назви, ця функція відповідає за діяльність з виробництва товарів або надання послуг.

Людські ресурси. Цей відділ відповідає за управління працівниками компанії.

Маркетинг. Відділ маркетингу відповідає за маркетинг, дослідження ринку, збут і післяпродажне обслуговування. Команда визначає потреби та бажання клієнтів і розробляє відповідний маркетинговий комплекс для їх задоволення. Маркетинг

отримує важливу інформацію за допомогою дослідження ринку, щоб визначити та зібрати таку інформацію, як потреби споживачів, ринкові перспективи та ринкова конкуренція. Потім він використовується як довідник для розробки маркетингових планів і конкурентних стратегій.

Облік і фінанси. Бухгалтерський і фінансовий відділ займається управлінням грошима всередині компанії, включаючи моніторинг надходження та вибуття грошей. Зокрема, на фінансову функцію покладено:

Приклади компаній із їхніми функціями:

Таблиця. 3.1.

Назва компанії	Функції компанії
CleanToday	Прибирання дому, прибирання офісу, прибирання магазину, прибирання гаражу, оренда приборів для прибирання, ...
MoveNow	Переїзд з квартири, переїзд офісу, переїзд магазину, прибирання при переїзді, оренда машин для переїзду, надання складу, ...
OofWork	Розробка сайту, розробка мобільної програми, допомога із бухгалтерією, допомога із дизайном, допомога із маркетингом, ...
CleanPlus	Прибирання дому, хімчистка, прибирання офісів, прибирання, ...
MyMaster	Створення сайтів, технічна підтримка, інтернет-маркетинг, ...

Приклади параметрів компанії:

Таблиця. 3.2.

Назва компанії	Рік заснування	Кількість співробітників	Дохід за рік
CleanToday	2015	65	50.000 ГРН
MoveNow	2014	30	42.000 ГРН
OofWork	2018	125	120.000 ГРН
CleanPlus	2016	15	50.000 ГРН
MyMaster	2015	15	100.000 ГРН

У кожної з цих компаній є свої бази даних, які можна охарактеризувати онтологіями або ж онтологічними мережами, коли база даних компаній складається із кількох, а отже і із кількох онтологій. Теми онтології можуть бути різні, однак в основу для цього виступають функції компанії.

Таблиця. 3.3.

CleanToday			
Функція	К-сть клієнтів	Середня площа	Середня тривалість
Прибирання дому	85 клієнтів	60 м ²	2 год.
Прибирання офісу	12 клієнтів	500 м ²	3 год.
Оренда приборів для прибирання	140 клієнтів	-	2 дні

...	...	...	...
-----	-----	-----	-----

Таблиця. 3.4.

MoveNow			
Функція	Тип будівлі	Середня відстань між місцями	Середній об'єм
Переїзд з квартири	Квартира	80 км.	2 машини
Переїзд офісу	Офіс	80 км.	6 машин
Надання складу	Склад	-	12 м ²
...	...	...	...

Таблиця. 3.5.

OofWork			
Функція	К-сть клієнтів	Середня тривалість	Найчастіше використовувані технології
Розробка сайту	75 клієнтів	2 місяці	HTML/CSS/JS
Розробка мобільної програми	250 клієнтів	4 місяці	Flutter
Допомога із	120 клієнтів	1 рік	Google Ads

маркетингом			
...	...	...	...

Таблиця. 3.6.

CleanPlus			
Функція	К-сть клієнтів	Середня площа	Середня тривалість
Прибирання дому	100 клієнтів	40 м ²	3 год.
Прибирання офісу	20 клієнтів	300 м ²	6 год.
Хімчистка	300 клієнтів	-	1 день
Продаж товарів	200 клієнтів	-	1 місяць
Допомога із маркетингом	5 клієнтів	-	1 рік
...	...	...	...

Таблиця. 3.7.

MyMaster			
Функція	К-сть клієнтів	Середня тривалість	Найчастіше використовувані технології
Розробка сайту	15 клієнтів	2 місяці	HTML/CSS/JS/

Розробка мобільної програми	25 клієнтів	4 місяці	Нaxe, Ionic Framework
Допомога із маркетингом	120 клієнтів	1 рік	Google Ads
Технічна підтримка	200 клієнтів	1 день	-
Послуги веб-дизайнера	10 клієнтів	2 тижні	Adobe illustrator
...	...	...	...

Як видно із прикладів наданих вище всі ці компанії мають різні функції, і кожна з цих функцій може мати різні параметри, проте це може бути і навпаки, є багато компаній із однаковими функціями і подібними параметрами, можна сказати що всі дані будуть усередненими. То ж в більшості випадків буде варіант знайти аналогічну компанію, або ж кілька компаній які є подібні, і використати їх онтології для вирішення власних задач.

У сучасному світі людство зіткнулося з великою кількістю даних, які неможливо опрацювати вручну. Таким чином, з'явилася необхідність у використанні спеціальних інтелектуальних систем, які забезпечують аналіз інформації та розробку альтернативних варіантів рішень. На даний момент більш ніж 70% з 500 найбільших компаній світу використовують засоби штучного інтелекту та інженерії знань як частину їх систем підтримки прийняття рішень [78].

Використання онтологій як бази для систем підтримки прийняття рішень має свої переваги. Якщо в минулому більшість систем були закритими, то в сьогоденні

онтології стали відкритими даними, доступними в інтернеті. Це означає, що інформація може бути доступною для всіх бажаючих.

Оскільки дані знаходяться у вільному доступі, можливо створювати системи, які агрегують інформацію з різних веб-сервісів та створюють все більші та складніші бази.

Розглядаючи нові підходи до побудови онтологій, з'являється можливість повторного використання існуючих баз знань. З метою розв'язання задач, пов'язаних зі системами підтримки прийняття рішень, вже було розроблено багато програмних систем. В окремих галузях, таких як управління, наявні спеціалізовані програмні засоби для створення систем підтримки прийняття рішень. Однак для розв'язання більш специфічних проблем, необхідне використання не одного алгоритму, а комбінації кількох.

В таких ситуаціях необхідно створити базову систему підтримки прийняття рішень, що працюватиме з онтологічними мережами. Ця система буде складатися з базового проекту з готовим пакетом бази знань, що представлені в онтологіях. При подальшій роботі над інтеграцією системи для розв'язання більш складних задач, необхідно буде вказати посилання на онтології, які пов'язані з конкретною задачею.

Щоб побудувати систему підтримки прийняття рішень та модель управління потрібна інформація. Однак, вже поточної статистично кількісної інформації, переважно, є недостатньо. Також, досвіду завжди потрібно ще більше. Основним джерелом інформації є експерти (керівники, користувачі, тощо). Експертам є легше надати дані в неформалізованому вигляді або ж на вербальному рівні. І якраз цим вимогам відповідає розроблений Сааті метод - метод аналізу ієрархій. А саме, це підтримка прийняття багатоцільових багатокритеріальних рішень при виборі однієї з багатьох альтернатив.

Для вирішення питань, в яких потрібно враховувати критерії та альтернативи, при прийнятті рішення необхідно формалізувати задачу і врахувати значення оцінки

для вибору рішення. Наприклад, якщо невелика компанія підходить до заключення використати в допомогу одну з аутсорсингових фірм. І в цьому випадку є можливість вибору фірми з урахуванням множини критеріїв та альтернатив, наприклад, досвіду, кваліфікації та портфолію. Критеріями також можуть виступати оцінка вартості запрошених послуг, оцінка навчання персоналу, оцінка термінів виконання завдання, тобто ряд критеріїв з яких є якісні. Для вирішення таких задач можна скористатися методом аналізу ієрархій.

Метод аналізу ієрархій (MAI) є математичним інструментом системного підходу до вирішення складних проблем прийняття рішень. Основне застосування методу полягає в підтримці процесу прийняття рішень шляхом ієрархічної композиції завдання та рейтингування альтернативних рішень [79].

Розв'язання проблеми вибору багатоцільового багатокритеріального рішення (стратегії) розглядається як процес поетапного встановлення пріоритетів цілей і критеріїв. Цей процес включає в себе вміння спостерігати та аналізувати результати спостережень, встановлювати взаємозв'язки між спостереженнями, оцінювати ступені щільності цих взаємозв'язків та синтезувати їх у загальне сприйняття. Принципи ідентичності та декомпозиції, дискримінації, порівняльного судження та синтезування є основою MAI.

Ієрархія є системою, в якій рівні розташовані та пронумеровані таким чином, що:

1. Нижній рівень містить рейтинговані альтернативи;
2. Вузли з вищих рівнів можуть домінувати тільки над вузлами нижчих рівнів.

Отже, в ієрархії зв'язки визначають шляхи однієї спрямованості - від вершини до альтернатив через проміжні рівні, які складаються з вузлів-критеріїв (рис. 3.5).

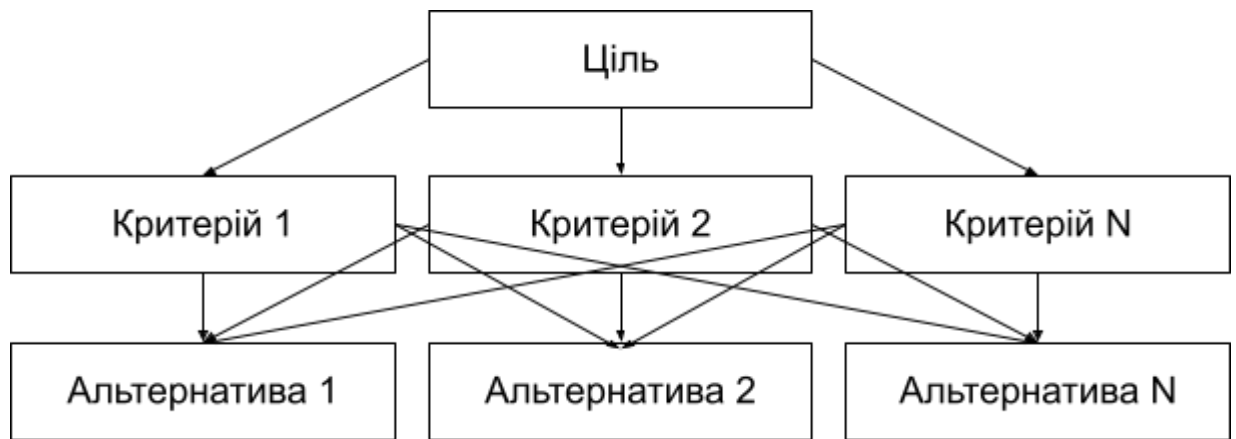


Рис. 3.5. Структура зв'язків ієрархії.

Основні етапи методу аналізу ієрархій:

1. Побудова ієрархічної структури задачі;
2. Визначення пріоритетів елементів ієрархічної структури;
3. Знаходження вектору пріоритетів;
4. Оцінювання узгодженості (однорідності) суджень експертів;
5. Ієрархічний синтез.

І так є компанія WeDo, ця компанія це біржа для фрілансерів і також агрегатор, у якої є дві основні категорії: прибирання та переїзд.

Таблиця. 3.8.

	Функції компанії
WeDo	Прибирання дому, прибирання офісу, переїзд з квартири, переїзд офісу, прибирання при переїзді, ...

Таблиця. 3.9.

	Рік заснування	Кількість співробітників	Дохід за рік	...
WeDo	2016	25 + 120	58.000 ГРН	...

		фрілансерів		
--	--	-------------	--	--

Перший етап полягає в тому щоб найти у сформованій мережі компаній, компанію яка є аналогічною до WeDo або ж множину подібних. Для цього потрібно порівняти функції та параметри компаній. Отож, функції компанії WeDo є в основному в категорії прибирання та переїзду. Далі у наявній онтологічній мережі є дві компанії які відповідають даним значенням, значить після першого етапу фільтрації результатом є компанія CleanToday із відповідністю у 50% та компанія MoveNow також із відповідністю 50%. Тепер розглянемо параметри компаній, із прикладів наведених вище, видно що ці дві компанії відповідають року заснування, кількості співробітників та іншим поданим параметрам. Як висновок, було відфільтровано до двох компаній: CleanToday і MoveNow. Тепер робиться вибірка онтологій цих компаній, та поєднуються у одній онтологічній мережі із відповідністю у 75%.

Наступний, другий етап, є в тому щоб із цієї великої онтологічної мережі зробити подальшу вибірку та виокремити онтології які мають найбільшу відповідність питанню. Для цього починається робота із питанням, потрібно розбити його на компоненти: функції та параметри, і чим детальніше це зробити тим відповідь буде кориснішою. Отож, нехай є таке перше питання:

- Скільки потрібно машин для переїзду з офісу? (Мінімальна відповідність для питання 50%.)

Так, питання сильно загальне, але це допоможе краще зрозуміти ідею. Перш за все, знаходимо ключові слова у питанні, для наданого вони будуть: скільки, машин, переїзд та офіс. Тепер побудуємо можливі функції цього питання, базова система згенерує в основному такі: переїзд, переїзд з офісу, оренда машин для переїзду. Далі

йдуть параметри, для цього також використовуються ключові слова, то ж параметрами в даному випадку можуть бути:

Таблиця. 3.10.

Тип роботи	Тип транспорту	Тип будівлі	...
Переїзд / Оренда	Машина	Офіс	...

Використовуючи дані які були отримані із питання, зробимо порівняльний аналіз із функціями побудованої онтологічної мережі вище. В даному випадку найбільший процент відповідності мають дві функції:

- Переїзд з офісу;
- Оренда машин для переїзду.

Далі порівняння піде по параметрах питання та параметрах функцій, і в результаті видно що дві функції відповідають питанню, приблизно із 80% ймовірністю для функції переїзд з офісу та 20% для функції оренди машин для переїзду. Проводиться вибірка по відфільтрованих онтологіях та створюється нова онтологічна мережа. А тепер СППР використовуючи цю онтологічну мережу видасть наступний результат:

- Із ймовірністю у 80% для переїзду з офісу буде потрібно 6 машин.

Також потрібно додати, що це не означає що друга відповідь, а саме: Із ймовірністю у 20% для оренди машин для переїзду буде потрібна 1 машина, є неправильною, просто вона недостатньо відповідає заданому питанню, враховуючи що мінімальна відповідність була виставлена на 50%.

Розглянемо другий, більш детальніший приклад. Візьмемо вже вище описану компанію яка орієнтується на допомогу з переїздом.

Таблиця. 3.11.

	Функції компанії
MoveNow	Переїзд з квартири, переїзд офісу, переїзд магазину, прибирання при переїзді, оренда машин для переїзду, надання складу, ...

Таблиця. 3.12.

	Рік заснування	Кількість співробітників	Дохід за рік	...
MoveNow	2014	30	42.000 ГРН	...

Для того щоб визначити кількість транспортних засобів для переїзду з офісу варто спочатку оцінити кількість вантажу. Це все надзвичайно індивідуально, тому що навіть в однаковому за площиною офісі може знаходитись різна кількість вантажу.

Вибір способу перевезення вантажу є одним з важливих завдань, які вирішують вантажовласники. При цьому альтернативні варіанти способу перевезення мають три самостійних, але взаємопов'язаних, напрямки - вибір виду або видів транспорту, вибір певного транспортного засобу певного виду (видів), вибір перевізника. Загалом на вибір способу перевезення вантажу впливає ряд об'єктивних і суб'єктивних умов. Усі ці процедури виконуються на основі одного або системи критеріїв з урахуванням обмежень. Ці обмеження обумовлені або цільовими функціями логістичної стратегії компанії, або факторами навколишнього макро- і мікроекономічного середовища. Наприклад, у системі розподілу можуть бути

встановлені обмеження щодо термінів доставки готової продукції, витрат на транспортування, збереження вантажу, розташування ланки логістичної системи в якій вантаж зберігається або перевантажується на інший вид транспорту, тощо.

Спосіб транспортування, вид транспорту та логістичні посередники вибираються на основі системи критеріїв. До основних критеріїв вибору способу транспортування та виду транспорту відносяться:

- Мінімум транспортних витрат;
- Визначений час транзиту (доставки вантажу);
- Максимальна надійність і безпека;
- Мінімальні витрати (пошкодження), пов'язані з товаром у дорозі;
- Місткість і доступність виду транспорту;
- Диференціація продукції.

Сторони, які беруть участь у плануванні транспортування, повинні надати опис вантажу, включаючи такі деталі, як обмеження щодо орієнтації та штабелювання, розміри, що охоплюють, розташування центру ваги та маса вантажу. Оператори також повинні переконатися, що небезпечний вантаж супроводжується супровідною документацією, яка підписана та заповнена. Небезпечні предмети повинні бути відповідно марковані, упаковані та класифіковані.

Це все показує наскільки трудомістким може бути порівняння і скільки функцій та параметрів може бути включено в опрацювання. Однак, рухаючись поетапно, як було описано у попередньому прикладі, і з достатньою кількістю даних в результаті буде виведено більш точніша, а значить і корисніша відповідь на задане питання.

3.2. Метод вибору провайдера знань на основі модифікованого методу TOPSIS

Сьогодні одним з головних напрямів досліджень в галузі штучного інтелекту є розробка та впровадження автономних інтелектуальних систем. Такі системи дозволяють вивчити різні аспекти інтелектуальних систем з ситуаційною обізнаністю, починаючи з отримання та інтерпретації інформації з зовнішнього середовища, та закінчуючи прийняттям рішень, аналізом результатів їх виконання, навчанням та оновленням моделей на основі результатів дій.

Результати наукових досліджень в області штучного інтелекту, розпізнавання образів, логічного виводу та інших видів міркування застосовуються у системах взаємодіючих інтелектуальних агентів. Кожен інтелектуальний агент має власну концептуалізацію світу, яку він постійно оновлює та узгоджує за результатами своєї діяльності. Ця концептуалізація відображає досвід, особливості середовища, в якому функціонує агент, та завдання, які він виконує. Концептуалізація знань агента формально подана його онтологією, яка містить як явні, так і неявні знання та припущення. Концептуалізація використовується агентом для інтерпретації даних, моделей та знань, що зберігаються в базі знань цього агента.

Процес прийняття рішень інтелектуальним агентом складається з наступних етапів [80]:

- Ідентифікація та концептуалізація проблемної ситуації та визначення самої проблеми.
- Побудова моделі прийняття рішень, що включає цілі, критерії та альтернативи.
- Оцінювання альтернатив, вибір та валідація обраного рішення.
- Реалізація обраного рішення.
- Аналіз результатів та оновлення знань агента на основі цього аналізу.

У застосуванні формальних концептуалізацій для прийняття рішень важливою проблемою є залежність релевантної концептуалізації від проблемної ситуації та

намірів агента. Наприклад, транспортний засіб може мати різні концептуалізації для вирішення задач планування маршрутів його руху, страхування та технічної діагностики. Гнучкий підхід до динамічного вибору та застосування онтологій потрібен через контекстну залежність релевантних концептуалізацій. Агент вибирає онтологію з необхідними концептами, відношеннями та аксіомами, що є релевантними для опису проблемної ситуації, аналогічно до вибору типу моделі для її опису. Застосування концептуалізацій як гнучкого засобу передбачає використання відносно невеликих онтологій, між якими встановлено відношення відображення та артикуляції. Це дозволяє гнучкий перехід між онтологіями та повторне використання результатів моделювання при вирішенні задач прийняття рішень.

У дослідженні [52] зазначено, що великі онтології з різною предметною областю рідко використовуються як єдине ціле. Автори запропонували методологію OntoMove для виділення частини онтології відповідно до конкретної ситуації з установленням відображень між первинною та виділеною онтологією. Така ж ідея про неможливість створення глобальної цілісної онтології висловлена в іншому дослідженні [81]. Використовуючи алгебраїчний підхід до подання онтологій, автори формально визначають правила артикуляції між окремими елементами онтології. У статті [82] надано визначення онтологічної мережі як набору онтологій, які пов'язані відношеннями відображення, модуляризації та версійності. Ця стаття присвячена розв'язанню задачі формального подання відношень між онтологіями в онтологічній мережі.

Онтологічна мережа - це множина онтологій, які мають відношення відповідності. У книзі [83] вони інтерпретуються як графові моделі, які вказують на відношення та взаємозалежності між класами (концептами), властивостями та ролями, які є елементами онтологій. Викладені результати теоретичного аналізу підтверджені прикладами відображення онтологій. У дослідженні [84] онтологічні мережі використовуються як інструмент для спільного зберігання та обміну даними

між онтологіями, що базуються на різних первинних припущеннях. Автори вказують на недостатність розроблених методів узгодження онтологій. Як рішення цієї проблеми, вони запропонували мета-модель для узгодження елементів різних онтологій. Загалом, аналіз літературних джерел підтверджує важливість та актуальність розробки гнучкого підходу до вибору концептуалізації проблемних ситуацій у задачах прийняття рішень в автономних системах.

Вибір концептуальної моделі знань, яка відповідає специфічній ситуації, є складним завданням, що має багаторівневий характер. Розв'язання задачі на одному рівні не може бути застосоване на інших рівнях. Щоб прийняти рішення, агентам часто потрібна додаткова інформація від інших агентів. Однак, важливо врахувати не тільки наявність інформації у респондента, але й його концептуалізацію, яку він використовує для інтерпретації даних. На концептуалізацію агента впливає його середовище, задачі, які він вирішує, досвід та репутація. Усі ці фактори впливають на відповідність його концептуалізації потребам децидента. Вибір агента-респондента за його параметрами схожий на вибір експерта для консультації, де потрібно враховувати досвід, репутацію та коло проблем, з якими він працює.

На наступному рівні аналізують та узгоджують онтології агента децидента та агента-кореспондента як структурні моделі знань у контексті задачі, яку необхідно вирішити. На цьому етапі перевіряють, чи містить онтологія респондента необхідні концепти та відношення для подання концептуальної моделі задачі. На наступному рівні порівнюють патерни та моделі знань як частини онтології в обох агентах для встановлення відповідності між елементами моделей, які використовуються для опису проблемної ситуації. На останньому рівні вирішується задача вибору інтерпретації концептів та відношень у контексті прийняття рішення. Визначається інтерпретація концепту, яка має властивості та обмеження, необхідні для подання проблемної ситуації. Кожен рівень вирішення задач потребує окремих досліджень. У цій роботі запропоновано метод вибору агента-кореспондента з використанням

модифікованого методу TOPSIS (Technique of Order Preference Similarity to the Ideal Solution), в якому задача вибору розглядається як багатокритеріальна задача шляхом порівняння з ідеальним вирішенням [85].

У багатьох випадках інтелектуальний агент, що приймає рішення, потребує додаткової інформації для того, щоб побудувати модель проблемної ситуації. У цьому випадку він звертається до іншого агента, якого ми будемо називати провайдером знань. Як правило, при цьому використовуються механізми веб-сервісів. Водночас, інформація про провайдера, що подається у профайлі провайдера є досить обмеженою.

Важливою частиною процесу прийняття рішення є побудова концептуалізації проблемної області, яка точно відображає суть проблемної ситуації. Для формального подання такої концептуалізації використовуються онтології. На якість концептуалізації проблемної області впливає також досвід роботи агента, що створює цю концептуалізацію, середовище в якому він працює. Тому, ще перед узгодженням онтологій доцільно вибрати агентів що працюють у середовищі, яке найбільш наближене до інформаційних потреб агента-децидента.

Частковим випадком цієї задачі є вибір агента-кореспондента, що за профілем своєї діяльності найбільш подібний до агента-децидента.

Цю задачу можна подати як задачу багатокритеріального вибору. Набір критеріїв залежить від конкретної задачі прийняття рішення.

Нехай $SCr = (cr_1, cr_2, ..., cr_n)$ - вектор обраних критеріїв.

Децидент також визначає відносну важливість цих критеріїв, що відображено у векторі відповідних ваг:

$$SW = (cw_1, ..., cw_2, ..., cw_n) \quad (6)$$

Визначена множина альтернатив серед агентів-провайдерів, що можуть надати необхідну інформацію:

$$SAI = \{al_1, al_2, ..., al_m\} \quad (7)$$

Пропонується подати вимоги щодо агента-провайдера у вигляді вектора значень критеріїв, що відповідають вимогам та описують ідеальний варіант вибору як точку у багатовимірному просторі критеріїв:

$$SCv^0 = (cv_1, cv_2, ..., cv_n) \quad (8)$$

Де cv_k – це ідеальне значення для критерію cr_k .

Для вирішення цієї задачі можна використати модифікацію метода TOPSIS яка дозволить знайти альтернативу у SAI що лежить найближче до ідеального значення, поданого вектором SCv^0 .

1. На першому кроці цього методу побудуємо $m \times n$ матрицю кожний рядок якої є значеннями критеріїв для всіх альтернатив:

$$Mcv = \begin{pmatrix} cv_{11} & ... & cv_{1n} & ... & ... & ... & cv_{m1} & ... & cv_{mn} \end{pmatrix} \quad (9)$$

2. Додамо до цієї матриці останнім рядком вектор SCv^0 , отримуючи матрицю $MCvi$.
3. Нормалізуємо цю матрицю по кожному критерію, отримавши нормалізовану матрицю MCv' .

4. На наступному кроці, на відміну від класичного методу TOPSIS, де ми визначаємо максимальні та мінімальні значення по кожному критерію, визначимо відстань для кожного значення критерію по альтернативах від ідеального значення. Для кожного числа cv_{ij} визначимо різницю

$d_{ij} = |cv_{ij} - cv_j^0|$. Для підрахунку відстані можна також скористатися

формулою для Евклідової відстані: $d_{ij} = \sqrt{(cv_{ij} - cv_j^0)^2}$.

5. В результаті отримаємо матрицю відстаней від ідеального значення для всіх альтернатив та критеріїв $MD = (d_{ij})$.

6. Виключимо в матриці відстаней нульовий рядок, що відповідає ідеальному випадку.
7. Для кожної альтернативи al_j підрахуємо зважену суму відстаней по формулі

$$L_j = \sum_{i=1}^n (cw_i * d_{ij}) \quad (10)$$

8. Та визначимо альтернативу, для якої значення L_j мінімально.

У використанні онтологій для роботи з предметними областями відбувається зміна підходу з фіксованих на гнучкі, контекстно-залежні концептуалізації. Замість однієї великої онтології, яку використовують усі інтелектуальні агенти для забезпечення семантичної інтероперабельності, застосовується мережа онтологій, що відображають досвід окремих агентів або відповідають конкретним задачам. Такий підхід дає змогу інтелектуальним агентам використовувати різні онтології для моделювання однієї ситуації з метою кращого розуміння. Семантична інтероперабельність забезпечується наявністю відображень між онтологіями.

Також, одним з варіантів комунікації між агентом-децидентом та агентом-кореспондентом є запит знань про спосіб вирішення проблеми, виконання завдання. Це знання подається у формі концептуальної моделі, що відображає послідовність етапів виконання завдання, з врахуванням визначеного контексту та обмежень. Припустимо, що перед початком такої комунікації агенти узгодили свої онтології, по меншій мірі у частині, необхідній для подання концептуальної моделі виконання завдання.

Для отримання необхідної моделі операції агент-децидент повинен сформулювати та передати агенту-кореспонденту специфікацію свого наміру. Така специфікація має відображати що має бути зроблено, не вказуючи як це зробити. У загальному випадку агент-децидент подає свій намір як бажану зміну, вказуючи початковий стан предметної області, результуючий стан, вимоги та обмеження щодо

процесу переходу системи з початкового у результуючий стан. Додатково, він надає тести, що перевіряють ступінь успішності досягнення результуючого стану. Для специфікації намірів агента-децидента у загальному випадку пропонується використати функціональну модель IDEF0.

Перевагою такого подання наміру є можливість специфікувати не тільки зміну стану предметної області, але й контекст такої зміни, поданий набором обмежень та вимог. Важливим є також надання додаткової інформації у вигляді тестів, критеріїв якості тощо.

Назва функції передає тільки загальну ідею щодо потрібної зміни у предметній області. Спосіб досягнення цієї зміни залежить від початкового стану (передумов), результуючого стану, обмежень. Формально, такий намір можна подати як концептуальну модель виконання завдання: $Tsk = (Init, Tn, Out, Cons, Tst)$, де Tsk – специфікація завдання, $Init$ – початковий стан, Tn – назва функції, що описує потрібну зміну, Out – результуючий стан, $Cons$ - обмеження, Tst – набір тестів для перевірки успішності виконання завдання.

Назва функції використовується агентом-кореспондентом для пошуку потрібної моделі у репозиторії моделей. Ця назва формується на основі узгодженого змісту концептів та відношень онтології. Є декілька підходів які можна використати для такого пошуку:

1. Пошук за критерієм мінімальної лінгвістичної відстані. З репозиторію обирають функції з назвами, що мають мінімальну відстань з Tn . При цьому враховують синоніми.
2. Використання класифікаторів функцій, які подають (найчастіше це бізнес-функції) у вигляді ієрархій назв. Такі класифікації формують спеціалізації функції, що дає змогу точніше відобразити операційний контекст. Прикладами таких класифікацій є [86, 87].

Після вибору можливих кандидатів з моделей у репозиторії агент-кореспондент перевіряє чи може бути задача вирішена з використанням моделі-кандидата з урахуванням початкового стану та вимог визначених у запиті. При успішному результаті, концептуальна модель виконання завдання надсилається агенту-дециденту.

3.3. Модель подання контекстно залежного змісту концептів онтології на основі теорії прототипів

У традиційному формулюванні онтології концепти подають як фіксовані та незмінні вузли у таксономії концептів. Такий підхід має переваги стабільності, ясності, можливості повторного використання знань, що використовують ці концепти, а також – спільного використання знань інтелектуальними агентами без додаткових узгоджень.

Водночас, на практиці (що знаходить відображення і у використанні концептів у природній мові) зміст концепту залежить від ситуації, в якій він використовується. Зміст концепту гнучко змінюється і особа, що його використовує, динамічно та неявно обирає зміст концепту, що релевантний до наявної ситуації [88, 89]. З бігом часу кожен концепт накопичує різноманітні інтерпретації, стає багатозначним, і містить багато змістовних нюансів.

Багатозначність - це властивість терміна або поняття мати кілька значень або інтерпретацій. Багатозначність може створювати проблеми при обробці інформації, оскільки різні користувачі можуть інтерпретувати терміни по-різному, що може призвести до непорозумінь та нечіткостей. Це відображено у багатстві способів опису світу природною мовою, використанні метафор, асоціацій, порівнянь.

З бігом часу розуміння та зміст концепту змінюється. Особа будує своє розуміння змісту концепту спостерігаючи об'єкти та явища у навколишньому світі, групуючи та класифікуючи їх. Об'єкти, що представляють приклади концепту

стають прототипами, до яких звертаються для розуміння змісту концепту. Пізніше подібні до первинного зразка об'єкти об'єднують у групи, використовуючи ту ж назву концепту. При цьому зміст концепту збагачується та стає багатозначним. Він починає відображати різні ситуації використання концепту. Процес створення та збагачення змісту концепту у когнітивній діяльності людини описано у теорії прототипів [90, 91]. Підхід теорії прототипів також застосовують [92] для моделювання нечіткостей та багатозначностей для систем зі штучним інтелектом. Процес еволюції змісту концепту в інтелектуального агента ніколи не припиняється, він постійно збагачується в процесі навчання, під час виконання практичних завдань.

Традиційні, фіксовані, не контекстні, онтології не відображають природної нечіткості та багатозначності концептів, залежності змісту від ситуації та властивостей агента, що його використовує. Також традиційний підхід не підтримує постійного навчання та еволюції змісту концепту що відбувається у процесі навчання.

У роботі використовується інший підхід до подання концепту та побудови контекстно-залежних онтологій в якому кожен концепт з одного боку зберігає свою багатозначність та нечіткість, а з іншого – дає змогу визначити чіткий зміст концепту якщо ідентифікована ситуація, в якій він використовується.

В основу цього підходу покладено теорію прототипів [91] що використовується також для моделювання нечіткостей у змісті концептів [92]. Подамо зміст концепту як дерево пов'язаних змістів, що постійно змінюється. Нечіткий зміст подається контекстними залежностями атрибутів, обмежень, відношень та аксіом.

У процесі навчання агента у межах концепту формуються кластери подібних змістів, що мають спільні властивості. Вони виступають носіями спільних властивостей, обмежень та аксіом. Вони також приймають участь у

контекстно-залежних відношеннях. Такі кластери виступають гілками дерева змістів. У процесі навчання відбувається формування та зміна цих кластерів. Збереження різних змістів концепту у вигляді дерева подібно до збереження різних версій документа у системі контролю версій - нові версії можна будувати на основі існуючих, але старі версії також завжди доступні до використання.

Концептуалізація контекстно-залежних інтерпретацій контексту розглядається у науковому напрямі контекстних онтологій. У ньому розрізняють локальні та спільні концептуалізації. Локальні концептуалізації зберігають у пам'яті інтелектуального агента та відображають у спільну концептуалізацію, коли виникає потреба повторного використання знань, або комунікації між агентами [93]. Поняття контексту визначено як конкретна локальна концептуалізація.

У [94] введено поняття моно-контекстної онтології (Mono-Context ontology) та багато контекстної онтології (Multi-Context ontology). У багато контекстній онтології кожен концепт може мати декілька локальних інтерпретацій.

Для опрацювання різних контекстів та логічного виводу на основі Описової логіки, було розроблено розширення мови OWL – OWL-C [95]. Ця мова базується на синтаксисі OWL та надає правила відображення (bridge rules) для пов'язання класів, об'єктів та ролей на синтаксичному та семантичному рівнях.

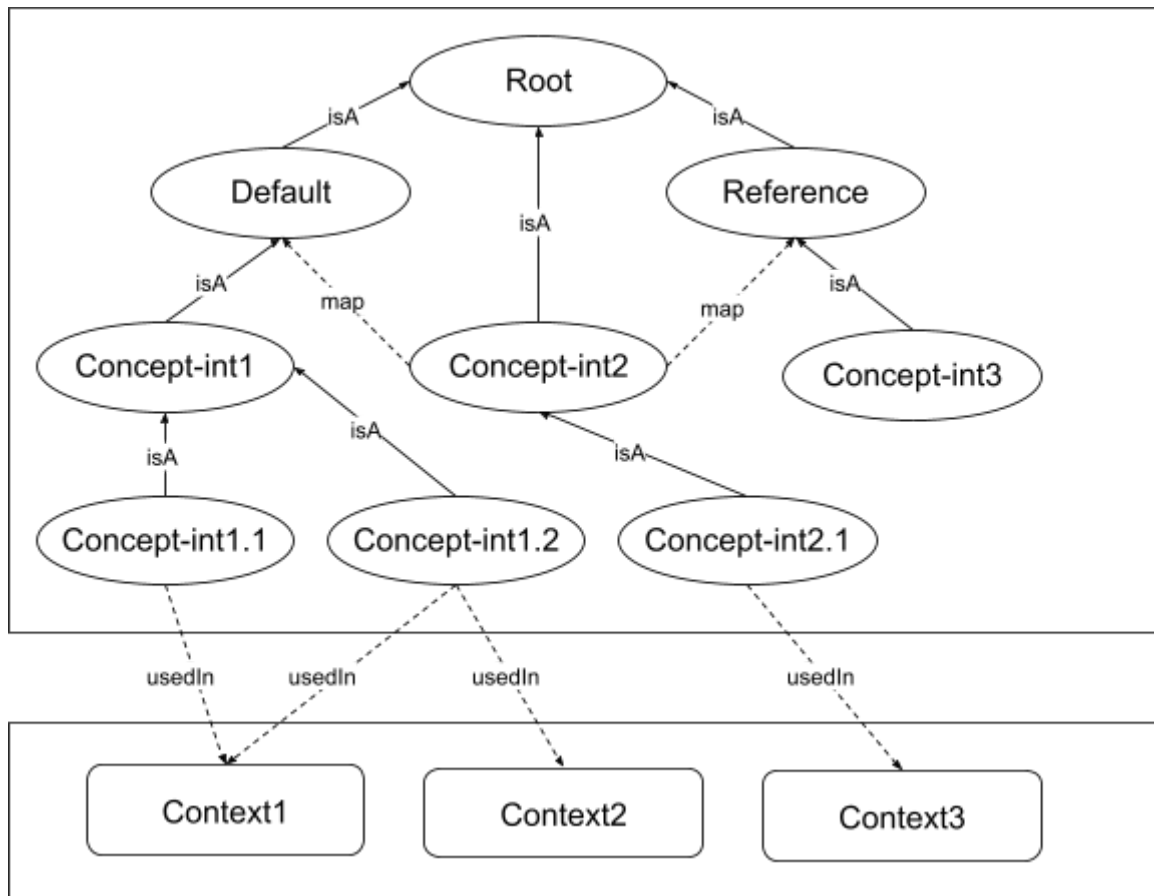


Рис. 3.4. Організація значень концепту в локальній онтології.

Метою підходу є зробити зміст концепта гнучким та залежним від специфікації проблемної ситуації. Подамо цей зміст у вигляді дерева, в якому корінь містить спільні властивості та іншу інформацію для всіх об'єктів дерева. Кореневий концепт має назву, яка видима та використовується іншими агентами для узгодження змісту та доступу.

Листями дерева виступають прототипи. Вони містять відмінну частину інтерпретації концепту, залежну від конкретної ситуації. Вихідний зміст концепту формується як об'єднання визначень на шляху з обраного прототипу до кореня дерева. Гілки дерева використовують для виокремлення властивостей спільних для декількох прототипів.

Традиційне визначення онтології подано кортежем з множинами концептів SCn , відношень SRI та аксіом SA :

$$On = (SCn, SRL, SA) \quad (11)$$

У формальній моделі контекстно-залежної онтології відношення та аксіоми є також контекстно-залежними та прив'язаними до конкретних вузлів дерева. Контекстно-залежне визначення концепту формується як об'єднання атрибутів та обмежень на шляху Pt_i від обраного прототипу Pro_i до кореневого концепту Rt :

$$Cn_i = Pro_i \cup Br_i^{su} \cup Rt \quad (12)$$

$$Br_i^{su} = \bigcup_{j \in Pt_i} Br_j \quad (13)$$

Кожен прототип Pro_i подається як кортеж:

$$Pro_i = (Pre_i, Par_i, SAtt_i, SCons_i, SRL_i, SA_i) \quad (14)$$

Де Pre_i це специфікація передумов використання прототипу Pro_i , Par_i – вказівник на батьківську гілку дерева, $SAtt_i$ – множина атрибутів, $SCons_i$ – множина обмежень, SRL_i – множина відношень які діють для даної інтерпретації концепту, SA_i – множина аксіом.

Множина передумов Pre_i може бути чіткою або нечіткою для моделювання можливої нечіткості при виборі інтерпретації конкретної задачі прийняття рішення.

Важливою відмінністю традиційного підходу контекстних онтологій та запропоновим підходом є те, що традиційний підхід підкреслює необхідність узгодження локальної онтології зі спільною онтологією. У запропонованому підході багатозначного подання змісту концепту увага зосереджена на виборі змісту концепту релевантного до проблемної ситуації. Задача узгодження зі спільною онтологією тут не стоїть, адже рішення приймається агентом самостійно, на основі моделей локальної бази знань. У випадку, коли потрібно отримати інформацію та

знання від інших інтелектуальних агентів, узгодження змісту відбувається з побудованим контекстом на основі обраного прототипу. Іншою перевагою підходу, що використовує прототипи є те, що модель концепту є гнучкою та дозволяє агенту проводити експерименти, будувати та змінювати зміст концепт у процесі навчання.

Висновки до розділу 3

Використання онтологічних мереж у системах підтримки прийняття рішень є ефективним методом розв'язання проблемних ситуацій в умовах багатозначності. Розглянуто, запропоновано та розроблено методи та алгоритми їх побудови, проектування та реалізації архітектури таких систем, а також алгоритми функціонування окремих модулів цих систем. Це дозволяє створювати потужні та ефективні системи підтримки прийняття рішень, які допомагають вирішувати складні задачі в умовах багатозначності.

Результати розділу опубліковано у [1, 2, 4, 5, 6, 12].

РОЗДІЛ 4. Практична реалізація прототипу системи підтримки прийняття рішень на основі онтологічних мереж

4.1. Опис та структура програмного забезпечення

На основі методів із третього розділу був розроблений прототип програмного забезпечення для аналізу роботи підприємства.

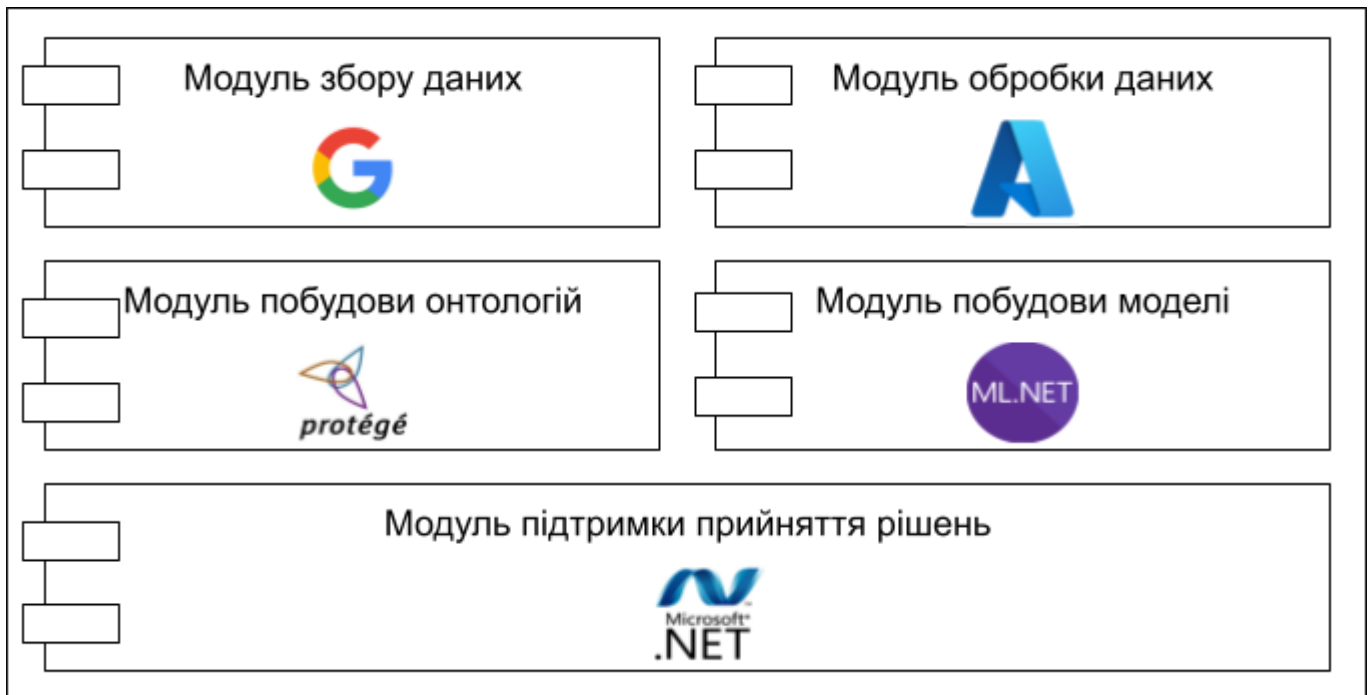


Рис. 4.1.1. Діаграма компонент ПЗ.

При розробці програмного забезпечення (ПЗ) було побудовано модель для штучного інтелекту (ШІ) з використанням онтологій для навчання цієї моделі та інтерфейс для користувача. Дані було зібрано з відкритих джерел використовуючи пошук в Google, а саме API Google Search. Зібрані дані було оброблено використовуючи методи обробки природної мови (Natural Language Processing - NLP) з допомогою Azure Cognitive Services. Онтології на основі оброблених даних було створено використовуючи Protégé. Модель була побудована з використанням ML.NET - це бібліотека машинного навчання, яка дозволяє розробляти моделі

машинного навчання для аналізу даних в .NET-платформі. Програмний інтерфейс було розроблено з використанням Windows Forms - це платформа інтерфейсу користувача для створення класичних додатків Windows.

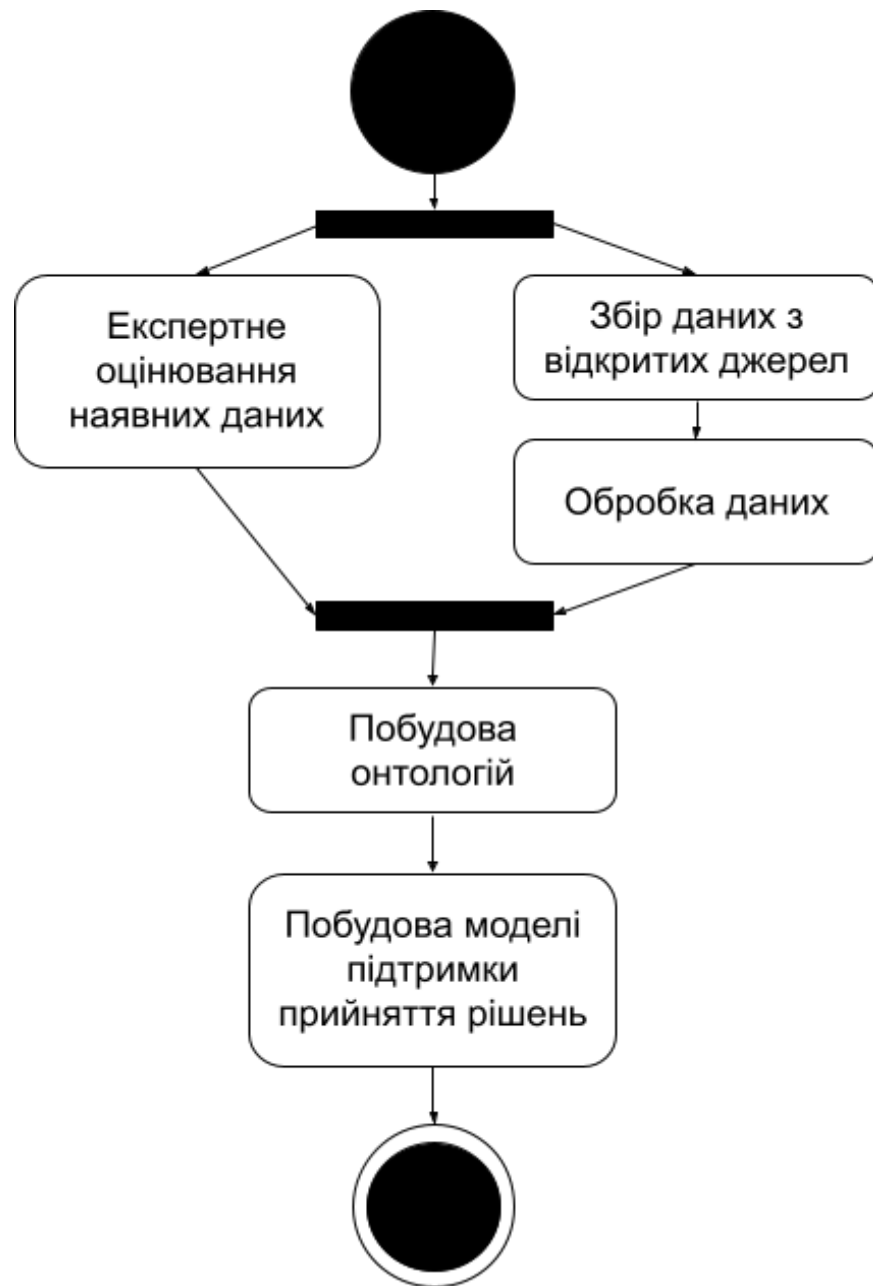


Рис. 4.1.2. Діаграма діяльності СПДР.

Модуль збору даних. Збирання даних з веб-сторінок за допомогою пошуку в Google може бути ефективним способом отримання великої кількості даних, для

цього було використано API Google Search. Однак, необхідно враховувати деякі фактори, щоб зробити цей процес ефективним і забезпечити якість даних. Для цього потрібно:

- Створити список ключових слів, які відповідають темі. Ці ключові слова можуть включати терміни, які пов'язані з темою, або запити, які можуть використовуватися для пошуку інформації з області.
- Використати фільтри Google, щоб забезпечити релевантні результати. Наприклад, фільтр за часом, щоб отримувати тільки недавні результати, або фільтр за типом файлу, щоб знайти тільки сторінки, які містять певний тип даних (наприклад, PDF-файли або картинки).
- Використати різні типи запитів, щоб отримати різноманітні результати. Наприклад, використовувати запити, що містять ключові слова у різному порядку, або запити зі схожими семантичними значеннями.
- Перевірити зібрані дані на валідність і правильність. Переконайтеся, що зібрані дані з довірених джерел і що вони точні і повні.
- Збирати дані з веб-сторінок відповідно до правил сайту та відповідних законів і правил використання даних.

Також, важливо зазначити, що збирання даних з веб-сторінок - це лише один із способів отримання даних. Інші можливі способи включають збір даних вручну чи використання існуючих баз даних.

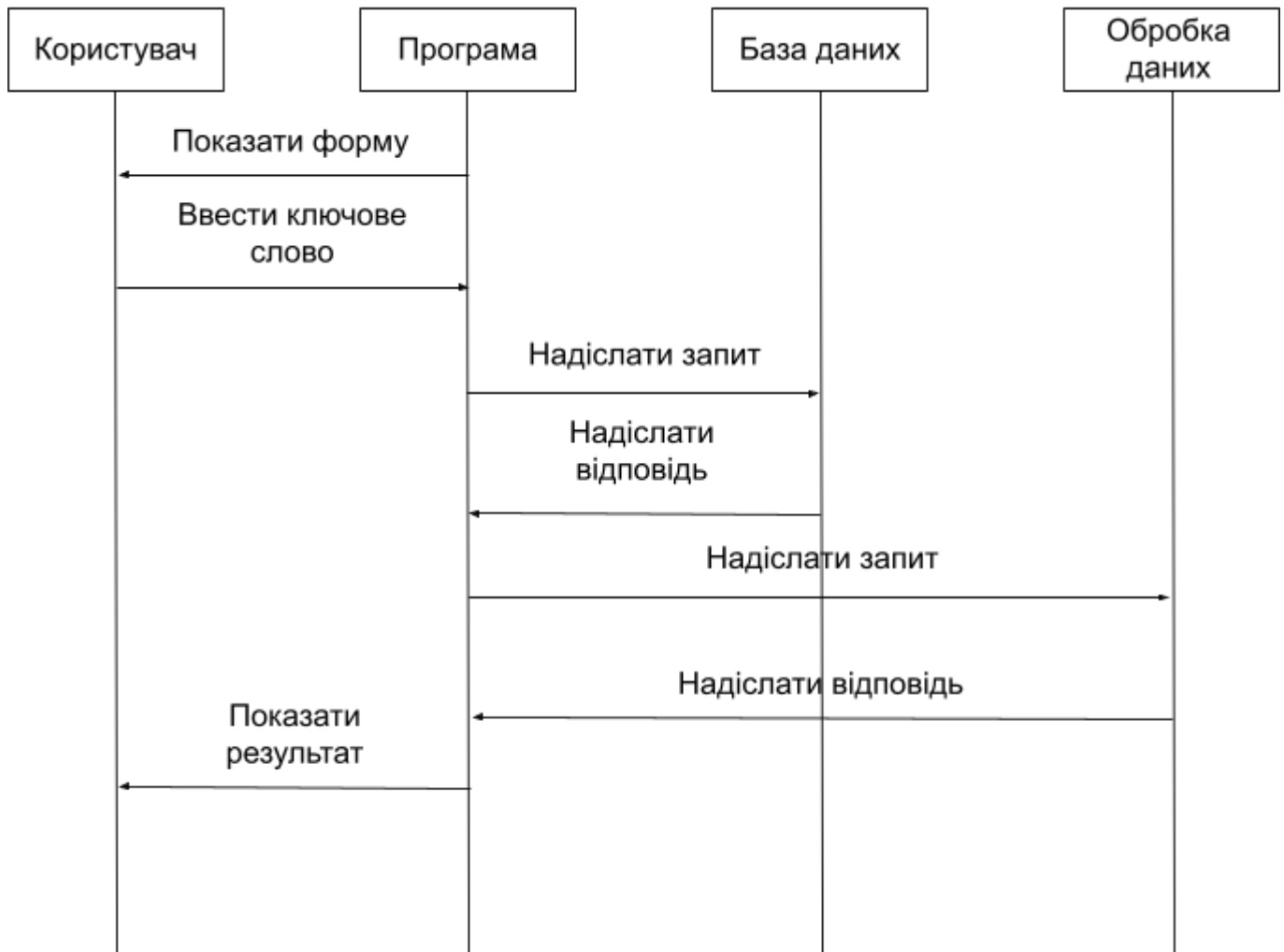


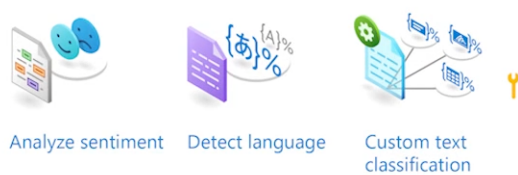
Рис. 4.1.3. Діаграма послідовності модуля збору та обробки даних.

Модуль обробки даних. Після збору даних з пошуку в Google, необхідно використовувати методи обробки природної мови, щоб перетворити неструктуровану інформацію в структурований формат, для цього було використано Azure Cognitive Services [96]. Для цього потрібно [97]:

- Розпізнавання іменованих сутностей (Named Entity Recognition - NER). Це дозволяє визначити іменовані сутності в тексті, такі як імена, назви організацій, місця, дати тощо. Це допоможе ідентифікувати ключові терміни, які потрібно включити в онтологію.

- Визначення залежностей між словами (Dependency Parsing). Це дозволяє визначити залежності між словами в тексті та зв'язки між ними. Це може допомогти встановити відношення між поняттями та створити його ієрархію.
- Кластеризація та тематичний аналіз. Це дозволяє групувати текстові дані в зв'язані групи та ідентифікувати ключові теми та терміни, які можуть бути використані в онтології.

Classify text



Extract information

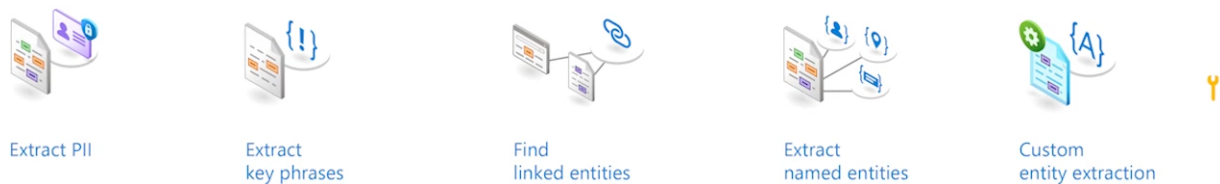


Рис. 4.1.4. Використані сервіси обробки даних від Azure.

Модуль побудови онтологій. Для створення онтологій було використано Protégé. Protégé є безкоштовною платформою для створення онтологій та відноситься до інструментів Semantic Web. Вона надає можливість створювати, редагувати та розширювати онтології, а також визначати правила логіки та відносин між об'єктами в онтології. Онтології створюються за допомогою графічного інтерфейсу, де можна визначати концепти, властивості, обмеження та ієрархії класів та інших об'єктів. Кожент пошуковий запит створює свою відповідну онтологію.

Після створення онтології потрібно виконати перевірку та оцінку її якості, можливий варіант повторного проходження всіх пунктів вище [98].

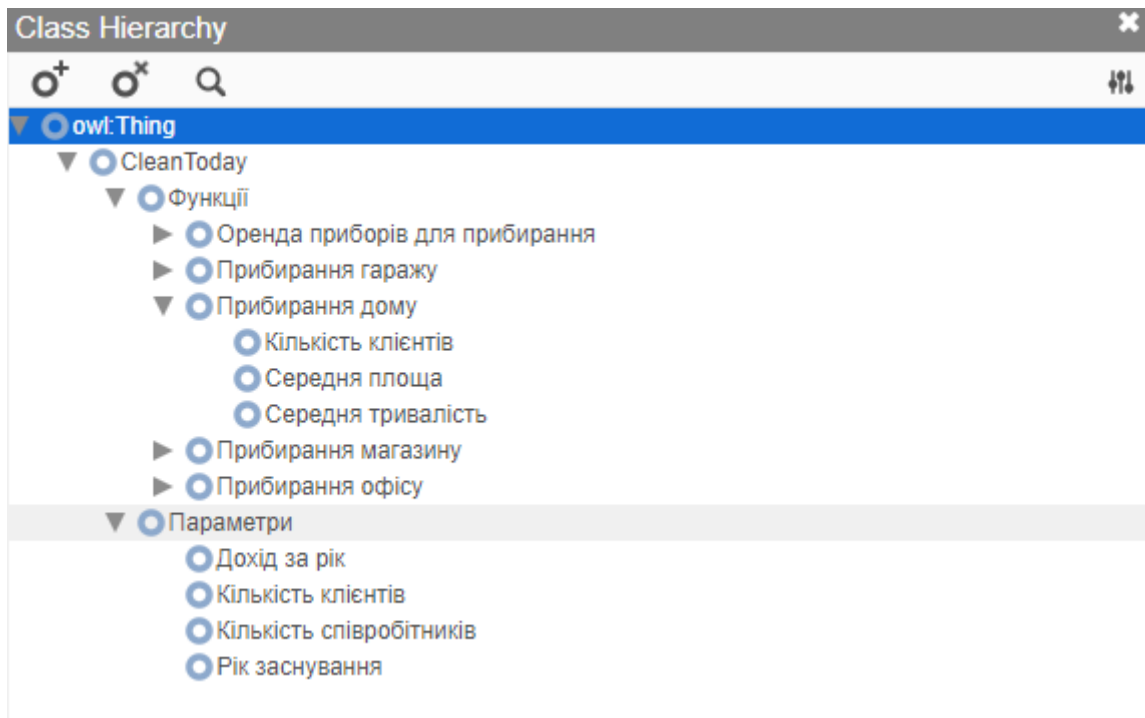


Рис. 4.1.5. Створення онтології за допомогою Protégé.

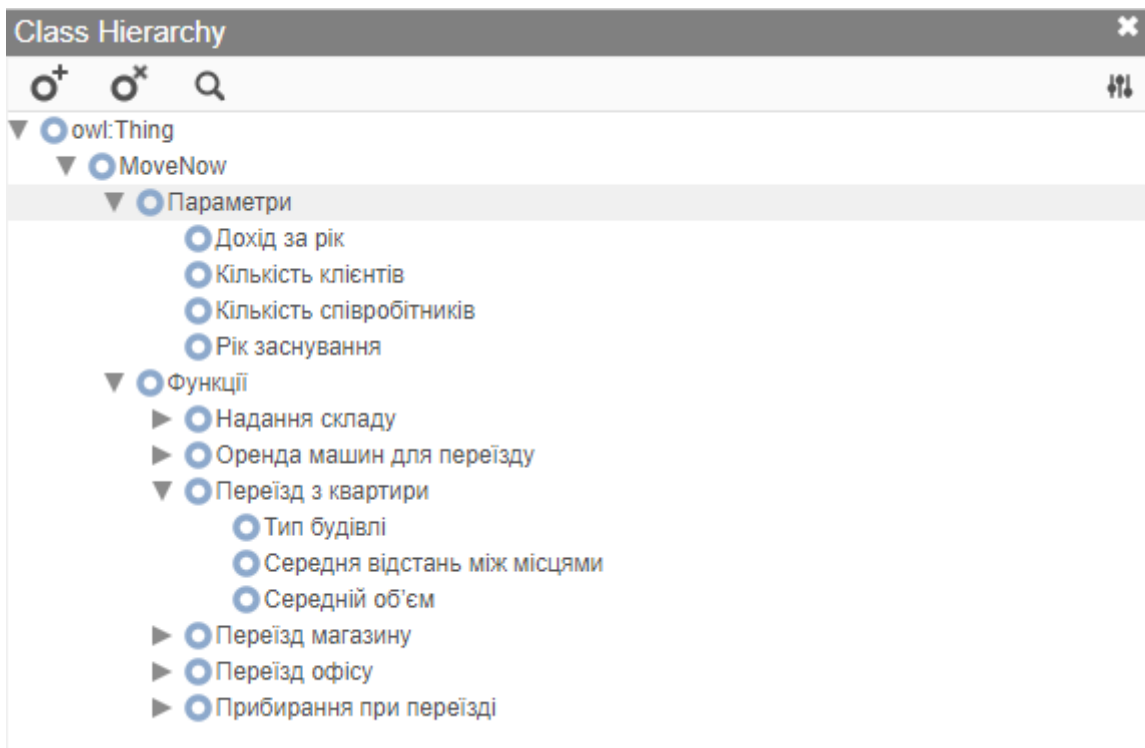


Рис. 4.1.6. Створення онтології за допомогою Protégé.

Онтології є ефективним інструментом для навчання штучного інтелекту, оскільки вони дозволяють створювати формальні моделі знань та відносин між ними. Навчання III ґрунтується на наданні системі доступу до великої кількості даних. Однак, для того, щоб система могла ефективно використовувати ці дані, їй потрібно мати розуміння про те, як вони пов'язані між собою, а також про те, які знання та контекст пов'язані з цими даними. Таким чином, застосування онтологій може допомогти системі зрозуміти ці знання та контекст і використовувати їх для розв'язання конкретних завдань. Що також може допомогти зменшити кількість необхідних даних для навчання, оскільки онтології дозволяють описувати знання та взаємозв'язки між ними.

Модуль побудови моделі. Модель була побудована з використанням ML.NET [99, 100, 101, 102]. ML.NET - це бібліотека машинного навчання, яка дозволяє розробляти моделі машинного навчання для аналізу даних в .NET-платформі. Тип задачі було визначено як регресія. Використано відповідний алгоритм - Support Vector Machine (SVM). Щоб навчити штучний інтелект з використанням ML.NET, потрібно навчити модель на основі підготовлених даних та обраного алгоритму. Для цього було використано клас MLContext та метод Fit(), який приймає вхідні дані та ознаки та видає навчену модель.

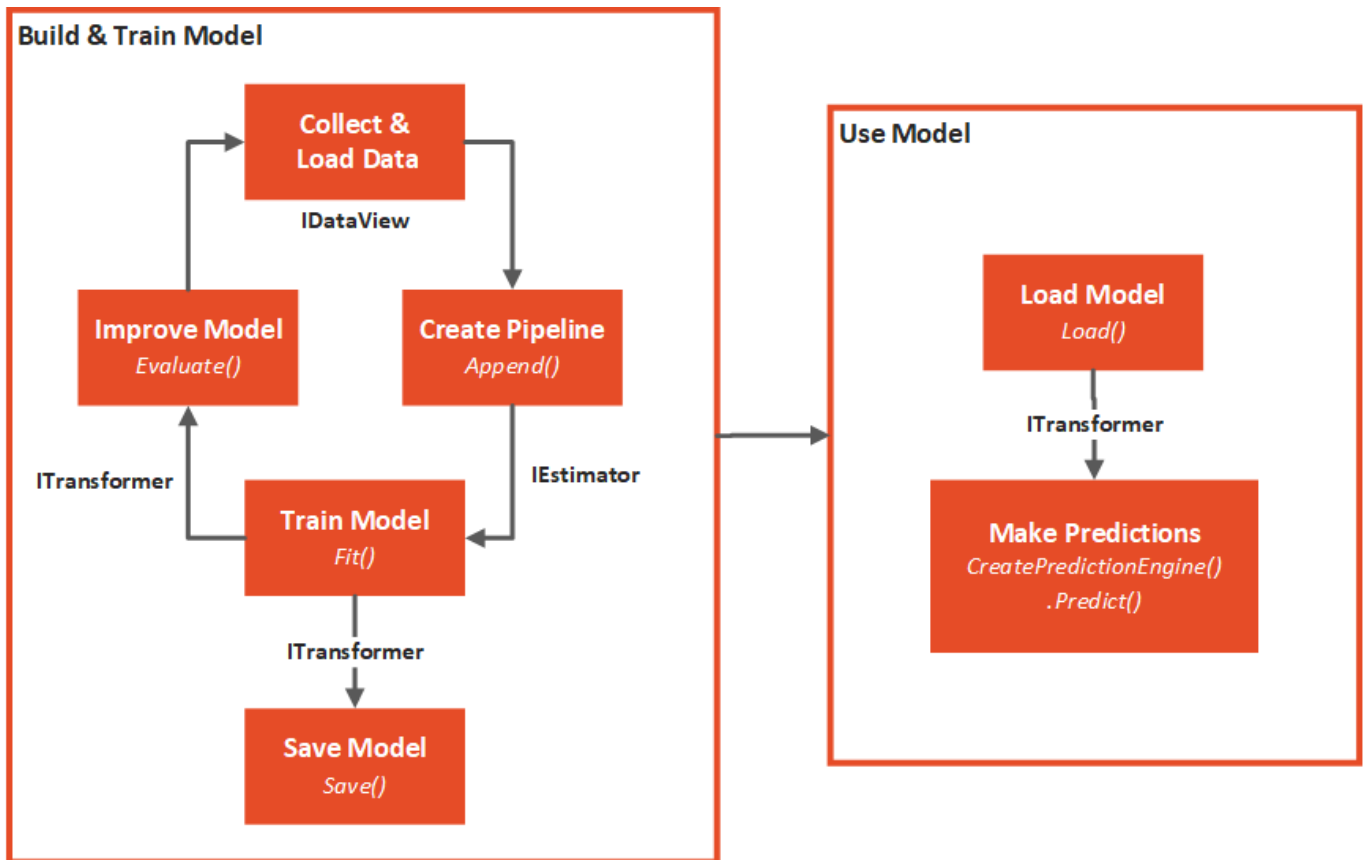


Рис. 4.1.7. Створення моделі за допомогою ML.NET.

Приклад використаного коду для навчання:

```
var context = new MLContext();

//Load the data.
var data = context.Data.LoadFromTextFile<Data>(
    path: "projects/quario/data.xml",
    hasHeader: true,
    separatorChar: '\t'
);

//Split the data into training and test sets.
```

```

var dataSplit = context.Data.TrainTestSplit(data, testFraction: 0.2);

//Define the training pipeline.
var pipeline = context.Transforms.Conversion.MapValueToKey("Label", "Label")
    .Append(context.Transforms.Text.FeaturizeText("Features", new
TextFeaturizingEstimator.Options
    {
        WordFeatureExtractor = new WordBagEstimator.Options { NgramLength = 1,
UseAllLengths = true },
        CharFeatureExtractor = new WordBagEstimator.Options { NgramLength = 3,
UseAllLengths = false }
    })
    .Append(context.Transforms.NormalizeLpNorm("Features", "Features"))
    .Append(context.Transforms.Conversion.MapKeyToValue("Label", "Label"));

//Train the model.
var model = pipeline.Fit(dataSplit.TrainSet);

//Evaluate the accuracy of the model on the test set.
var predictions = model.Transform(dataSplit.TestSet);
var metrics = context.MulticlassClassification.Evaluate(predictions, "Label");

Console.WriteLine($"Micro-accuracy: {metrics.MacroAccuracy}");
Console.WriteLine($"Macro-accuracy: {metrics.MacroAccuracy}");

```

Приклад використаного коду для відповіді на питання від користувача:

```

var predictionEngine = context.Model.CreatePredictionEngine<Data,
Prediction>(model);

var prediction = predictionEngine.Predict(new Data { Text = "Question" });

Console.WriteLine($"Answer: {prediction.Label}");

```

Модуль підтримки прийняття рішень. Програмний інтерфейс було розроблено з використанням Windows Forms [103]. Інтерфейс складається з двох компонент: поле вводу і блок виводу. А також, однієї функції як виконується при нажатті Enter на клавіатурі, що і викликає метод описаний вище після чого результат відображається в блоці виводу.

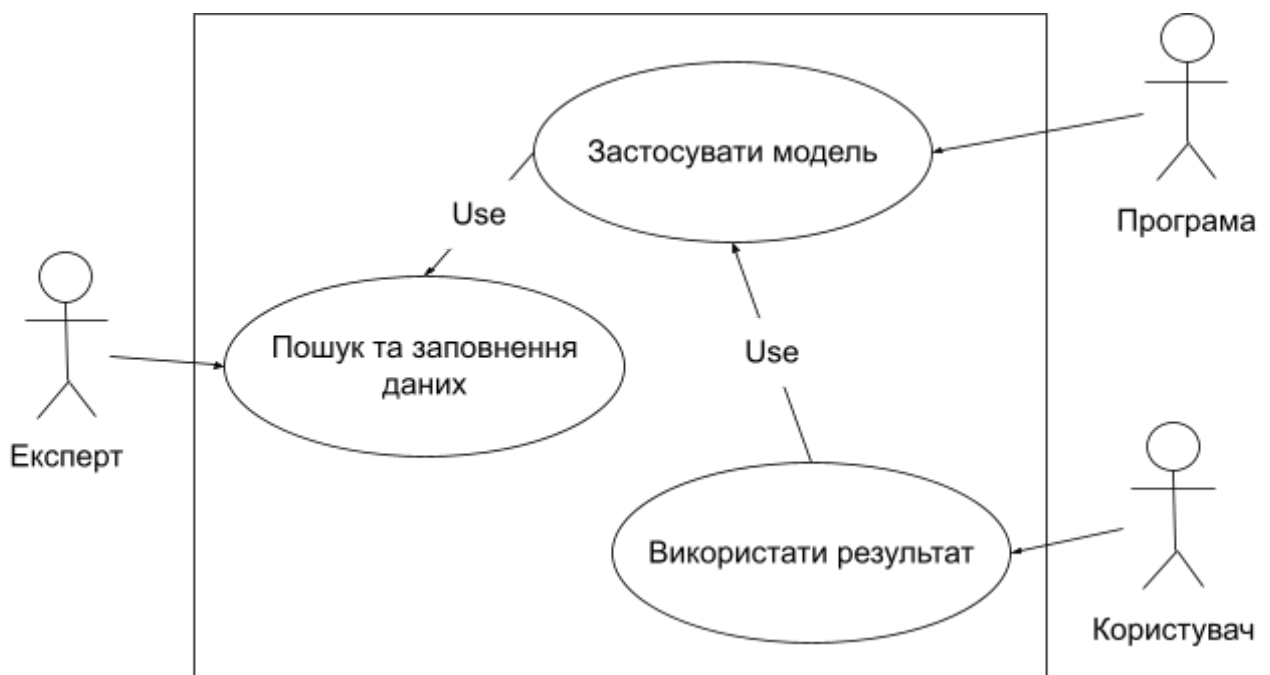


Рис. 4.1.8. Діаграма прецедентів модуля підтримки прийняття рішень.

4.2. Вхідні дані та очікуваний результат експерименту

Для користування необхідно:

1. Встановити пакет .NET Framework 4.7.2+ з офіційного сайту - <https://dotnet.microsoft.com/download/dotnet-framework> - по наданій інструкції.
2. Перейти у папку з програмою.
3. Запустити файл - Quarior.exe.

Потрібно зазначити що для роботи програми виконання має відбуватись на операційній системі Windows, а також комп'ютер підключений до інтернету. Інсталяції додаток не потребує. Робота з програмою тільки на англійській мові.

Навчання моделі було проведено на обмеженій вибірці запитів, що в основному належали до п'яти категорій: веб-розробка, мобільна розробка, сервіси прибирання, сервіси переїзду та сервіси реклами. Така обмежена вибірка запитів під час навчання моделі призводить до певних обмежень в її здатності розуміти і відповідати на запити, які не належать до цих категорій. Також, важливо враховувати, що навчання було здійснене на основі публічних даних з інтернету, і, як наслідок, це може відображати певні шаблони про те, які запити та відповіді є коректними.

Головна сторінка ПЗ містить два блоки, перший для вводу питання і другий для виведення відповіді (рис. 4.1). Щоб отримати результат, потрібно ввести питання у перший блок, приклади питань наведені на рис. 4.2. Після чого натиснути кнопку Ввід на клавіатурі. Далі ПЗ заблокує інтерфейс і почне аналізувати питання та формувати відповідь на це питання (рис. 4.3). Результати ПЗ буде виведено у другому блоці, у форматі: питання - відповідь (рис. 4.4). Після чого поле для вводу питання буде розблоковано для подальшої роботи.

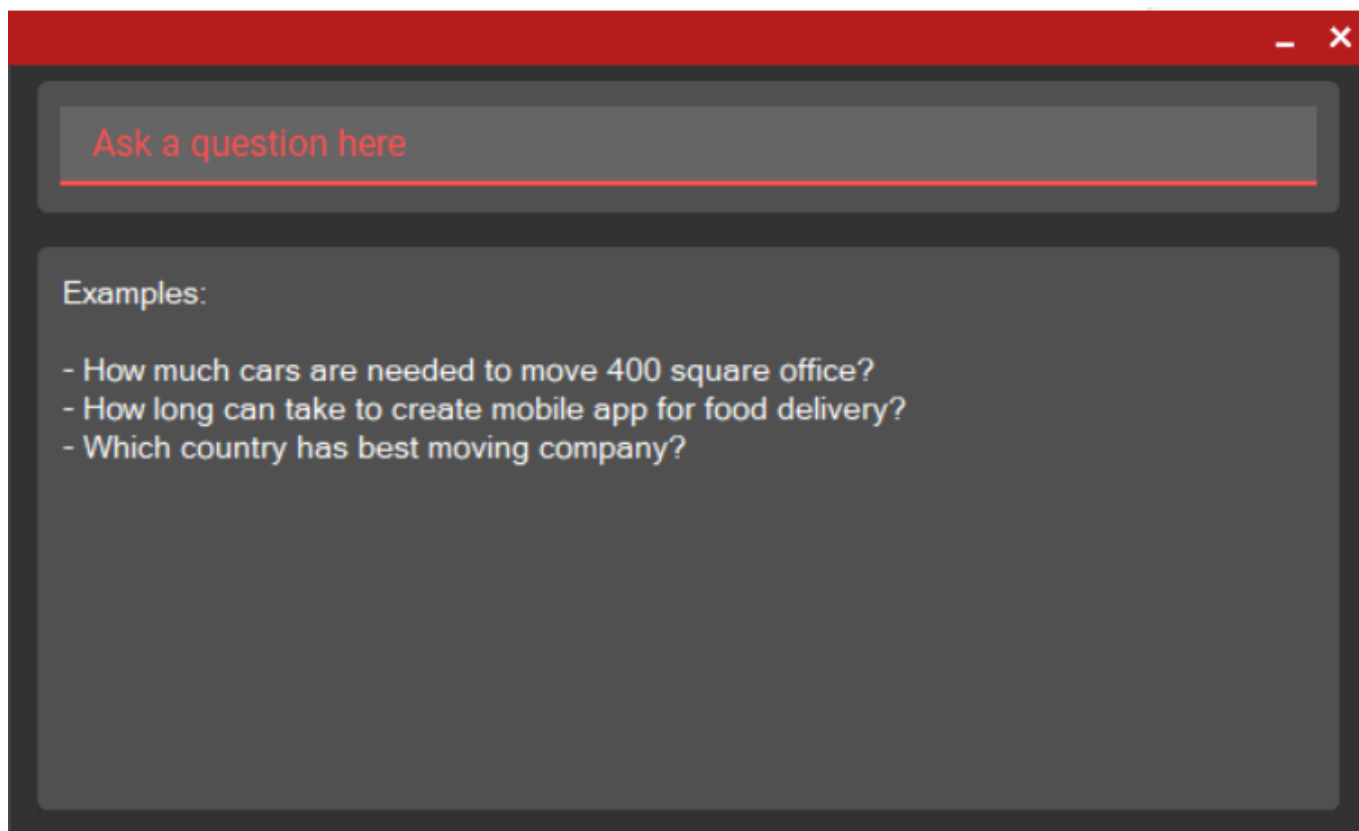


Рис. 4.1. Вигляд головної сторінки ПЗ.

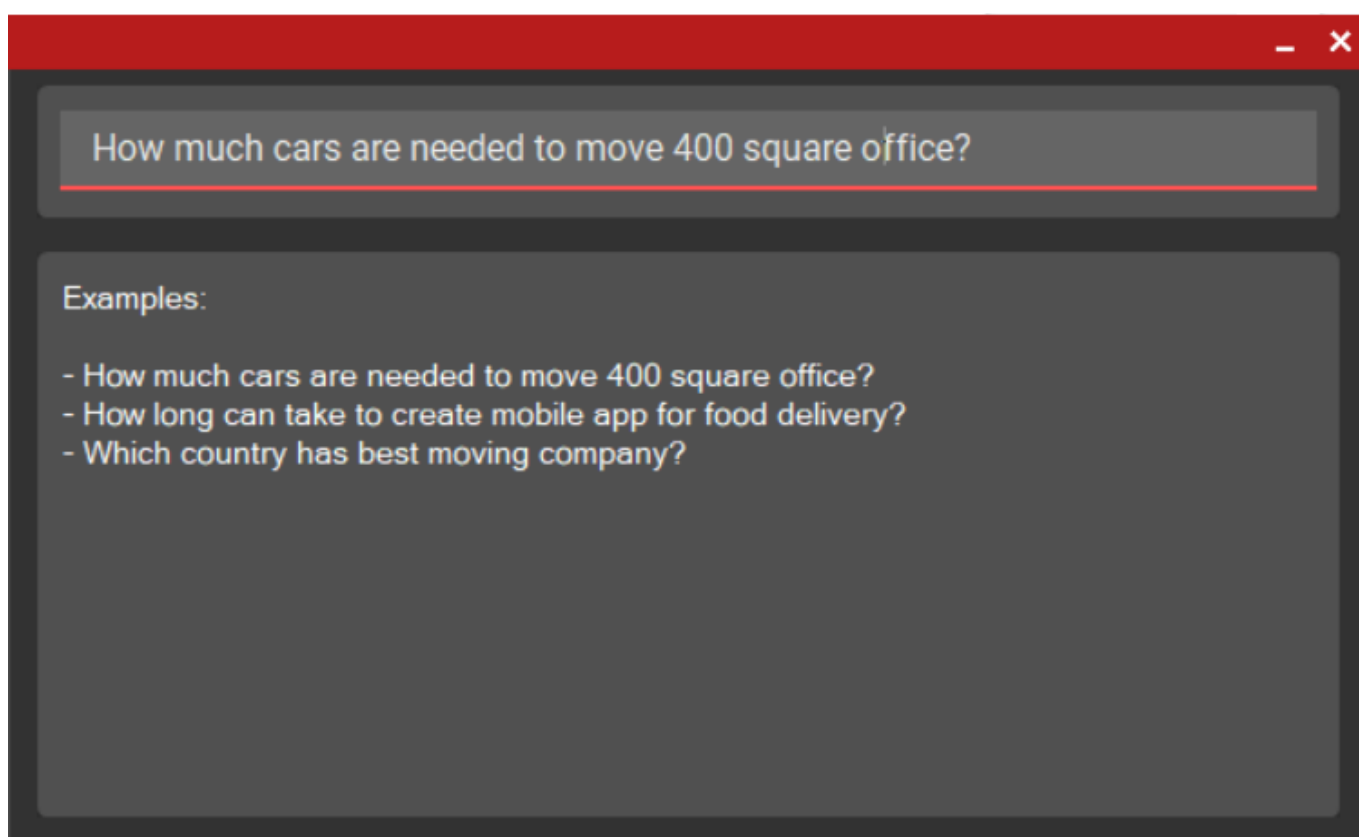


Рис. 4.2. Приклад використання ПЗ.

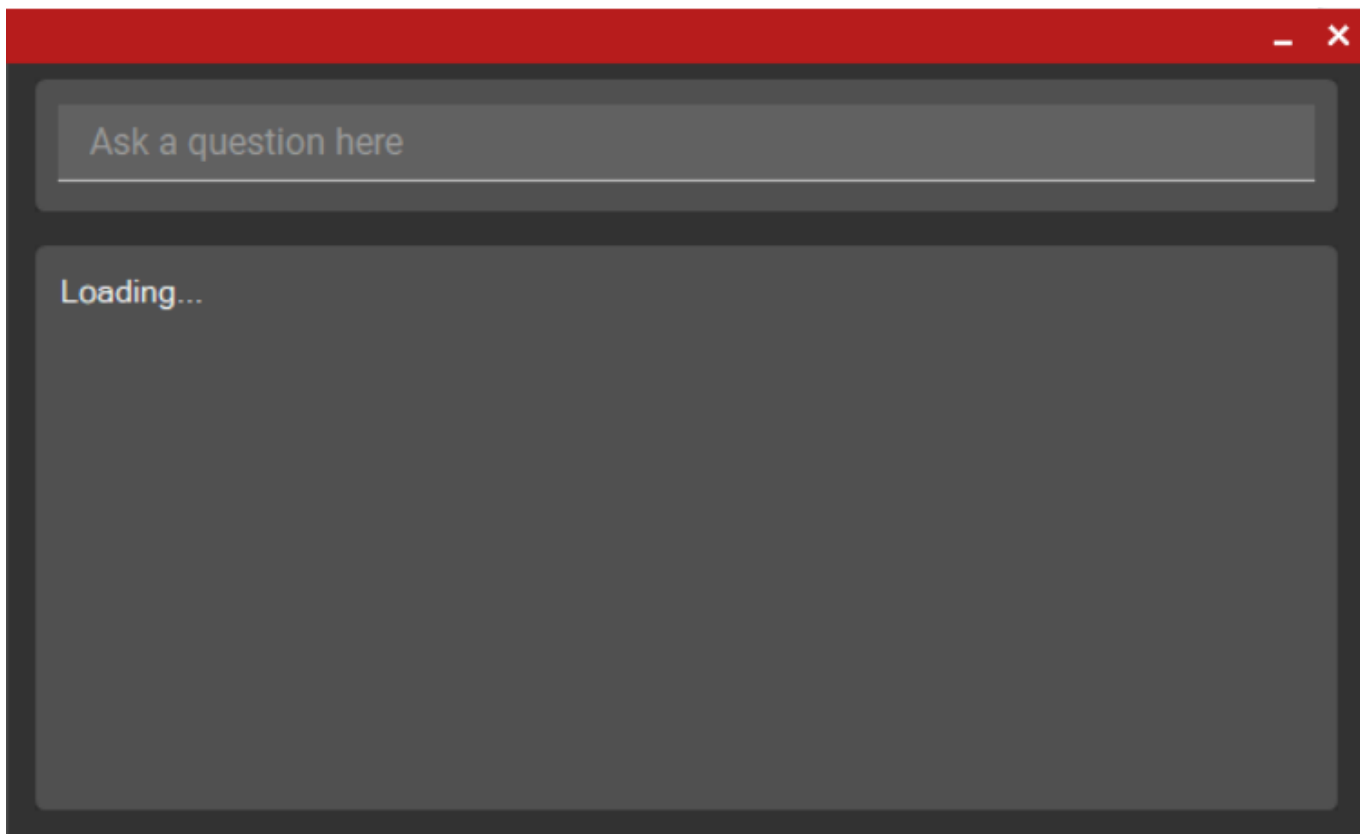


Рис. 4.3. Вигляд очікування на результати ПЗ.

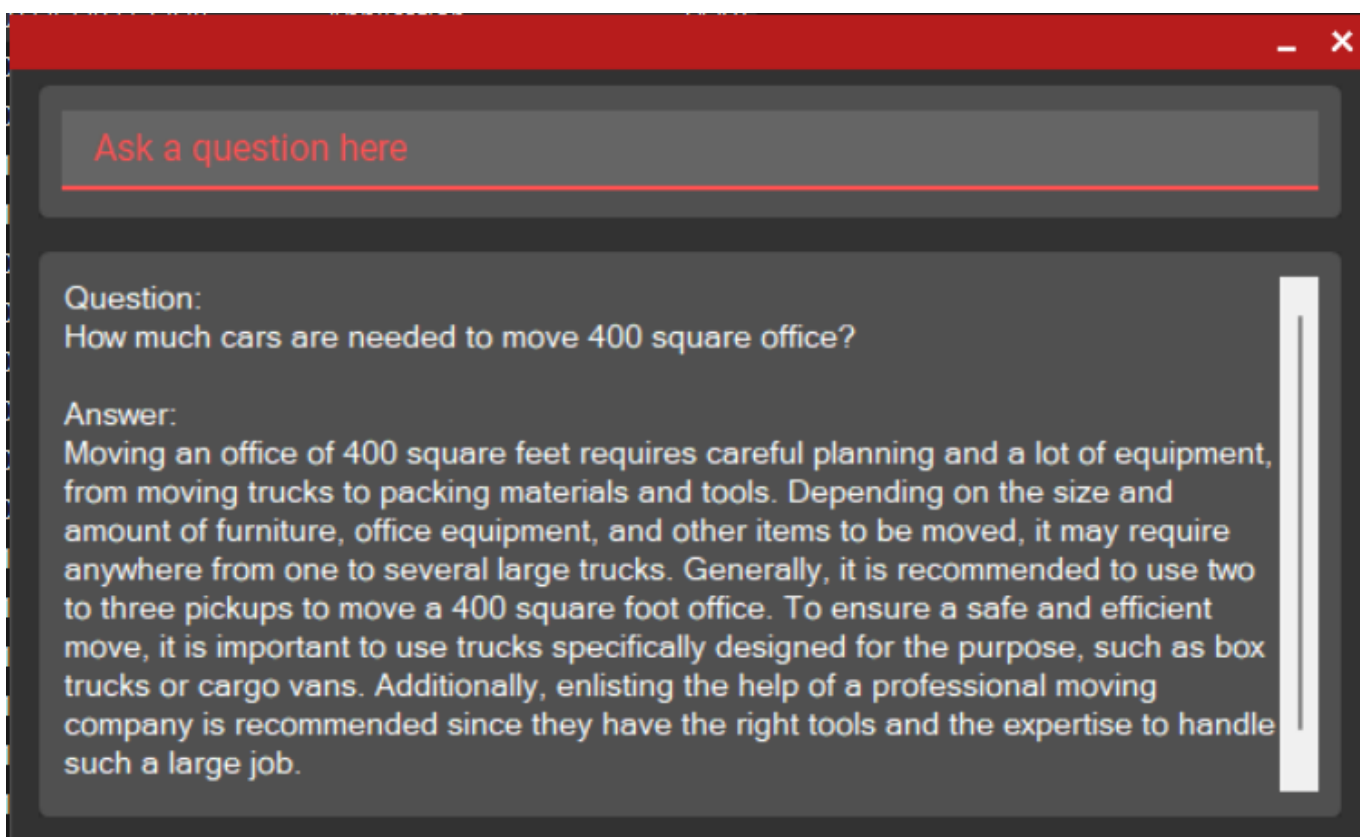


Рис. 4.4. Приклад роботи ПЗ № 1.

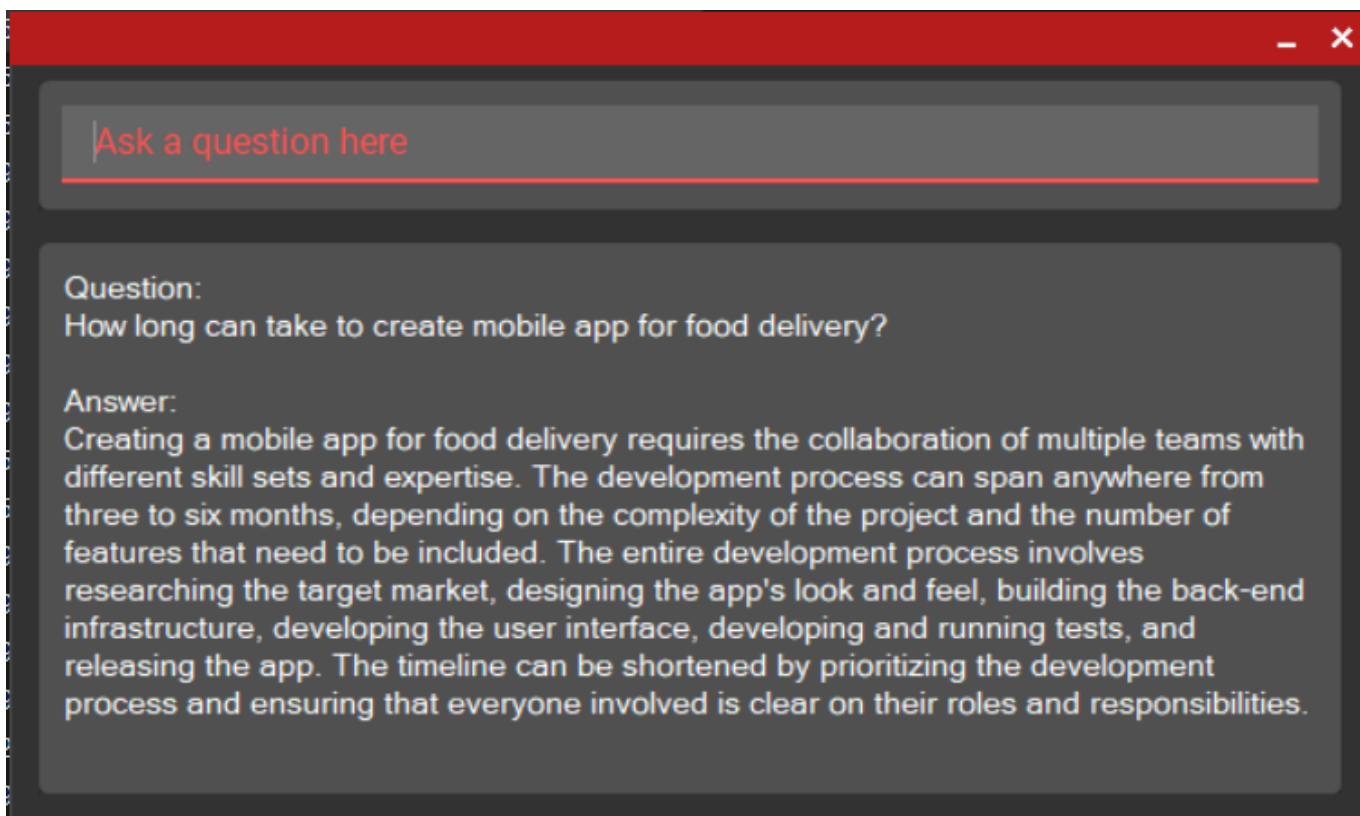


Рис. 4.5. Приклад роботи ПЗ № 2.

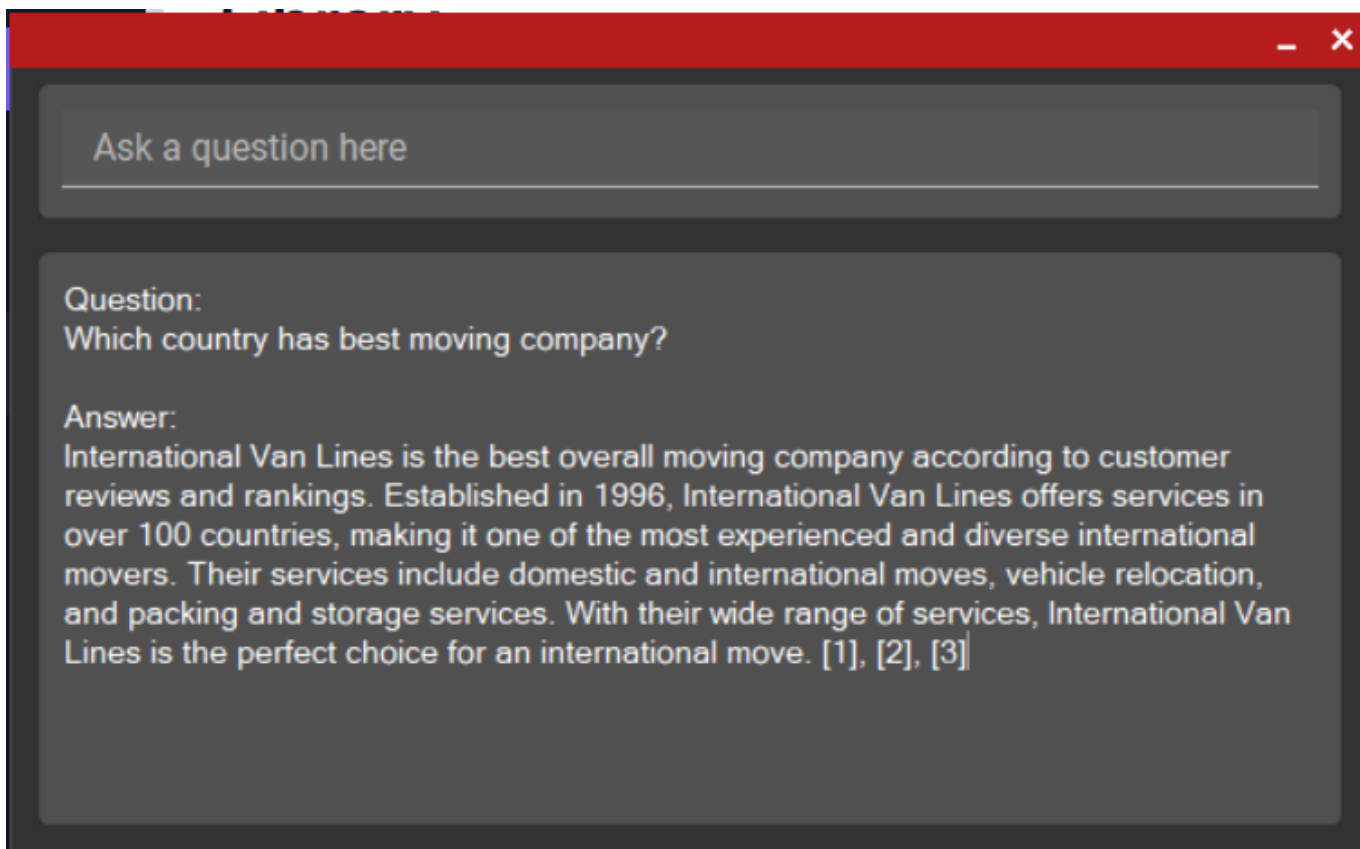


Рис. 4.6. Приклад роботи ПЗ № 3.

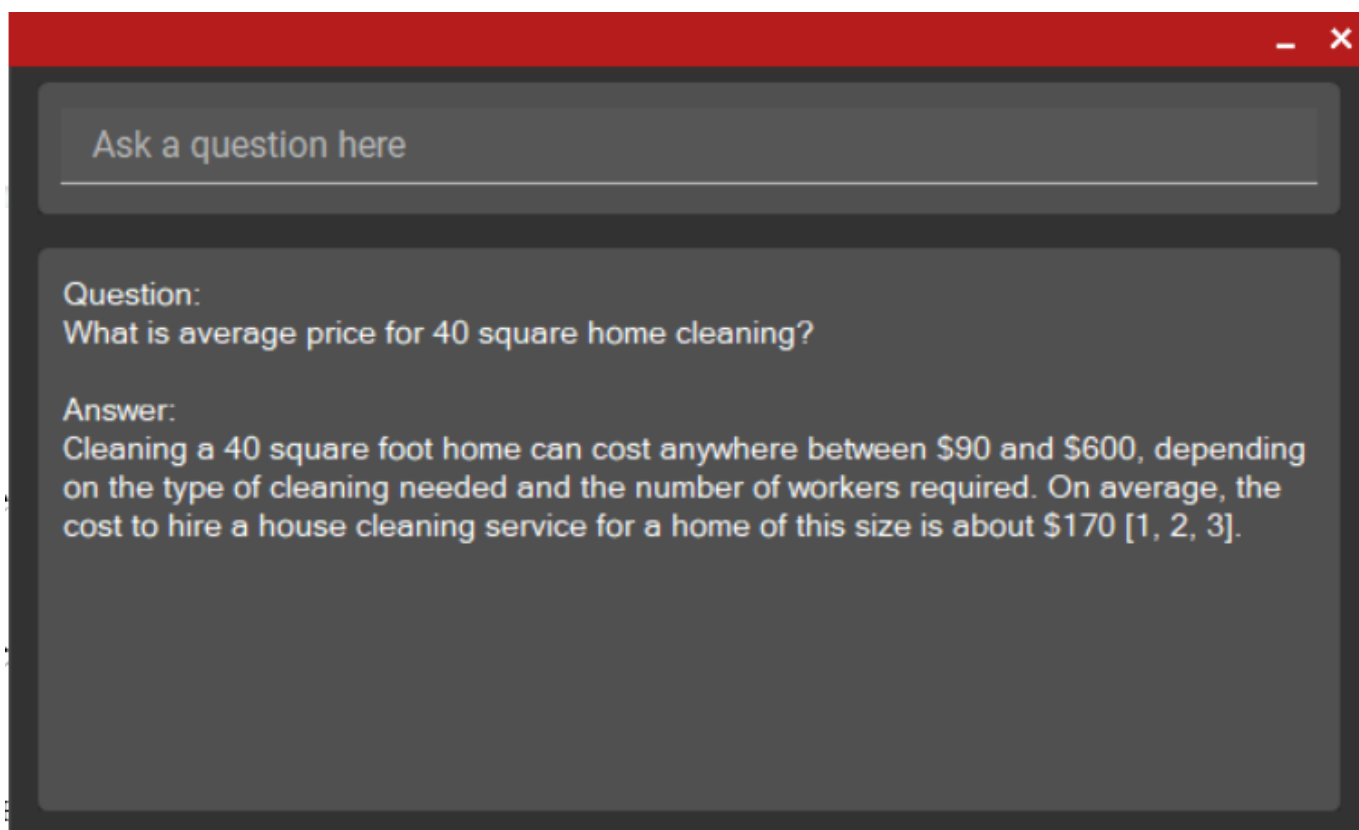


Рис. 4.7. Приклад роботи ПЗ № 4.

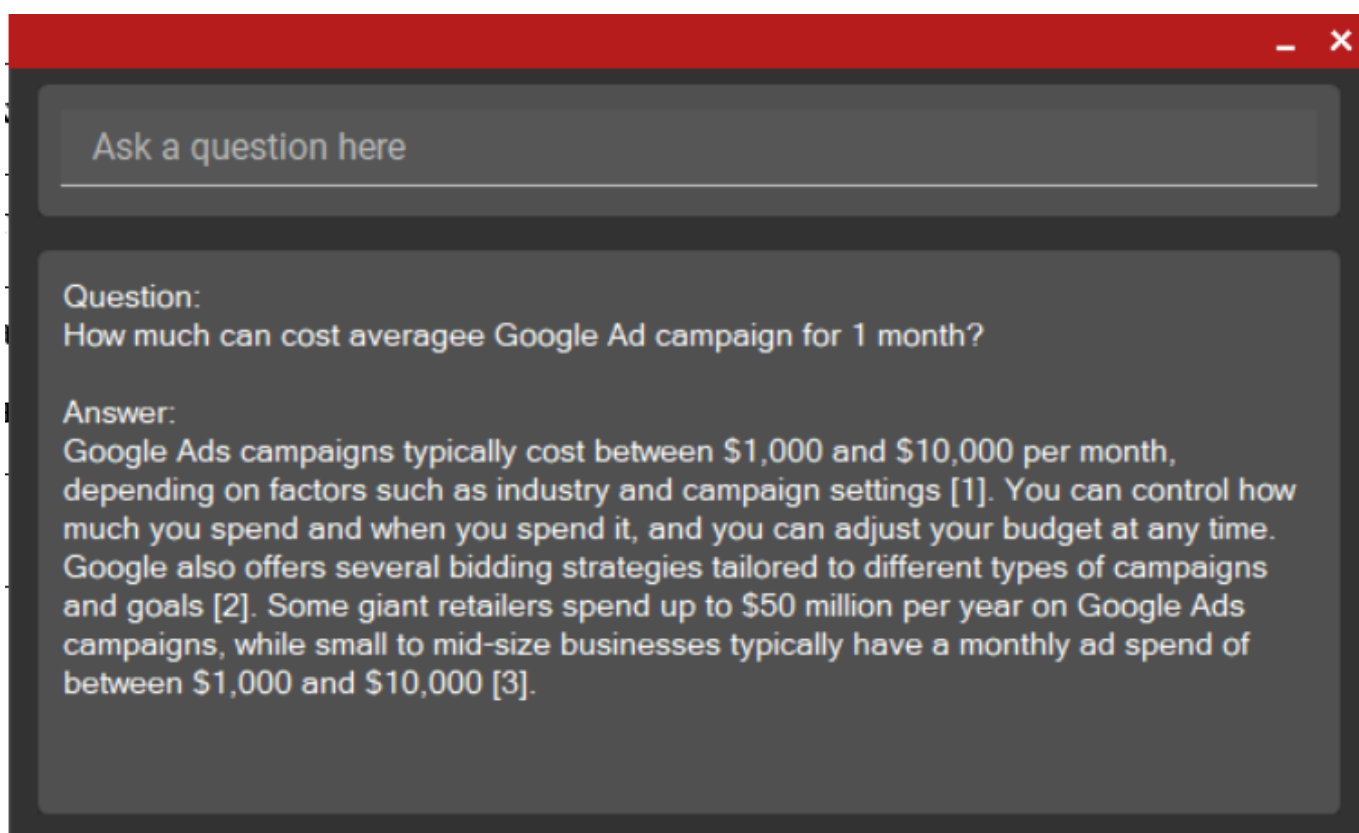


Рис. 4.8. Приклад роботи ПЗ № 5.

Для налаштування програми використовується текстовий файл, який може бути відкритий та відредагований будь-яким текстовим редактором. Загалом є два типи налаштувань (рис. 4.9):

Перший тип включає в себе основні налаштування:

- `conformity` - дозволяє задати процент відповідності. Можливі значення знаходяться в діапазоні 0-100.
- `check_google` - дозволяє вказати ПЗ чи включити в результат дані додатково знайдені через Google Search в даний момент. Можливі значення: `true/false`.

Другий тип включає в себе специфічні налаштування, з допомогою яких потрібно задати функції та параметри компанії.

- `business_name` - дозволяє задати ім'я компанії. В поточній версії ПЗ не несе ніякого впливу на результат.
- `business_function(1-n)` - дозволяє задати функції компанії, які в процесі будуть використані для порівняння.
- `business_parameter(1-n)` - дозволяє задати параметри компанії, які в процесі будуть використані для порівняння. Параметри задаються в форматі - `назва:значення`.

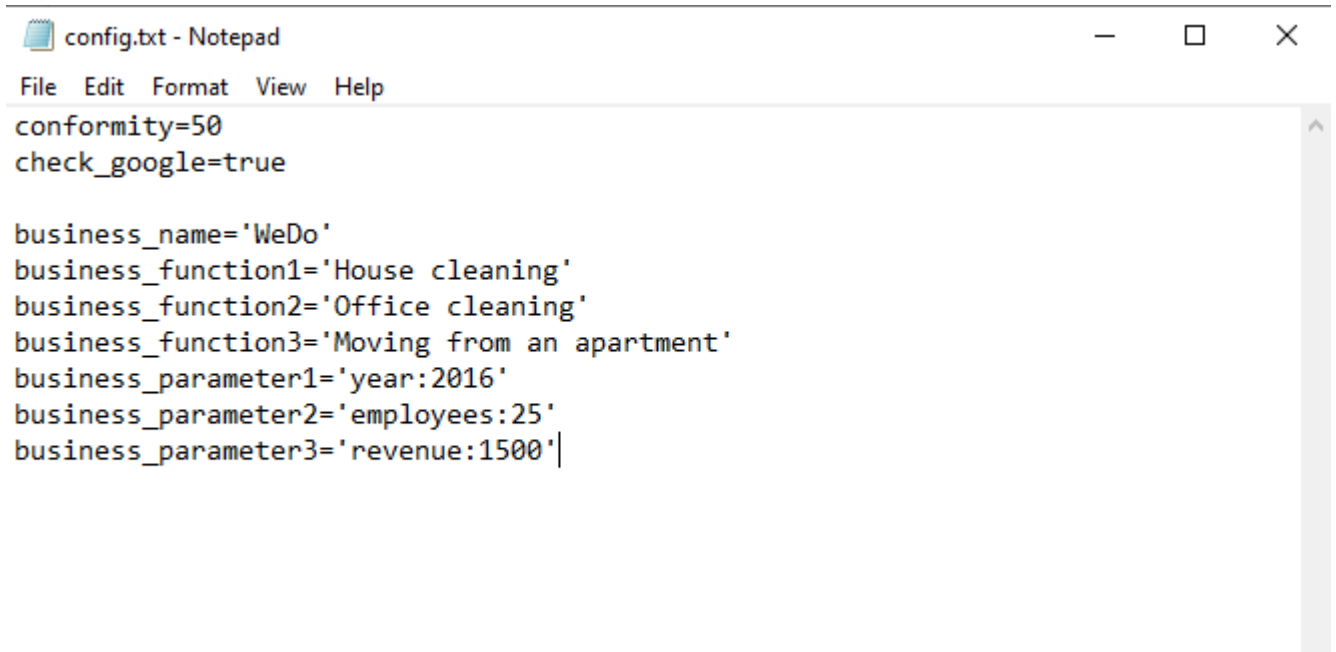


Рис. 4.9. Вигляд налаштувань ПЗ.

4.3. Аналіз результатів та застосовності розробленого програмного забезпечення

Розглянемо та проаналізуємо приклади наведені вище. Загалом програмному забезпеченню було задано п'ять питань і на всі з них був виданий результат.

Для порівняння результатів було використано - Perplexity AI. Це новий інструмент, який діє як пошукова система. Коли користувач вводить запитання, модель переглядає Інтернет, щоб дати відповідь. А також відображає джерело інформації, яку він надає.

Перше питання:

- How much cars are needed to move 400 square office? (*Скільки машин потрібно для переїзду офісу площею 400 квадратів?*)

Відповідь на перше питання:

- Moving an office of 400 square feet requires careful planning and a lot of equipment, from moving trucks to packing materials and tools. Depending on the

size and amount of furniture, office equipment, and other items to be moved, it may require anywhere from one to several large trucks. Generally, it is recommended to use two to three pickups to move a 400 square foot office. To ensure a safe and efficient move, it is important to use trucks specifically designed for the purpose, such as box trucks or cargo vans. Additionally, enlisting the help of a professional moving company is recommended since they have the right tools and the expertise to handle such a large job. *(Перенесення офісу площею 400 квадратних футів потребує ретельного планування та великої кількості обладнання, від транспортувальних вантажівок до пакувальних матеріалів та інструментів. Залежно від розміру та кількості меблів, офісного обладнання та інших предметів, які потрібно перемістити, може знадобитися від однієї до кількох великих вантажівок. Як правило, для переміщення офісу площею 400 квадратних футів рекомендується використовувати два-три пікапи. Щоб забезпечити безпечне та ефективне переміщення, важливо використовувати вантажівки, спеціально розроблені для цієї мети, наприклад, фургони або фургони. Крім того, рекомендується заручитися допомогою професійної транспортної компанії, оскільки вони мають необхідні інструменти та досвід для виконання такої великої роботи.)*

Відповідь Perplexity AI на перше питання:

- The question is unclear and cannot be answered directly with the given information. However, assuming that the question is asking how many cars are needed to move the contents of a 400 square foot office, the answer would depend on the amount and size of the items being moved, as well as the size of the cars being used for transportation. There are several factors that can affect the cost and logistics of moving an office, including the distance of the move, the amount and type of furniture and equipment being moved, and the number of employees involved in the move [1]. The cost of moving a 1,500 square foot office can range from \$750 to

\$4,500, while the cost of moving a 3,000 square foot office can range from \$1,500 to \$9,000 [2]. In terms of transportation, the amount of cars needed would depend on the size and capacity of the cars being used, as well as the amount and size of the items being moved. Some moving companies charge based on the amount of space needed in a trailer or moving truck, with larger offices requiring larger vehicles [3]. Overall, the number of cars needed to move a 400 square foot office cannot be determined without more specific information about the items being moved and the transportation being used. *(Питання незрозуміле, і на нього неможливо відповісти прямо з допомогою наданої інформації. Однак якщо припустити, що запитання полягає в тому, скільки машин потрібно для переміщення вмісту офісу площею 400 квадратних футів, відповідь залежатиме від кількості та розміру предметів, які переміщуються, а також від розміру автомобілів, які використовуються для транспортування. Є кілька факторів, які можуть вплинути на вартість і логістику переїзду офісу, включаючи відстань переїзду, кількість і тип меблів і обладнання, що переміщуються, а також кількість працівників, залучених до переїзду [1]. Вартість переїзду офісу площею 1500 квадратних футів може коливатися від 750 до 4500 доларів США, тоді як вартість переїзду офісу площею 3000 квадратних футів може коливатися від 1500 до 9000 доларів США [2]. З точки зору транспортування, кількість необхідних автомобілів залежатиме від розміру та місткості автомобілів, що використовуються, а також від кількості та розміру предметів, що переміщуються. Деякі транспортні компанії стягують плату залежно від кількості місця, необхідного в трейлері чи вантажівці, а для великих офісів потрібні більші транспортні засоби [3]. Загалом, кількість автомобілів, необхідних для переміщення офісу площею 400 квадратних футів, неможливо визначити без більш конкретної інформації про предмети, які переміщуються, і використовуваний транспорт.)*

How much cars are needed to move 400 square office?

PERPLEXITY [View Detailed](#)

The question is unclear and cannot be answered directly with the given information. However, assuming that the question is asking how many cars are needed to move the contents of a 400 square foot office, the answer would depend on the amount and size of the items being moved, as well as the size of the cars being used for transportation.

There are several factors that can affect the cost and logistics of moving an office, including the distance of the move, the amount and type of furniture and equipment being moved, and the number of employees involved in the move^[1]. The cost of moving a 1,500 square foot office can range from \$750 to \$4,500, while the cost of moving a 3,000 square foot office can range from \$1,500 to \$9,000^[2].

In terms of transportation, the amount of cars needed would depend on the size and capacity of the cars being used, as well as the amount and size of the items being moved. Some moving companies charge based on the amount of space needed in a trailer or moving truck, with larger offices requiring larger vehicles^[3].

Overall, the number of cars needed to move a 400 square foot office cannot be determined without more specific information about the items being moved and the transportation being used.

[] 3 SOURCES [View List](#)

1  nexusautotransport 2  profitableventure 3  F forbes

Рис. 4.10. Відповідь Perplexity AI на перше питання.

Оцінка відповіді на перше питання:

- Відповідь ПЗ є більш конкретною та практичною відповіддю на запитання, оскільки вона містить деякі загальні вказівки щодо кількості та типу транспортних засобів, необхідних для переміщення офісу певного розміру. Також рекомендується скористатися професійними послугами з переїзду, що є корисною порадою для всіх, хто планує переїзд. Однак у відповіді Perplexity

AI визнається, що питання є дещо незрозумілим, і надається додатковий контекст щодо факторів, які можуть вплинути на логістику та вартість переміщення офісу. Отже, обидві відповіді мають свої переваги, але відповідь ПЗ більш безпосередньо стосується поставленого питання.

Друге питання:

- How long can take to create mobile app for food delivery? *(Скільки часу може зайняти створення мобільного додатку для доставки їжі?)*

Відповідь на друге питання:

- Creating a mobile app for food delivery requires the collaboration of multiple teams with different skill sets and expertise. The development process can span anywhere from three to six months, depending on the complexity of the project and the number of features that need to be included. The entire development process involves researching the target market, designing the app's look and feel, building the back-end infrastructure, developing the user interface, developing and running tests, and releasing the app. The timeline can be shortened by prioritizing the development process and ensuring that everyone involved is clear on their roles and responsibilities. *(Створення мобільного додатку для доставки їжі вимагає співпраці кількох команд із різними навичками та досвідом. Процес розробки може тривати від трьох до шести місяців, залежно від складності проекту та кількості функцій, які необхідно включити. Весь процес розробки включає дослідження цільового ринку, розробку зовнішнього вигляду програми, побудову внутрішньої інфраструктури, розробку інтерфейсу користувача, розробку та виконання тестів і випуск програми. Час можна скоротити, визначивши пріоритети процесу розробки та переконавшись, що всі учасники чітко розуміють свої ролі та обов'язки.)*

Відповідь Perplexity AI на друге питання:

- The time it takes to create a mobile app for food delivery can vary depending on the complexity of the app. According to [1], back-end development can take 6-8 weeks, while [2] estimates that the development of an iPhone or Android app can take around \$25,000 – \$50,000. If the app requires backend development or complex functionality, that cost could go up significantly. [3] suggests that a food delivery app consists of four apps with various functions and different features, and it should include customer app, restaurant app, delivery app, and admin panel. [4] states that the average delivery app development time is 6.5 months. Therefore, the time it takes to create a mobile app for food delivery can range from a few weeks to several months, depending on the complexity of the app. *(Час, необхідний для створення мобільного додатку для доставки їжі, може відрізнятися залежно від складності додатка. За даними [1], бекенд-розробка може тривати 6-8 тижнів, тоді як [2] оцінює, що розробка програми для iPhone або Android може коштувати близько 25 000 – 50 000 доларів США. Якщо програмі потрібна розробка серверної частини або складна функціональність, вартість може значно зрости. [3] припускає, що додаток для доставки їжі складається з чотирьох додатків з різними функціями та особливостями, і він повинен включати додаток для клієнтів, додаток для ресторану, додаток для доставки та панель адміністратора. [4] стверджує, що середній час розробки програми для доставки становить 6,5 місяців. Таким чином, час, необхідний для створення мобільного додатку для доставки їжі, може становити від кількох тижнів до кількох місяців, залежно від складності додатка.)*

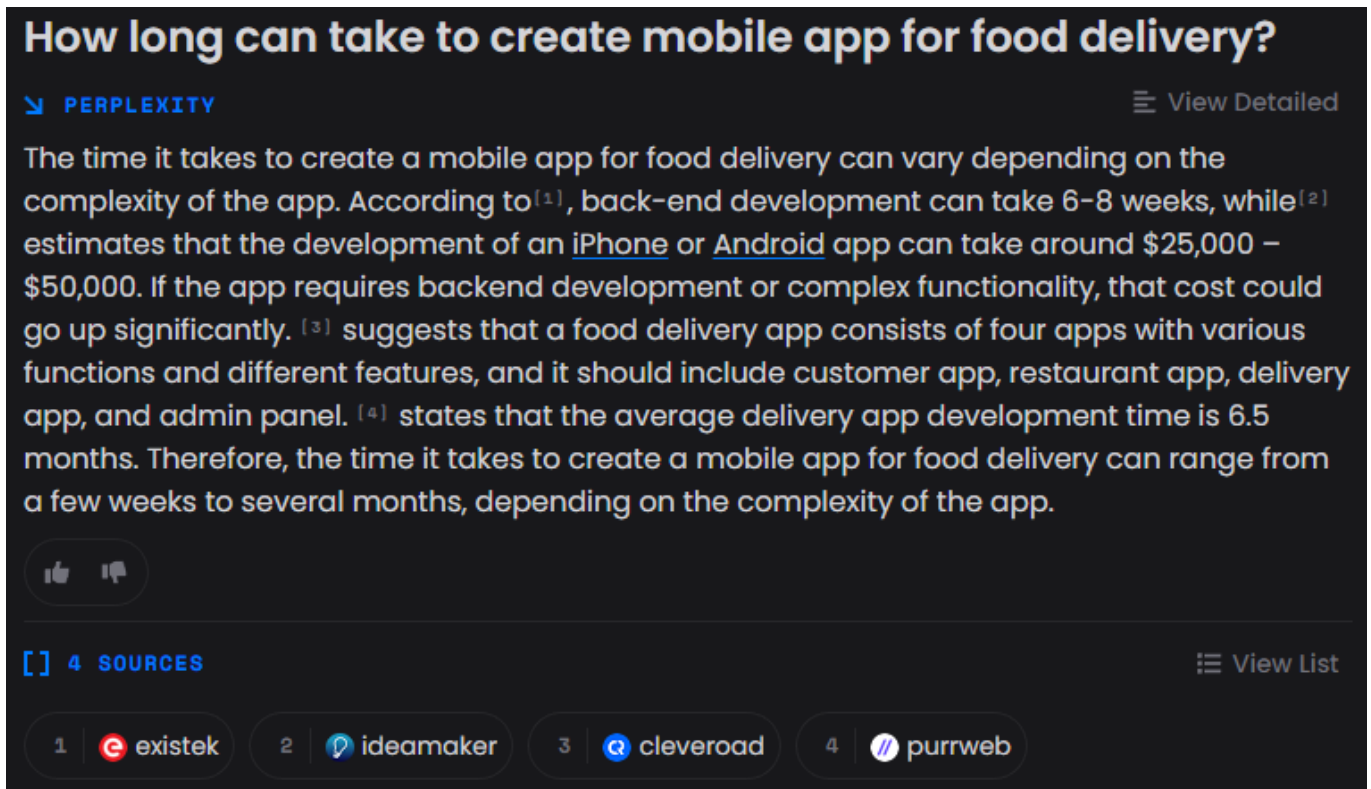


Рис. 4.11. Відповідь Perplexity AI на друге питання.

Оцінка відповіді на друге питання:

- Відповідь ПЗ і відповідь Perplexity AI надають відповідну інформацію про час, потрібний для створення мобільного додатку для доставки їжі. Відповідь ПЗ містить загальний графік процесу розробки від трьох до шести місяців із поясненням різних етапів. Відповідь Perplexity AI містить більш конкретні оцінки на основі різних джерел і висвітлює фактори, які можуть вплинути на часову шкалу, як-от складність програми та потреба у внутрішній розробці. Обидві відповіді є корисними, але відповідь Perplexity AI містить більш детальну інформацію та цитує конкретні джерела, що може зробити її більш корисною для тих, хто планує створити мобільний додаток. Тому відповідь Perplexity AI можна вважати кращою.

Третє питання:

- Which country has best moving company? (*У якій країні найкраща транспортна компанія?*)

Відповідь на третє питання:

- International Van Lines is the best overall moving company according to customer reviews and rankings. Established in 1996, International Van Lines offers services in over 100 countries, making it one of the most experienced and diverse international movers. Their services include domestic and international moves, vehicle relocation, and packing and storage services. With their wide range of services, International Van Lines is the perfect choice for an international move. (*International Van Lines є найкращою транспортною компанією за відгуками клієнтів і рейтингами. Заснована в 1996 році, International Van Lines пропонує послуги в понад 100 країнах, що робить її одним із найдосвідченіших і різноманітних міжнародних перевізників. Їхні послуги включають внутрішні та міжнародні переїзди, переміщення транспортних засобів, а також послуги пакування та зберігання. Завдяки широкому спектру послуг International Van Lines є ідеальним вибором для міжнародного переїзду.*)

Відповідь Perplexity AI на третє питання:

- There is no single country that has the best moving company. However, there are several international and long-distance moving companies that are highly rated by various sources. These include North American Van Lines, United Van Lines, International Van Lines, Allied Van Lines, and Bellhop [1][2][3][4][5]. It is recommended to research and compare different moving companies to find the one that best fits your needs and budget. (*Немає жодної країни з найкращою транспортною компанією. Однак існує кілька компаній, які займаються міжнародними переїздами та міжміськими перевезеннями, які мають високі оцінки з різних джерел. До них належать North American Van Lines, United Van Lines, International Van Lines, Allied Van Lines і Bellhop [1][2][3][4][5].*)

Рекомендується дослідити та порівняти різні транспортні компанії, щоб знайти ту, яка найкраще відповідає вашим потребам і бюджету.)

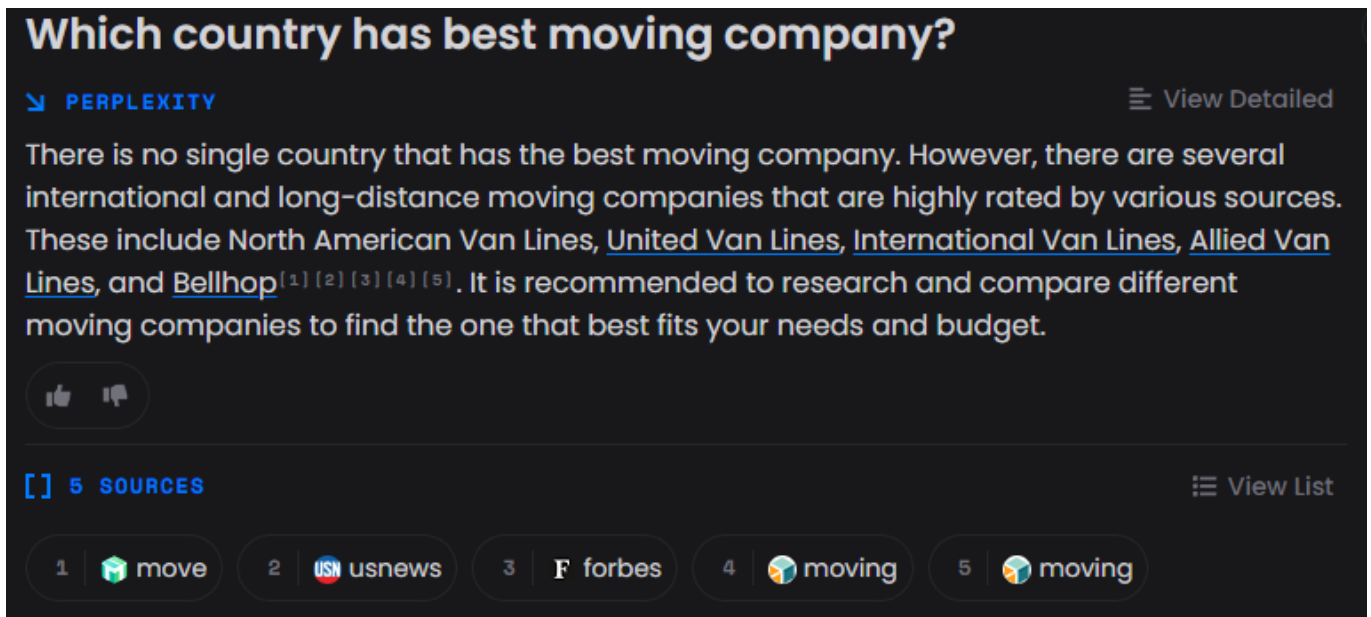


Рис. 4.12. Відповідь Perplexity AI на третє питання.

Оцінка відповіді на третє питання:

- Відповідь Perplexity AI є кращою відповіддю, оскільки вона визнає, що немає жодної країни з найкращою транспортною компанією, і надає список високо оцінених міжнародних та міжміських транспортних компаній, які можна розглянути для переїзду. Він також рекомендує дослідити та порівняти різні транспортні компанії, щоб знайти ту, яка найкраще відповідає потребам і бюджету. Ця інформація більш корисна та точна порівняно з відповіддю ПЗ, у якій рекомендовано лише одну компанію без надання будь-якого контексту чи порівняння з іншими транспортними компаніями.

Четверте питання:

- What is average price for 40 square home cleaning? (Яка середня ціна за прибирання будинку 40 квадратів?)

Відповідь на четверте питання:

- Cleaning a 40 square foot home can cost anywhere between \$90 and \$600, depending on the type of cleaning needed and the number of workers required. On average, the cost to hire a house cleaning service for a home of this size is about \$170. *(Прибирання будинку площею 40 квадратних футів може коштувати від 90 до 600 доларів, залежно від типу необхідного прибирання та кількості необхідних працівників. У середньому вартість послуг з прибирання будинку такого розміру становить близько 170 доларів.)*

Відповідь Perplexity AI на четверте питання:

- The average price for cleaning a 40 square home varies depending on the level of cleaning required and the cleaner. According to the sources, the average cost of house cleaning services is \$40-\$65 per hour per cleaner [1]. The cost per square foot ranges from \$0.08 to \$0.30, depending on the level of cleaning required [2]. The cost calculator on HomeGuide.com estimates that cleaning a 2,000-square-foot home with 3 bedrooms and 2 bathrooms costs \$150 to \$250 to clean or about \$0.07 to \$0.12 per square foot [1]. It is recommended to get quotes from multiple cleaners to compare prices and services. *(Середня ціна за прибирання будинку площею 40 квадратів залежить від необхідного рівня прибирання та прибиральника. Згідно з джерелами, середня вартість послуг з прибирання житла становить 40-65 доларів США за годину одного прибиральника [1]. Вартість квадратного фута коливається від \$0,08 до \$0,30 залежно від необхідного рівня очищення [2]. Калькулятор витрат на HomeGuide.com підрахував, що прибирання будинку площею 2000 квадратних футів з 3 спальнями та 2 ванними кімнатами коштує від 150 до 250 доларів США на прибирання або приблизно від 0,07 до 0,12 доларів США за квадратний фут [1]. Рекомендується отримати пропозиції від кількох прибиральників, щоб порівняти ціни та послуги.)*

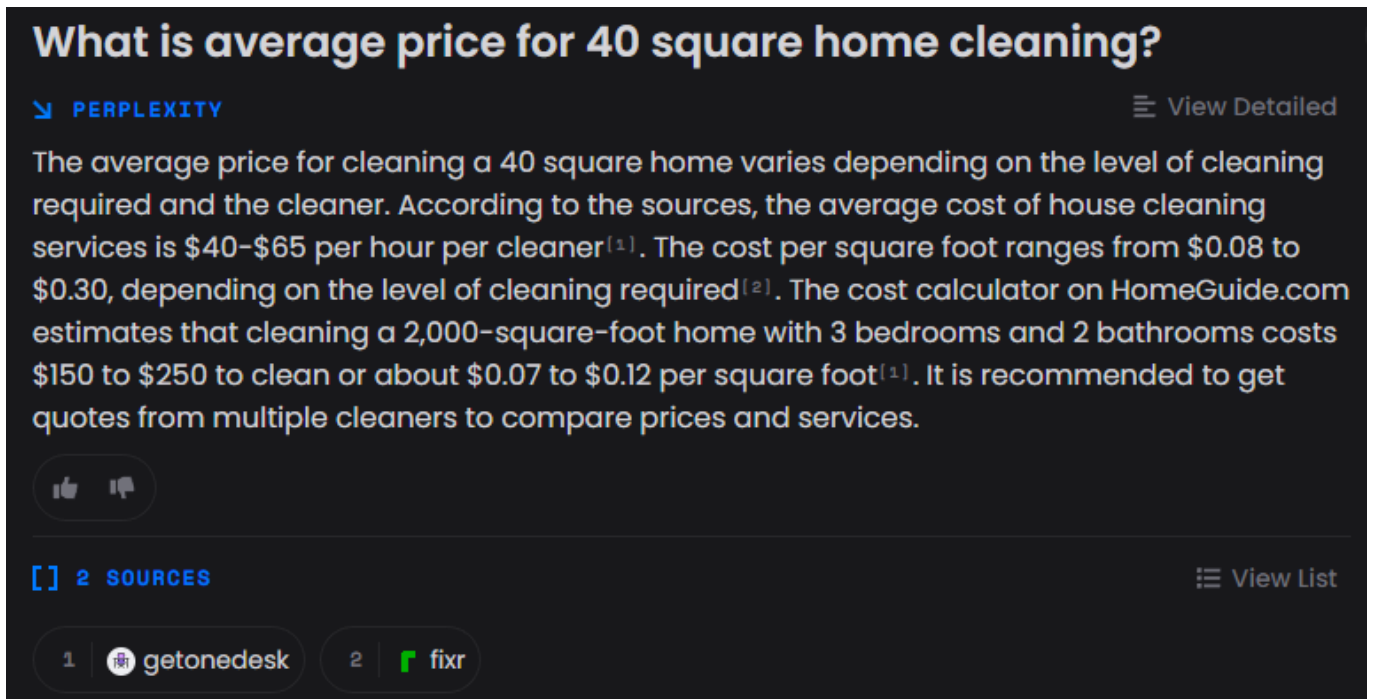


Рис. 4.13. Відповідь Perplexity AI на четверте питання.

Оцінка відповіді на четверте питання:

- Відповідь ПЗ надає подібну оцінку вартості прибирання будинку площею 40 квадратних футів, але надає ширший діапазон можливих витрат (від 90 до 600 доларів США) порівняно з відповіддю Perplexity AI, яка надає більш точну оцінку на основі вартості за годину та вартості за квадратний фут. Обидві відповіді підкреслюють, що вартість прибирання будинку такого розміру може змінюватися залежно від таких факторів, як необхідний рівень прибирання та кількість необхідних прибиральників.

П'яте питання:

- How much can cost average Google Ad campaign for 1 month? (*Скільки може коштувати в середньому рекламна кампанія Google Ad за 1 місяць?*)

Відповідь на п'яте питання:

- Google Ads campaigns typically cost between \$1,000 and \$10,000 per month, depending on factors such as industry and campaign settings. You can control how much you spend and when you spend it, and you can adjust your budget at any time. Google also offers several bidding strategies tailored to different types of campaigns and goals. Some giant retailers spend up to \$50 million per year on Google Ads campaigns, while small to mid-size businesses typically have a monthly ad spend of between \$1,000 and \$10,000. *(Кампанії Google Ads зазвичай коштують від 1000 до 10 000 доларів США на місяць залежно від таких факторів, як галузь і налаштування кампанії. Ви можете контролювати, скільки ви витрачаєте та коли ви витрачаєте, і ви можете змінити свій бюджет у будь-який час. Google також пропонує кілька стратегій призначення ставок, адаптованих до різних типів кампаній і цілей. Деякі гігантські роздрібні торговці витрачають до 50 мільйонів доларів на рік в кампанії Google Ads, тоді як малі та середні підприємства зазвичай витрачають на рекламу від 1000 до 10 000 доларів на місяць.)*

Відповідь Perplexity AI на п'яте питання:

- The cost of an average Google Ads campaign for one month can vary widely depending on factors such as the industry, competition, and target audience. In 2023, companies can expect to spend anywhere from \$9,000 to over \$30,000 per month on Google Ads [1]. To calculate a monthly budget, you can multiply your average daily budget by 30.4 [2]. The average cost per click on the Search Network is between \$1 and \$2, while the average CPC on the Google Display Network is under \$1 [3]. The most expensive keywords can cost \$50 or more per click [3]. However, a recent study by Google showed that the average ROI on Google Ads is 800%, or \$8 for every \$1 spent [4]. *(Вартість середньої кампанії Google Ads за один місяць може значно відрізнятися залежно від таких факторів, як галузь, конкуренція та цільова аудиторія. У 2023 році компанії можуть розраховувати*

витрачати на Google Ads від 9000 до понад 30 000 доларів США на місяць [1]. Щоб розрахувати місячний бюджет, ви можете помножити свій середній щоденний бюджет на 30,4 [2]. Середня ціна за клік у пошуковій мережі становить від 1 до 2 доларів, тоді як середня CPC в Медійній мережі Google становить менше 1 долара [3]. Найдорожчі ключові слова можуть коштувати 50 доларів або більше за клік [3]. Однак нещодавнє дослідження Google показало, що середня рентабельність інвестицій у Google Ads становить 800%, або 8 доларів США за кожен витрачений 1 долар [4].)

The screenshot shows a dark-themed interface for Perplexity AI. At the top, the question "How much can cost average Google Ad campaign for 1 month?" is displayed in a large, bold, white font. Below the question, the Perplexity logo is visible on the left, and a "View Detailed" link is on the right. The main body of the response is a paragraph of white text providing information about Google Ads costs, including a range from \$9,000 to over \$30,000 per month, and a calculation method involving a 30.4 multiplier. It also mentions average CPC for Search and Display Networks, and a recent study by Google showing an 800% ROI. Below the text, there are thumbs up and down icons. At the bottom, a section titled "4 SOURCES" lists four sources: 1. webfx, 2. google, 3. scorpion, and 4. wordstream, each with its respective logo.

Рис. 4.14. Відповідь Perplexity AI на п'яте питання.

Оцінка відповіді на п'яте питання:

- Обидві відповіді надають достовірну інформацію про вартість кампанії Google Ads, але відрізняються конкретними цифрами. Відповідь ПЗ передбачає загальний діапазон від 1000 до 10 000 доларів США на місяць, зазначаючи, що

деякі гігантські роздрібні торговці можуть витратити до 50 мільйонів доларів на рік. Відповідь Perplexity AI містить більш конкретні оцінки, згідно з якими компанії можуть розраховувати витратити від 9000 до понад 30 000 доларів США на Google Ads у 2023 році, а середня ціна за клік у пошуковій мережі становить від 1 до 2 доларів США. Обидві відповіді підкреслюють, що вартість кампанії Google Ads може змінюватися залежно від різних факторів і що компанії можуть контролювати свої витрати та коригувати свій бюджет за потреби.

І відповідь ПЗ, і відповідь Perplexity AI надають корисну інформацію та розуміння поставлених запитань. Відповідь ПЗ, як правило, надає більш конкретну відповідь, часто з вузьким діапазоном значень або конкретною оцінкою, тоді як відповідь Perplexity AI, як правило, надає ширший спектр інформації, включаючи середні витрати або діапазон оцінок. Відповідь ПЗ часто надає більше деталей про фактори, які можуть впливати на витрати чи результати, тоді як відповідь Perplexity AI зазвичай містить більше джерел і статистичних даних. Загалом обидва типи відповідей можуть бути корисними для надання вичерпної відповіді на запитання залежно від необхідного рівня деталізації та конкретності.

Проаналізувавши отриману програму, можна зробити висновки, що одержаний основний функціонал необхідний для системи підтримки прийняття рішень, який дозволяє задавати питання та отримувати на них відповіді. Відповіді зачасту включають в себе опис та рекомендації. Дану програму також варто було б розширити таким чином, щоб компанія мала б більш розширені налаштування, наприклад можливість додатково окремо надавати ключові слова та формувати тон відповіді. Додати відображення джерела інформації. Додати можливість збереження запитів користувачів. А також вбудувати в ПЗ перекладач, який б надав можливість користуватись програмою без знання англійської мови.

Висновки до розділу 4

На основі поставлених завдань та засобів вирішення із третього розділу було реалізовано прототип програмного забезпечення. Наведено опис, приклади та демонстрацію програмного забезпечення. Описано варіанти для майбутнього розширення.

ВИСНОВКИ

У дисертаційній роботі розв'язано актуальне науково-прикладне завдання, яке полягає у розробленні методів та засобів підтримки прийняття рішення в умовах багатозначності опису предметної області. Дослідження, проведені в рамках даної дисертаційної роботи, мали на меті дослідити та вирішити низку проблем, пов'язаних з прийняттям рішень в умовах багатозначності. Розробленню нових та покращенню наявних методів та засобів підтримки прийняття рішень, що могли б допомогти фахівцям у різних галузях зробити більш обґрунтовані та ефективні рішення.

Під час наукового дослідження вирішено такі завдання та отримано такі результати:

1. Аналіз існуючих підходів до побудови систем підтримки прийняття рішень показав, що необхідно удосконалювати процеси концептуалізації проблемних областей з огляду на контекстну залежність та середовище функціонування інтелектуального агента - децидента.

Це дозволить забезпечити ефективне прийняття рішень у процесі роботи інтелектуального агента. Наприклад, врахування контекстуальних факторів може допомогти уникнути помилкових рішень, а адаптація до середовища дозволить інтелектуальному агенту ефективніше взаємодіяти зі своїм оточенням. У цілому, аналіз підходів до побудови систем підтримки прийняття рішень демонструє, що вдосконалення процесів концептуалізації проблемних областей є ключовим етапом для забезпечення ефективної роботи інтелектуального агента - децидента у процесі прийняття рішення.

2. Аналіз методів застосування онтологічного моделювання для концептуалізації проблемних ситуацій показав, що однією з головних переваг онтологічного моделювання є можливість точного та контекстно-орієнтованого подання моделей предметних областей у процесі прийняття рішення.

Однак, при використанні онтологічного моделювання для концептуалізації проблемних ситуацій необхідно мати на увазі, що створення онтології є складним процесом, який вимагає багато ресурсів та часу. Крім того, у разі якщо ми маємо справу з складною предметною областю, можуть виникнути проблеми з узгодженням онтологій, що може призвести до неточностей у визначенні понять та зв'язків між ними.

Тому, для досягнення точності та контекстно-орієнтованості у використанні онтологічного моделювання для концептуалізації проблемних ситуацій, необхідно проводити багатократне та багаторівневе узгодження онтологій. Це допоможе забезпечити точне та повне подання моделей предметних областей, що, у свою чергу, дозволить прийняти більш обґрунтовані та ефективні рішення в проблемних ситуаціях.

3. Розроблено метод багаторівневого узгодження структур на основі онтологічних мереж, що дозволяє зменшити кількість помилок концептуалізації. Цей метод включає узгодження предметних областей, узгодження онтологій, узгодження змісту концептів і визначення стандартів для спільного розуміння і використання цих концептів.

Такий підхід до узгодження знань дозволяє забезпечити взаємодію між різними системами та організаціями, які працюють зі знаннями в певній предметній області. Багаторівневе узгодження структур забезпечує ефективну і точну передачу знань та підвищує якість прийняття рішень в даній області.

4. Розроблено метод вибору провайдера знань на основі модифікованого методу TOPSIS, який забезпечує максимальну відповідність між вимогами користувача та профілем діяльності провайдера знань. Для досягнення цього метод використовує модифікований метод TOPSIS, який дозволяє зіставляти провайдерів знань за допомогою матриці прийняття рішень.

Основна ідея методу полягає в тому, щоб враховувати вимоги користувача,

враховуючи профіль діяльності провайдера знань, який найбільш повно відповідає цим вимогам. Це досягається шляхом представлення вимог як ідеального варіанту вибору у просторі критеріїв та порівнянні кожного провайдера знань з ідеальним варіантом.

Онтологія використовується для явного відображення вимог до предметної області провайдера знань. Це дозволяє враховувати важливі параметри, які відображаються в онтології, що забезпечує більш точне і повне порівняння провайдерів знань.

5. Запропоновано модель подання контекстно залежного змісту концептів онтології на основі теорії прототипів, яка полягає у використанні гнучкого підходу до визначення концепту, де кожен концепт має свій прототип - типовий представник, який може мати різні характеристики залежно від контексту його використання.

Це дозволяє агентові обирати чітку інтерпретацію концепту залежно від поточного контексту та завдання, яке він вирішує. Крім того, в процесі діяльності агент може будувати нові інтерпретації концепту, що дозволяє розширювати його знання та адаптуватися до нових ситуацій. Такий підхід до подання концептів є особливо корисним для систем штучного інтелекту, які повинні вирішувати завдання у складних та змінних середовищах.

6. Реалізовано прототип програмного забезпечення на основі визначених завдань та відповідних рішень. Розроблене програмне забезпечення може використовуватися для аналізу підприємства, що дозволяє отримувати інформацію про нові тенденції на ринку та відповідно налаштовувати свої бізнес-стратегії. Використання відповідного програмного забезпечення може значно покращити ефективність та точність роботи підприємства, сприяти підвищенню задоволеності клієнтів та розвитку конкурентоспроможності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Карпов І., Буров Є. *Використання онтологічних мереж у системах підтримки прийняття рішень в умовах неоднозначності*. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2020. Вип. 7. С. 8–15.
2. Burov Y., Mykich K., Karpov I. *Building a versatile knowledge-based system based on reasoning services and ontology representation transformations*. International Scientific and Technical Conference "Computer Sciences and Information Technologies". 2020. 15th IEEE Intern. sci. and techn. conf. on computer sciences and information technologies CSIT 2020, Lviv-Zbarazh, Ukraine, 23-26 Sept. 2020. Vol. 2. P. 255–260.
3. Карпов І., Буров Є. *Онтології у процесі прийняття рішень*. Вісник Хмельницького національного університету. Серія: «Технічні науки». 2023. № 2, т. 1. С. 149–153.
4. Burov Y., Karpov I. *Contextual concept meaning alignment based on prototype theory*. CEUR Workshop Proceedings. 2023. Vol. 3403 : Computational linguistics and intelligent systems 2023 : proc. of the 7th Intern. conf. on computational linguistics and intelligent systems. Vol. III: Intelligent systems workshop. Kharkiv, Ukraine, Apr. 20-21, 2023. P. 137–146.
5. Буров Є., Карпов І. *Вибір концептуалізації проблемної ситуації інтелектуальним агентом в задачах прийняття рішення*. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2023. Вип. 13.
6. Карпов І., Буров Є. *Підхід до побудови системи підтримки прийняття рішень на основі онтологій*. Scientific Collection «InterConf». 2022. № 99 : Proc. of the 4th Intern. sci. and pract. conf. "International scientific discussion: problems, tasks and prospects", Brighton, Great Britain, Febr. 19-20, 2022. С. 849–855.

7. Карпов І., Буров Є. *Особливості застосування онтологій в системах підтримки прийняття рішень: теоретичний та практичний аспект*. Грааль науки. 2022. № 12-13. Proc. of the III Correspondence intern. sci. and pract. conf. «An integrated approach to science modernization: methods, models and multidisciplinary», Apr. 29th, 2022. С. 307–310.
8. Карпов І., Буров Є. *Застосування онтологій для прикладних задач*. III CISP Conference «Globalization of scientific knowledge: international cooperation and integration of sciences» III Міжнародна науково-практична конференція. 2021. Вінниця - Відень. С. 344–345.
9. Карпов І., Буров Є. *Онтологічне моделювання: цілі та засоби*. Scientific Collection «InterConf», (107): with the Proceedings of the 12th International Scientific and Practical Conference «Science and Practice: Implementation to Modern Society». 2022. Manchester, Great Britain. С. 517–519.
10. Карпов І., Буров Є. *Особливості використання онтологій у системах підтримки прийняття рішень: закономірності та особливості*. I International Scientific and Theoretical Conference «Technologies and strategies for the implementation of scientific achievements» I Міжнародна науково-теоретична конференція. European Scientific Platform. 2022. Стокгольм, SWE. С. 76–77.
11. Карпов І., Буров Є. *Проблеми онтологічного підходу у віртуальному навчальному середовищі*. III International Scientific and Practical Conference «Education and science of today: intersectoral issues and development of sciences». III Міжнародна науково-практична конференція. 2022. Кембридж, GBR. С. 187–188.
12. Karpov I., Burov Y. *Use of ontological networks in decision support systems under ambiguity*. Debats scientifiques et orientations prospectives du developpement scientifique : materiales de la conferencia internacional científica y práctica - "ΛΟΓΟΣ". Paris. Vol. 3. 2021. С. 97–103.

- 13.Іванов, О. *Системи підтримки прийняття рішень в управлінні*. Київ: Видавництво України. 2018. С. 45-60.
- 14.Петров, В. *Аналіз методів підтримки прийняття рішень*. Журнал Управління, 12(3). 2017. С. 112-125.
- 15.Сидоренко, Л. *Інформаційні технології в системах підтримки прийняття рішень*. Комп'ютерні науки, 8(2). 2019. С. 78-89.
- 16.Мельник, А. *Роль інтелектуальних систем в процесі прийняття рішень*. Інновації в управлінні, 5(1). 2020. С. 33-45.
- 17.Inmon W. H. *Building the Data Warehouse* / W. H. Inmon. - Willey. - 2005.
- 18.Inmon W. H. *Using the Data Warehouse* / W. H. Inmon, R. Hackthorn. Willey. - 1994.
- 19.Hackathorn R. *Reinventing Enterprise Systems Via Data Warehousing* / R. Hackthorn. - Washington, DC: The Data Warehousing Institute Annual Conference. - 1995.
- 20.Кравчук, І. *Моделювання систем підтримки прийняття рішень на основі нейромереж*. Інформатика, 11(4). 2016. С. 256-267.
- 21.Гончаренко, О. *Експертні системи в підтримці прийняття рішень*. Технічні науки, 9(3). 2018. С. 190-201.
- 22.Литвиненко, В. *Оцінка ефективності систем підтримки прийняття рішень*. Економіка та управління, 14(2). 2017. С. 102-115.
- 23.Demarest M. *Building the Data Mart* / M. Demarest. - DBMS. - 1994. - № 7. - P. 44-50.
- 24.Федоренко, М. *Бізнес-аналітика в системах підтримки прийняття рішень*. Інформаційні системи, 6(1). 2019. С. 66-77.
- 25.Савченко, Ю. *Методи оптимізації в системах підтримки прийняття рішень*. Оптимізація процесів, 7(2). 2020. С. 88-99.

- 26.Шевченко, Т. *Використання алгоритмів машинного навчання в системах підтримки прийняття рішень*. Інтелектуальні системи, 10(3). 2016. С. 134-145.
- 27.Marakas, G. M. *Decision support systems in the twenty-first century*. 1999.
- 28.Scott, M. S. *Management Decision Systems: Computer-based Support for Decision Making*. 1971.
- 29.Іваненко, О. *Онтології в інформаційних системах: поняття та класифікація*. Вісник Київського національного університету, 23(1). 2018. С. 45-50.
- 30.Петренко, Л. *Розвиток онтологій в управлінні знаннями*. Наукові праці Національного університету "Острозька академія", 12(2). 2019. С. 112-118.
- 31.Сидоренко, В. *Методи побудови онтологій для семантичного вебу*. Інформаційні технології та комп'ютерне моделювання, 15(3). 2020. С. 34-40.
- 32.Коваленко, І. *Онтологічний підхід до моделювання знань в інтелектуальних системах*. Кібернетика та системний аналіз, 53(4). 2017. С. 25-32.
- 33.Мельник, Р. *Онтології як інструмент інтеграції даних в наукових дослідженнях*. Науковий вісник Ужгородського університету, 18(1). 2016. С. 89-94.
- 34.Григоренко, Ю. *Основи побудови онтологій в системах управління знаннями*. Вісник Харківського національного університету, 22(2). 2015. С. 56-61.
- 35.Литвиненко, О. *Роль онтологій у сучасних інформаційних системах*. Наукові записки Національного університету "Острозька академія", 11(1). 2014. С. 102-107.
- 36.Шевченко, Т. *Онтології в семантичному вебі: принципи побудови та розвиток*. Вісник Львівського національного університету, 20(1). 2011. С. 120-125.
- 37.Савченко, В. *Онтології в інформаційному пошуку: поняття та застосування*. Інформаційні системи та мережі, 14(3). 2012. С. 65-70.
- 37.Gruber T. R. *A translation approach to portable ontology specifications*. / T. R. Gruber // Knowledge Acquisition. – 1993. – Vol. 5. – P. 199–220.

38. Федорчук, М. *Онтологічні моделі в системах прийняття рішень*. Комп'ютерні науки та інженерія, 17(4). 2013. С. 78-84.
39. Noy, N. F. *Ontology mapping*. In Handbook on ontologies. Springer, Berlin, Heidelberg. 2009. PP. 573-590.
40. Katsumi, M., & Grüninger, M. *What is ontology reuse?* In FOIS. 2016.
41. Katsumi, M., & Grüninger, M. *Choosing ontologies for reuse*. Applied Ontology, 12(3-4). 2017. PP. 195-221.
42. Ricardo, F. *Ontology pattern languages*. 2016.
43. Reginato, C. C., & Dos Santos, V. A. *Pattern-Driven Semantic System Integration*. Ninth International Conference on Complex, Intelligent, and Software Intensive Systems. 2015. PP. 89-96.
44. *Fallibilism. The Basics of Philosophy*.
https://www.philosophybasics.com/branch_fallibilism.html.
45. Herre, H. *General Formal Ontology (GFO): A foundational ontology for conceptual modelling*. Theory and applications of ontology: computer applications. Springer, Dordrecht. 2010. C. 297-345.
46. Olive, A. *Conceptional Modeling of Information Systems*. / Olive Antony. Springer Berlin Heidelberg. 2007. P. 471.
47. Heflin, Jeff, and James Hendler. *Semantic interoperability on the web*. Maryland univ college park dept of computer science. 2000.
48. Klein, M. *Combining and relating ontologies: an analysis of problems and solutions*. 2001.
49. Euzenat, J. *Algebras of ontology alignment relations*. The Semantic Web-ISWC 2008: 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings 7. PP. 387-402.
50. Euzenat, J., David, J., Locoro, A. & Inants, A. *Context-based ontology matching and data interlinking*. 2015.

51. Nicola Guarino, Daniel Oberle, and Steffen Staab. *What is an ontology?* / Handbook of ontologies.
52. Bhatt, Mehul, Wenny Rahayu, Sury Prakash Soni, and Carlo Wouters. *OntoMove: A knowledge based framework for semantic requirement profiling and resource acquisition*. Australian Software Engineering Conference (ASWEC'07). 2007. PP. 137-146.
53. Wouters, C., Dillon, T., Rahayu, W. and Chang, E. *A practical walkthrough of the ontology derivation rules*. International Conference on Database and Expert Systems Applications. Springer, Berlin, Heidelberg. 2002. PP. 259-268.
54. C. D. Vescovo, B. Parsia, U. Sattler, T. Schneider. *The Modular Structure of an Ontology: Atomic Decomposition*. IJCAI. 2011.
55. Noy N., McGuinness D. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL - 01-05 and Stanford Medical Informatics Technical Report SN. 2000.
56. Горборуков В. В. *Використання онтологій у системах підтримки прийняття рішень* / В. В. Горборуков, О. Є. Стрижак, О. В. Франчук // Математичне моделювання в економіці: Зб. наук. пр. 2013. Вип. 3. С. 33-39.
57. Geeraerts, Dirk. *Prospects and problems of prototype theory*. Diacronia 4. 2016. PP. 1-16.
58. Hernandez N., Mothe J. *TtoO: une méthodologie de construction d'ontologie de domaine à partir d'un thésaurus et d'un corpus de référence*. Rapport interne, IRIT. 2006.
59. Li Niu. *Cognition-Driven Decision Support for Business Intelligence. Models, Techniques, Systems and Applications*. / Li Niu, Jie Lu, Guangquan Zhang.- Springer-Verlag. 2009. С. 235.
60. Катренко, А. В., В. В. Пасічник, and В. П. Пасько. *Теорія прийняття рішень*. К.: Видавнича група BHV 448. 2009.

61. Gangemi, A., Catenacci, C., Ciaramita, M., & Lehmann, J. *Good grid: A metaontology-based framework for ontology evaluation and selection*. In EON@WWW.
62. Simperl, E. P. *Reusing ontologies on the Semantic Web: A feasibility study*. Data Knowl. Eng., 68. 2009. PP. 905-925.
63. Gangemi, Aldo, Carola Catenacci, Massimiliano Ciaramita, and Jos Lehmann. *A theoretical framework for ontology evaluation and validation*. In SWAP. 2005. Vol. 166, P. 16.
64. Wang, Xia, Yihong Ding, and Yi Zhao. *Similarity Measurement about Ontology-based Semantic Web Services*. In SemWS. 2006.
65. Chandramohan, D., Shailesh K., and Ashokkumar, S. *A study of finding similarities in web service using metrics*. Publication in International Journal of Scientific & Engineering Research (IJSER) 2, no. 6. 2011.
66. Rahim, N. R., Nordin, S. & Dom, R. M. *A Clinical Decision Support System based on Ontology and Causal Reasoning Models*. Jurnal Intelek, 14(2). 2019. PP. 187-197.
67. Xia, W., Ding, Y., Zhao Y. *Similarity Measurement about Ontology-based Semantic Web Services*. In SemWS. 2006.
68. Miah, S.J., Gammack, J. & Kerr, D. *Ontology development for context-sensitive decision support*. Semantics, Knowledge and Grid, Third International Conference. IEEE. 2007. C. 475-478.
69. Sánchez, D., Batet, M., Isern, D. & Valls, A. *Ontology-based semantic similarity: A new featurebased approach*. Expert Syst. Appl. 2012.
70. Prasenjit Mitra and Gio Wiederhold. *An Ontology-Composition Algebra*. // Handbook on Ontologies; eds. S. Staab, R. Studer. Berlin; Heidelberg; New York: Springer-Verlag. 2003. PP. 93–113.
71. Raz et al. *Fast and Efficient Context-Aware Services*. John Wiley & Sons. 2006.

72. Prasenjit, M. & Gio, W. *An Ontology-Composition Algebra*. 2003.
73. Niu, L., Lu, J., Zhang, G. *Cognition-Driven Decision Support for Business Intelligence. Models, Techniques, Systems and Applications*. Springer-Verlag. 2009. P. 235.
74. Іванов, С. *Інноваційне підприємництво: виклики та можливості*. Український журнал економіки, 4(1). 2018. С 22-28.
75. Мельник, О. *Стратегії розвитку підприємництва в умовах глобалізації*. Економіка та управління, 8(2). 2020. С. 34-40.
76. Takai, J. *The role of innovation in business strategy*. Journal of Business Innovation, 12(1). 2013. С. 45-60.
77. Савченко, Т. *Управління ризиками в бізнес-процесах*. Український журнал управління, 8(2). 2020. С. 45-51.
78. Aditya Das. *Artificial intelligence and decision support systems*. 2013.
79. Tarapata, Z. *Multicriteria weighted graphs similarity and its application for decision situation pattern matching problem*. Proceedings of the 13th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics. 2007. PP. 1149-1155.
80. Катренко, А. В., & Пасічник, В. В. *Прийняття рішень: теорія та практика*. Львів: «Новий Світ–2000». 2013.
81. Mitra, P., & Wiederhold, G. *An ontology-composition algebra*. Handbook on Ontologies. 2004. PP. 93–113.
82. Diaz, A., Motz, R., & Rohrer, E. *Making ontology relationships explicit in a ontology network*. AMW. 2011. P. 749.
83. Savić, M., Ivanović, M., Jain, L. C., Savić, M., Ivanović, M., & Jain, L. C. *Analysis of Ontology Networks*. 2019. PP. 143–175. Springer International Publishing. 2018. PP. 187–196.

84. Bonacin, R., Calado, I., & dos Reis, J. C. *A Metamodel for Supporting Interoperability in Heterogeneous Ontology Networks*. Springer International Publishing. 2018. PP. 187–196.
85. Zavadskas, E. K., Mardani, A., Turskis, Z., Jusoh, A., & Nor, K. M. *Development of TOPSIS method to solve complicated decision-making problems - An overview on developments from 2000 to 2015*. International Journal of Information Technology & Decision Making, 15(03). 2016. PP. 645-682.
86. *International standard industrial classification*.
87. *United Nations Standard Products and Services Code (UNSPSC)*.
88. D. Hofstadter. *Fluid Concepts and Creative Analogies Computer Models of the Fundamental Mechanisms of Thought*. Basic books. 1995.
89. Hofstadter, Douglas R. *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*. Basic books. 1995.
90. Hampton, James A. *Concepts as prototypes*. Psychology of learning and motivation 46. 2006. PP. 79-113.
91. Geeraerts, Dirk. *Prospects and problems of prototype theory*. Diacronia 4. 2016. PP. 1-16.
92. J. Lawry, Y. Tang. *Uncertainty Modelling for Vague Concepts: A Prototype Theory Approach*. Artificial Intelligence 173, no. 18. 2009. PP. 39–58.
93. Bouquet, Paolo, Fausto Giunchiglia, Frank Van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. *Contextualizing ontologies*. Journal of Web Semantics 1, no. 4. 2004. PP. 325-343.
94. Benslimane, D., Arara, A., Falquet, G., Maamar, Z., Thiran, P., Gargouri, F. *Contextual Ontologies*. In: Yakhno, T., Neuhold, E.J. (eds) *Advances in Information Systems*. ADVIS 2006. Lecture Notes in Computer Science, vol 4243. 2006.

95. Bouquet, Paolo, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. *C-owl: Contextualizing ontologies*. International Semantic Web Conference. Springer, Berlin, Heidelberg. 2003. PP. 164-179.
96. V. Pino, O. Erickson, R. Nascimento, M. Campos. *Introducing the Structural Bases of Typicality Effects in Deep Learning*. Image and Vision Computing 113. 2021.
97. F. Costello, M. Keane. *Efficient Creativity: Constraint-guided Conceptual Combination*. Cognitive Science 24, no. 2. 2000. PP. 299–349.
98. Aldo, G., Catenacci, C., Ciaramita, M., Lehmann, J. *A theoretical framework for ontology evaluation and validation*. SWAP, vol. 166. 2005. P. 16.
99. Міщенко Н. *Штучний інтелект-виклик часу* / Науковий світ. № 10. 2006. С. 12-13.
100. Лубко Д. В. Шаров С. В. *Методи та системи штучного інтелекту* / Навчальний посібник: Напрямки використання штучного інтелекту. 2019. С. 16-25.
101. Jarred Capellman. *Hands-On Machine Learning with ML.NET: Getting started with Microsoft ML.NET to implement popular machine learning algorithms in C#*. 2020.
102. Dino Esposito. *Programming ML.NET (Developer Reference)*. 2022.
103. T. Erl. *SOA Principles of Service Design*. Prentice Hall PTR. 2007.

ДОДАТОК А
СПИСОК ПУБЛІКАЦІЙ ТА АПРОБАЦІЯ РЕЗУЛЬТАТІВ

Наукові праці, в яких опубліковані основні результати дисертації:

1. Карпов І., Буров Є. *Використання онтологічних мереж у системах підтримки прийняття рішень в умовах неоднозначності*. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2020. Вип. 7. С. 8–15.
2. Burov Y., Mykich K., Karpov I. *Building a versatile knowledge-based system based on reasoning services and ontology representation transformations*. International Scientific and Technical Conference "Computer Sciences and Information Technologies". 2020. 15th IEEE Intern. sci. and techn. conf. on computer sciences and information technologies CSIT 2020, Lviv-Zbarazh, Ukraine, 23-26 Sept. 2020. Vol. 2. P. 255–260.
3. Карпов І., Буров Є. *Онтології у процесі прийняття рішень*. Вісник Хмельницького національного університету. Серія: «Технічні науки». 2023. № 2, т. 1. С. 149–153.
4. Burov Y., Karpov I. *Contextual concept meaning alignment based on prototype theory*. CEUR Workshop Proceedings. 2023. Vol. 3403 : Computational linguistics and intelligent systems 2023 : proc. of the 7th Intern. conf. on computational linguistics and intelligent systems. Vol. III: Intelligent systems workshop. Kharkiv, Ukraine, Apr. 20-21, 2023. P. 137–146.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Буров Є., Карпов І. *Вибір концептуалізації проблемної ситуації інтелектуальним агентом в задачах прийняття рішення*. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2023. Вип. 13.
2. Карпов І., Буров Є. *Підхід до побудови системи підтримки прийняття рішень на основі онтологій*. Scientific Collection «InterConf». 2022. № 99 : Proc. of the

- 4th Intern. sci. and pract. conf. "International scientific discussion: problems, tasks and prospects", Brighton, Great Britain, Febr. 19-20, 2022. С. 849–855.
3. Карпов І., Буров Є. *Особливості застосування онтологій в системах підтримки прийняття рішень: теоретичний та практичний аспект*. Грааль науки. 2022. № 12-13. Proc. of the III Correspondence intern. sci. and pract. conf. «An integrated approach to science modernization: methods, models and multidisciplinary», Apr. 29th, 2022. С. 307–310.
 4. Карпов І., Буров Є. *Застосування онтологій для прикладних задач*. III CISP Conference «Globalization of scientific knowledge: international cooperation and integration of sciences» III Міжнародна науково-практична конференція. 2021. Вінниця - Відень. С. 344–345.
 5. Карпов І., Буров Є. *Онтологічне моделювання: цілі та засоби*. Scientific Collection «InterConf», (107): with the Proceedings of the 12th International Scientific and Practical Conference «Science and Practice: Implementation to Modern Society». 2022. Manchester, Great Britain. С. 517–519.
 6. Карпов І., Буров Є. *Особливості використання онтологій у системах підтримки прийняття рішень: закономірності та особливості*. I International Scientific and Theoretical Conference «Technologies and strategies for the implementation of scientific achievements» I Міжнародна науково-теоретична конференція. European Scientific Platform. 2022. Стокгольм, SWE. С. 76–77.
 7. Карпов І., Буров Є. *Проблеми онтологічного підходу у віртуальному навчальному середовищі*. III International Scientific and Practical Conference «Education and science of today: intersectoral issues and development of sciences». III Міжнародна науково-практична конференція. 2022. Кембридж, GBR. С. 187–188.
 8. Karpov I., Burov Y. *Use of ontological networks in decision support systems under ambiguity*. Debats scientifiques et orientations prospectives du developpement

scientifique : materiales de la conferencia internacional científica y práctica -
"ΛΟΓΟΣ". Paris. Vol. 3. 2021. C. 97–103.

ДОДАТОК Б
АКТИ ВПРОВАДЖЕННЯ

ЗАТВЕРДЖУЮ

Директор ТОВ „Інтелектуальні
Вендингові системи”« » Тарас КОЗАК
« » 2023 р.**АКТ****про впровадження результатів
дисертаційної роботи аспіранта кафедри «Інформаційні системи та мережі»
Національного університету «Львівська політехніка»
Карпова Ігоря Андрійовича**

Цей акт підтверджує, що результати дисертаційної роботи на здобуття наукового ступеня доктора філософії Карпова І. А., були використані та впроваджені в практичну діяльність при розробці інтелектуальної системи підтримки прийняття рішень для роботи з клієнтами. Автором було розроблено методи та побудовано СППР на основі онтологічних мереж з використанням штучного інтелекту.

Терміни проведення досліджень: вересень 2022 р. - грудень 2022 р.

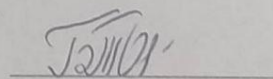
Впровадження отриманих результатів дисертаційного дослідження Ігорем Карповем полягає у наступному:

- аналіз систем підтримки прийняття рішень та методів концептуалізації проблемних областей у них;
- розробка методів онтологічного моделювання предметних областей у системах підтримки прийняття рішень;
- реалізація та апробація розроблених методів.

Використання розроблених у дисертаційному дослідженні методів та засобів дало змогу побудувати та реалізувати інтелектуальну систему підтримки прийняття рішень для роботи з клієнтами.

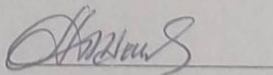
Члени комісії:

к.т.н., доцент кафедри ICM



Тарас БАСІЮК

к.т.н., доцент кафедри ICM



Андрій ВАСИЛЮК

ЗАТВЕРДЖУЮ



Проректор з науково-педагогічної роботи
Національного університету
«Львівська політехніка»

Олег ДАВИДЧАК

2023 р.

АКТ

**про впровадження в навчальний процес результатів
дисертаційної роботи на здобуття наукового ступеня доктора філософії
Карпова Ігоря Андрійовича**

Цей акт складений про те, що результати дисертаційної роботи Карпова І. А. на тему «Побудова системи підтримки прийняття рішень на основі онтологічних мереж», представленої на здобуття наукового ступеня доктора філософії, використовуються у навчальному процесі кафедри «Інформаційні системи та мережі» Національного університету «Львівська політехніка». Матеріали дисертаційного дослідження використовуються під час викладання дисциплін «Методології системного аналізу» та «Моделювання процесів аналізу даних».

Зокрема, у навчальному процесі використовуються запропоновані І. А. Карповим:

- Методи онтологічного моделювання систем підтримки прийняття рішень (дисципліна «Методології системного аналізу» для студентів освітньо-кваліфікаційного рівня «бакалавр», що навчаються за напрямом 6.124.00.00 «Системний аналіз», тема 12 «Системи та моделі»);

- Методи аналізу бізнес процесів у системах підтримки прийняття рішень (дисципліна «Моделювання процесів аналізу даних» для студентів освітньо-кваліфікаційного рівня «магістр», що навчаються за напрямом 8.124.00.03 «Аналіз даних (Data Science)», тема 4 «Методи моделювання процесів аналізу даних»).

Директор ІКНІ,
д.т.н., професор

Микола МЕДИКОВСЬКИЙ

Завідувач кафедри ІСМ,
д.т.н., професор

Василь ЛИТВИН

Професор кафедри ІСМ,
д.т.н., професор

Андрій БЕРКО



"ЗАТВЕРДЖУЮ"

Проректор з наукової роботи
Національного університету
«Львівська політехніка»

Іван ДЕМИДОВ

2023 р.

використання наукових результатів дисертаційної роботи

Карпова Ігоря Андрійовича

«Побудова системи підтримки прийняття рішень на основі онтологічних мереж»,
представленої на здобуття наукового ступеня доктора філософії за спеціальністю 124 –
системний аналіз при виконанні держбюджетної НДР

Комісія в складі: голови комісії – начальника науково-дослідної частини, д.т.н., ст. докл. Небесного Р. В. та членів комісії – завідувача кафедри інформаційних систем та мереж, д.т.н., професора Литвина В. В., завідувача відділу науково-організаційного супроводу наукових досліджень, к.т.н. Лазько Г. В. і в. о. заступника начальника планово-фінансового відділу Фаст І. І., цим актом підтверджують, що результати дисертаційної роботи Карпова І. А., використовувалися при виконанні держбюджетної НДР: «Система підтримки прийняття рішень розпізнавання мультиспектральних образів на основі технологій машинного навчання та онтологічного підходу» (номер державної реєстрації 0120U102203), в якій І. А. Карповим було обгрунтовано підходи до конструювання систем підтримки прийняття рішень, які базуються на використанні мереж онтологій та надають можливість динамічного вибору концептів і відношень, що дозволяє створювати системи підтримки прийняття рішень, які потребують інтеграції даних з різних предметних областей та в умовах неоднозначності.

Голова комісії:

начальник науково-дослідної
частини, д.т.н., ст. докл.

Роман НЕБЕСНИЙ

Члени комісії:

зав. відділу науково-організаційного
супроводу наукових досліджень, к.т.н.

Галина ЛАЗЬКО

в. о. заст. нач. планово-фінансового відділу

Ірина ФАСТ

зав. каф. інформаційних систем та мереж,
д.т.н., проф.

Василь ЛИТВИН