

**НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ПРОБЛЕМ МОДЕЛЮВАННЯ В ЕНЕРГЕТИЦІ
ІМ. Г.Є. ПУХОВА**

Кваліфікаційна наукова праця
на правах рукопису

КОВАЛЬ ОЛЕКСАНДР ВАСИЛЬОВИЧ

УДК 004.82 : 004.89

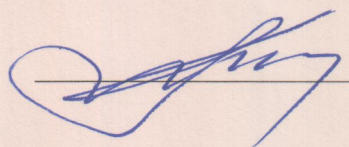
ДИСЕРТАЦІЯ

**МЕТОДИ ТА ЗАСОБИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ
СЦЕНАРІЇВ АНАЛІТИЧНОЇ ДІЯЛЬНОСТІ**

Спеціальність 01.05.02 – математичне моделювання та обчислювальні методи
Галузь знань – інформаційні технології

Подається на здобуття наукового ступеня доктора технічних наук

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

 О.В. Коваль

Науковий консультант – Додонов Олександр Георгійович
доктор технічних наук, професор

Київ – 2021

Анотація

Коваль О.В. Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності. – На правах рукопису.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 01.05.02 «Математичне моделювання та обчислювальні методи» – Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України, Київ, 2021.

В дисертаційній роботі вирішено важливу науково-прикладну проблему дослідження, формалізації та розробки теоретичних основ та практичних засад комп'ютерного моделювання сценаріїв аналітичної діяльності (АнД) в різних предметних областях, зокрема при проектуванні складних технічних систем.

Запропоновано теоретичний підхід до комп'ютерного моделювання багаторівневих сценаріїв АнД, який відрізняється від відомих тим, що оснований на поєднанні переваг візуального моделювання – графічної нотації BPMN з керованою моделлю і семантичного аналізу на базі онтологічної моделі предметної області задач АнД та подальшої серіалізації BPMN-моделі в програмне середовище, що зменшує кількість логічних розривів в комп'ютерній моделі сценарію АнД та суттєво зменшує час моделювання в цілому. В основі запропонованого підходу лежить метод перетворення BPMN-моделі в модель представлення предметної області сценарію АнД мовою OWL - OWL-модель, який відрізняється від відомих тим, що дозволяє проводити семантичний аналіз коректності логічно зв'язаних фрагментів сценарію вирішення задачі АнД та його верифікацію (особливо багаторівневих сценаріїв). Запропоновано інформаційну технологію конвертації BPMN-моделі в OWL-модель сценарію аналітичної діяльності, що дозволяє перетворювати метадані опису сценарію та його предметної області в базі знань, накопичуючи нові знання, що дає можливість побудови нових сценаріїв АнД, використовуючи вже існуючі в базі знань. Набули подальшого розвитку математична модель обчислення розподілу значень можливих подій сценарію та інтелектуальний метод визначення наступних подій сценарію АнД, який відрізняється від відомих тим, що дозволяє

виконувати формування дерева рішень для класифікації можливих наступних кроків сценарію, що дає можливість значно підвищити точність розбиття графу складного сценарію на відповідні гілки при обчисленні імовірності наступного кроку сценарію. Удосконалено метод реалізації сценаріїв АнД на основі Web-сервісів, який забезпечує можливість виконання завдань кінцевого користувача шляхом динамічного формування послідовності Web-сервісів. Набули подальшого розвитку інтелектуальний програмний засіб, що навчається на діях користувача та пропонує користувачу найбільш доцільні наступні кроки вирішення завдання АнД, та процес автоматизованої побудови сценаріїв АнД на основі бібліотеки сценаріїв (прецедентів) для різних предметних областей.

Реалізація запропонованих моделей, методів та комп'ютерних засобів в рамках тривірневої комп'ютерної моделюючої системи, призначеної для тестування і оцінювання запропонованого теоретичного підходу моделювання сценаріїв АнД, забезпечила можливість вирішення задачі побудови та реалізації сценаріїв як в лабораторних дослідженнях, так і в умовах застосування в реальних моделюючих системах.

ABSTRACT

Koval O.V. Methods and tools for computer modeling of analytical scenarios. – As the manuscript.

The thesis for obtaining a scientific degree of Doctor of Technical Sciences in specialty 01.05.02 – Mathematical modeling and computational methods, Pukhov Institute for Modelling in Energy Engineering of the NAS of Ukraine, Kyiv, 2021.

The thesis solves an important scientific and technical problem of research, formalization and development of theoretical and practical bases of computer modeling of analytical activity scenarios in various subject domains, at designing of complex technical systems.

A theoretical approach to computer modeling of multilevel analytical activity scenarios is proposed, which unlike the known ones is based on combining the advantages of visual modeling - graphic BPMN notation with controlled model and

semantic analysis based on ontological model of subject domain of analytical activity problems and further serialization of the BPMN model into a software environment that reduces the number of logical gaps in the computer model of the analytical activity scenario and significantly reduces the overall simulation time. The proposed approach is based on the method of transforming the BPMN-model into a model of representation of the subject domain of the analytical activity scenario with OWL language - OWL-model, which unlike the known ones allows to conduct a semantic analysis of the accuracy of logically related fragments of the analytical activity solution and its verification (especially in multilevel scenarios). The proposed information technology for converting BPMN-model into OWL-model of analytical activity scenario, which allows to convert metadata description of the scenario and its subject domain in the knowledge base, accumulating new knowledge, which allows to build new analytical activity scenario using already existing in the knowledge base.

An approach is proposed to solve the problem of ensuring the semantic compatibility of the processes of modeling the steps of a multilevel scenario of analytical activity through the use of the ontology of the subject area to resolve conflicts. A set of axioms and theorems that theoretically substantiate the conditions for ensuring the semantic compatibility of transitions is presented and proved. An approach to resolving differences that arise in transitions is proposed. This approach is based on the ontological model of the semantic mediator, which is applied similarly to the approach in Semantic Web Service technologies.

The mathematical model for calculating the distribution of values of possible scenario events and the intelligent method for determining subsequent events of the analytical activity scenario are further developed, which unlike the known ones allows to perform the formation of a decision tree to classify the possible next steps of the scenario and significantly increases the accuracy of splitting the graph of a complex scenario into appropriate branches when calculating the probability of the next step of the scenario. The method of implementing the analytical activity scenario based on Web-services has been improved, which provides the ability to perform end-user tasks by dynamically forming a sequence of Web-services. Further developed are an

intelligent software tool, which is taught by the user's actions and offers the user the most appropriate next steps to solve the analytical activity problem, and the process of automated construction of analytical activity scenarios based on a library of scenarios (precedents) for different subject domains.

The implementation of the proposed models, methods and computer tools in a three-tier computer modeling system designed to test and evaluate the proposed theoretical approach to modeling the analytical activity scenarios, provided an opportunity to solve the problem of building and implementing scenarios in laboratory research and in terms of application in real modeling systems.

The offered and created in work algorithms and software for processes of modeling the scenarios of analytical activity provide computer realization and interaction of the received models for graphic representation of analytical activity scenarios, semantic models of subject domain of analytical activity problems and knowledge of process for construction and execution of analytical activity scenarios as adaptive object model. The developed intelligent software tools allow to automate the construction of the analytical activity scenario taking into account its previous steps and on the basis of derivation of precedents that have already been accumulated as a result of preliminary modeling of analytical activity scenarios. This approach reduces the logical errors and semantic incoherence of the description of multi-level scenarios or their individual branches and thus leads to a significant reduction of modeling time of analytical scenarios.

The created methods and tools of mathematical and computer modeling of analytical activity scenarios are used to solve the problem of construction and realization of modeling scenarios for functioning of measuring hydroacoustic system to improve the algorithms of processing the hydroacoustic signals; scenarios of energy supply distribution for optimization the energy consumption; scenarios of interactive processing of multidimensional data; construction and implementation of scenarios of analytical activities in the problems of spatial modeling; monitoring of financial indicators of budget execution at all levels.

The results of the thesis research were implemented in the Subsidiary Enterprise “National Development Fund”, LLC “Research and Production Enterprise “Symbol”, LLC “Engineering Logic”, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

Список опублікованих праць за темою дисертації

16. 1. Додонов А.Г., Сенченко В.Р., Коваль А.В. Аналитика и знания в компьютерных системах. Киев: ИПРИ НАН Украины, «КПИ имени Игоря Сикорского», 2020. 315 с.
17. Додонов О.Г., Коваль О.В., Глоба Л.С., Бойко Ю.Д. Комп'ютерне моделювання інформаційно-аналітичних систем, Київ: ИПРИ НАН України, 2017. 239 с.
18. Kuzminykh V., Koval O., Melnyk U., Otroh S. Evaluating the Quality of Modeling the Scenario of Information Analysis on a Branched Network. *Сучасний захист інформації*. 2019. № 3(39). С. 70–76.
19. Додонов О.Г., Сенченко В.Р., Коваль О.В., Бойченко А.В. Моделювання сценаріїв аналітичної діяльності на основі нотації BPMN OWL. *Реєстрація, зберігання і обробка даних*. 2020. Т. 22, № 1. С. 31–48.
20. Koval O.V., Kuzminykh V.O., Svistunov S.Y., Xu Beibei, Zhu Shiwei. Data collection for analytical activities using adaptive microservice architecture. *Реєстрація, зберігання і обробка даних*. 2021. Т. 23, № 1. С. 11–31.
21. Додонов О.Г., Коваль О.В., Сенченко В.Р., Швайко В.Г. Формування та реалізація сценаріїв аналітики в задачах просторового моделювання. *Реєстрація, зберігання і обробка даних*. 2020. Т. 22, № 3. С. 39–57.
22. Kuzminykh V., Koval O., Otrokh S. Refining the typical scenarios by additional factors. *Математичне та комп'ютерне моделювання. Серія: Технічні науки*. 2019. Вип. 20. С. 68–78.
23. Додонов О.Г., Коваль О.В., Сенченко В.Р., Шпурик В.В. Автоматизована система формування сценарію аналітичної діяльності. *Реєстрація, зберігання і обробка даних*. 2019. Т. 21, № 1. С. 11–22.
24. Novogrudska R., Globa L., Koval O. The Simplification Method of Engineering Task Sequences used for Engineering Knowledge Portals. *Вісник Харківського Національного університету. Серія: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2018. Вип. 37. С. 37–44.
25. Koval O.V., Kuzminykh V.A., Khaustov D.V. Using stochastic automaton for data consolidation. *Наукові вісті НТУУ «КПІ»*. 2017. № 2. С. 29–36.

26. Коваль А.В., Бойко Ю.Д., Волкова Е.А. Особенности сценарно-целевого подхода к анализу объектов действенной аналитики. *Системні дослідження та інформаційні технології*. 2015. № 1. С. 57–67.

27. Коваль О.В., Кузьмініх В.О. Реалізація сценарного підходу в управлінні проектами на основі типових задач. *Реєстрація, зберігання і обробка даних*. 2015. Т. 17, № 1. С. 77–87.

28. Коваль О.В., Зайцева А. Верифікація комп'ютерної моделі системи інформаційного управління. *Вісник НТУУ «КПІ». Серія: Інформатика, управління та обчислювальна техніка*. 2014. № 61. С. 43–48.

29. Koval O.V., Zaytseva K.A., Boyko Yu.D. Formation of Analytical activity Scenarios. *Системні дослідження та інформаційні технології*. 2014. № 1. С. 20–25.

30. 15. Діденко О.О., Сенченко В.Р., Коваль О.В. Система моніторингу державного бюджету України. *Наукові вісті НТУУ «КПІ»*. 2013. № 6. С. 37–48.

31. Koval A.V., Kuzminykh V.A., Voronko M.P., Khaustov D.V. Development of a Scenario-Based Project Management System Construction in Enterprises with the Functional Organizational Structure. *Informatyka Automatyka Pomiaru w Gospodarce i Ochronie Środowiska*. 2013. No. 4. P. 26–30.

32. Кузьмініх В.О., Коваль О.В., Воронько М.П. Оцінка часу виконання типових задач проектів на підприємствах з функціональною організаційною структурою, *Реєстрація, зберігання і обробка даних*. 2012. Т. 14, № 3, С. 70–80.

33. Кузьмініх В.О., Коваль О.В., Хаустов Д.В., Коростельова Є.Ю. Управління агрегованими групами проектів, *Реєстрація, зберігання і обробка даних*. 2011. Т. 13, № 3. С. 109–116.

34. Коваль О.В. Узагальнена архітектура аналітичної складової корпоративних інформаційно-аналітичних систем. *Реєстрація, зберігання і обробка даних*. 2011. Т. 13, № 2. С. 53–73.

35. Додонов О.Г., Коваль О.В., Дзюбаненко Р.І., Цепков П.А., Маюров М.О., Жидовленко Ю.О. Концептуальні рішення створення автоматизованої системи екстреної допомоги населенню за єдиним телефонним номером 112. *Реєстрація, зберігання і обробка даних*. 2010. Т. 12, № 2. С. 165–180.

36. Додонов О.Г., Коваль О.В., Сенченко В.Р. Методологія побудови корпоративних інформаційно-аналітичних систем. *Реєстрація, зберігання і обробка даних*. 2007. Т. 9, № 4. С. 60–75.

37. Kuzminykh V., Koval O., Xu B., Zhu Sh. Microservice architecture of the system for assessing the level of international activity. *International scientific journal «Internauka»*. 2021. No. 6. С. 67–79.

38. Globa L., Novograduska R., Koval O. The Approach to Users Tasks Simplification on Engineering Knowledge Portals. *Advances in Intelligent Systems and Computing*. 2018. Vol. 889. P. 150–158. (Scopus).

39. Globa Larysa, Novograduska Rina, Koval Alexander, and Senchenko Vyacheslav. *Ontology for Applications Development. IntechOpen*. 2018. Ch. 2. P. 29–53.
40. Globa Larysa, Novograduska Rina, Koval Alexander and Senchenko Vyacheslav. *Examples of Ontology Model Usage in Eengineering Fields. IntechOpen*. 2018. Ch. 3. P. 55–81.
41. Koval O., Globa L., Novograduska R. The approach to web services composition. *Advances in Intelligent Systems and Computing*. 2017. Vol. 534. P. 293–304. (Scopus).
42. O. Koval, V. Kuzminykh, S. Otrokh and V. Kravchenko. «Optimization of Scenarios for Collecting Information Streaming Wide-Area Network», 3rd International Conference on Advanced Information and Communications Technologies (AICT), 2-6 July, 2019, Lviv. 2019. P. 213–215. (Scopus)
43. Oleksandr Koval, Valeriy Kuzminykh and Maksym Voronko. «Standard Analytic Activity Scenarios Optimization based on Subject Area Analysis», in Selected Papers of the XIX International Scientific and Practical Conference «Information Technologies and Security» (ITS 2019) CEUR Workshop Proceedings, Vol. 2577, P. 37–46, 2019. (Scopus).
44. Aleksandr Koval, Safwan Al Salaimh and Oleh Andriichuk. «Usage of Expert Classification in Diagnostic Expert Systems' Knowledge Bases Construction», in Selected Papers of the XVIII International Scientific and Practical Conference «Information Technologies and Security» (ITS 2018), CEUR Workshop Proceedings, Vol. 2318, P. 139–149, 2018. (Scopus).
45. V. O. Kuzminykh, A. V. Koval and M. V. Osypenko. «Methods of Machine Training on the Basis of Stochastic Automatic Devices in the Tasks of Consolidation of Data from Unsealed Sources», in Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security (ITS 2017) CEUR Workshop Proceedings, Vol. 2067, P. 63–68, 2017. (Scopus)
46. L. S. Globa, R. L. Novograduska, and A. V. Koval. «Ontology Model of Telecom Operator Big Data», in 2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 4–7 June 2018 Batumi, Georgia. IEEE, 2018. P. 1–5.
47. R. L. Novograduska, L. S. Globa, O. V. Koval, and V. R. Senchenko. «Ontology model of intelligent modeling system for marine facilities identification», in International Conference Radio Electronics & Info Communications (UkrMiCo'2017), 11-15 September 2017, Kyiv. Odesa, 2017. P. 536–541. (Scopus).
48. V. R. Senchenko, and A. V. Koval. «The technology of semantic modeling for knowledge management systems in environment Protégé». *Материалы XVII Международной научно-практической конференции «Информационные технологии и безопасность (ITS-2017)»*, 30 ноября, 2017, Киев. Киев: ООО «Инжиниринг», 2017, вып. 17. С. 211–235.

49. Khazanovych Y., Gerayimchuk M., Koval A., Senchenko V. «Ontological Design of an Intelligent System for Studying Hydroacoustical Processes», IEEE International Conference on Power, Control, Signals & Instrumentation Engineering (ICPCSI-2017), Saveetha Engineering College, Chennai, India, 2017. P. 135–138.

50. V.R. Senchenko, O.V. Koval, L.S. Globa, and R.L. Novograduska. «Intelligent modeling system based on cloud-technology», International Conference Radio Electronics & Info Communications UkrMiCo 2016. IEEE, 2016.P. 449–452. (Scopus).

51. Глоба Л.С., Коваль О.В., Новогрудська Р.Л., Сенченко В.Р. Створення сценаріїв обробки даних на основі онтології. Матеріали 18-й Міжнародної науково-технічної конференції SAIT–2016 «Системний аналіз та інформаційні технології», 30 травня, 2016 р., Київ. Київ: ІПСА КПІ, 2016. С. 262–265.

52. O.V. Koval, and K.A. Volkova. Computer modelling of management information system with actionable analytics. Матеріали 17-ї міжнародної науково-технічної конференції SAIT–2015 «Системний аналіз та інформаційні технології», 25 червня, 2015 р., Київ. Київ: ІПСА КПІ, 2015. С. 32.

53. Коваль А.В., Бойко Ю.Д., Зайцева Е.А. Модель сценарно-целевого подхода при построении информационно-аналитической системы. Системный анализ и информационные технологии: сб. тезисов 16-ой междунар. науч.-техн. конф. (Київ, 26 мая 2014 р.). Київ: ННК «ІПСА», 2014. С.105-106.

54. Зайцева Е.А., Коваль А.В. Комп'ютерне моделювання соціально-економічного розвитку регіону. Системный анализ и информационные технологии: сб. тезисов 15-ой междунар. науч.-техн. конф. (Київ, 31 мая 2013 р.). Київ: ННК «ІПСА», НТУУ «КПІ», 2013. С. 103.

55. Коваль А.В., Сенченко В.Р. Построение системы анализа выполнения госбюджета на основе сценарного подхода. Системный анализ и информационные технологии: сб. тезисов 14-ой междунар. науч.-техн. конф. (Київ, 24 апр. 2012 р.). Київ: ІПСА НТУУ «КПІ», 2012. С. 202-203.

56. Коваль А.В., Діденко О.О. Імітаційне моделювання засобів підтримки діяльності в корпоративних інформаційно-аналітичних системах. Сучасні проблеми наукового забезпечення енергетики: зб. матеріалів ІХ міжн. наук.-практ. конф. аспірантів, магістрів, студентів. Київ. 2011. С. 239.

57. Калішук А.А., Коваль О.В. Автоматизація генерації сценаріїв аналітичної діяльності. Сучасні проблеми наукового забезпечення енергетики: збірник доповідей VII між нар. наук.-практ. конф. аспірантів, магістрів, студентів. Київ, 2009. С. 135.

58. Додонов О.Г., Коваль О.В. Урядова інформаційно-аналітична система з питань надзвичайних ситуацій. Системи підтримки прийняття рішень. Теорія і практика: збірник доповідей наук.-практ. конф. з міжнародною участю. Київ: ІПММС НАНУ, 2007. С. 29–32.

59. Коваль А.В., Сенченко В.Р. Система моніторингу фінансово-економічних та макроекономічних показників як ефективний інструмент

прийняття управлінських рішень. Проблеми моніторингу та забезпечення економічної безпеки держави: збірник наукових праць наук.-практ. семінару з міжнародною участю. Київ, 2007. С. 58–60.

ЗМІСТ.....	11
ВСТУП	17
РОЗДІЛ 1. ПРОБЛЕМИ ПОБУДОВИ КОМП'ЮТЕРНИХ СИСТЕМ МОДЕЛЮВАННЯ СЦЕНАРІЇВ АНАЛІТИЧНОЇ ДІЯЛЬНОСТІ.....	33
1.1 Особливості аналітичної діяльності для різних предметних областей	33
1.2 Визначення сценарію як елемента аналітичної діяльності.....	45
1.3 Методологічні засади моделювання сценаріїв аналітичної діяльності	57
1.4 Аналіз сучасного стану підходів до моделювання процесів обробки даних	65
1.5 Невирішені проблеми підвищення якості формування сценаріїв аналітичної діяльності.....	77
1.6 Постановка задач дослідження.....	80
1.7 Висновки до першого розділу.....	81
РОЗДІЛ 2. ТЕОРИТИЧНІ ЗАСАДИ МОДЕЛЮВАННЯ СЦЕНАРІЇВ АНАЛІТИЧНОЇ ДІЯЛЬНОСТІ НА ОСНОВІ ОНТОЛОГІЧНИХ ТА ПРОЦЕСНИХ МОДЕЛЕЙ	84
2.1 Адаптація концепції процесного моделювання до побудови сценаріїв аналітичної діяльності	84
2.1.1 Мови та нотації моделювання процесів	86
2.1.2 Базові положення нотації BPMN 2.0.....	91
2.1.3 Методологія побудови сценаріїв аналітичної діяльності на основі OWL та нотації BPMN.....	94
2.2 Теоретичні засади семантичного моделювання сценаріїв аналітичної діяльності.....	100
2.3 Особливості формування онтології для моделювання сценарію в графічній нотації	106

2.4	Створення мета-моделі онтології нотації BPMN 2.0.....	111
2.5	Критерії повноти опису для забезпечення співвідношення між різними моделями сценаріїв	119
2.6	Оцінка результату конвертації моделі BPMN в OWL-файл.....	121
2.7	Висновки до другого розділу	129
РОЗДІЛ 3. ІНТЕЛЕКТУАЛІЗАЦІЯ ПРОЦЕСУ ФОРМУВАННЯ СЦЕНАРІЮ АНАЛІТИЧНОЇ ДІЯЛЬНОСТІ		130
3.1	Забезпечення семантичної сумісності процесів складного сценарію	130
3.1.1	Теоретичне обґрунтування вирішення невідповідностей при взаємодії кроків сценарію.....	135
3.1.2	Типовий алгоритм встановлення взаємодії між двома процесами сценарію	139
3.2	Парадигма інтелектуалізації кроків сценаріїв в процесі аналітичної діяльності	141
3.3	Інтегрована модель опису знань для побудови сценаріїв аналітичної діяльності	143
3.3.1	Задання ваг важливості понять онтології для задач семантичного типу	150
3.3.2	Методи задання ваг важливості понять онтології на основі інтелектуального аналізу даних.....	152
3.4	Інтелектуалізація програмних засобів виконання сценарію аналітичної діяльності на основі методу машинного навчання	154
3.5	Моделювання сценаріїв аналітичної діяльності на основі онтології та прецедентів.....	166
3.6	Формальна модель сценарію формування висновку на основі нечіткої онтології.....	173
3.7	Висновки до третього розділу.....	177
РОЗДІЛ 4. МЕТОДИ ВИРІШЕННЯ ЗАДАЧ ВИКОНАННЯ СЦЕНАРІЇВ аналітичної діяльності		180

4.1	Метод виконання сценаріїв АнД на основі різноманітних Web-сервісів	180
4.2	Методи та моделі формування сценаріїв аналітичної діяльності на порталах знань.....	186
4.2.1	Модель функціонування порталу знань на основі сценарного підходу	186
4.2.2	Формальна алгебраїчна система виконання сценаріїв на порталі знань	191
4.2.3	Метод динамічного формування сценаріїв на порталі знань.	197
4.3	Метод семантичної верифікації знань Web-контенту.....	200
4.4	Модель користувача інтегрованої платформи моделювання сценаріїв аналітичної діяльності	208
4.5	Висновки до четвертого розділу.....	212
Розділ 5. Комп'ютерна моделююча система моделювання сценаріїв АнД.....		215
5.1	Структура та архітектура комп'ютерної моделюючої системи (інтегрованого програмного інструментального середовища) для тестування і оцінки запропонованого теоретичного підходу моделювання сценаріїв АнД	215
5.2	Структура семантичної інтерпретації знань при моделюванні сценаріїв АнД.....	226
5.3	Визначення та формування словника відносин та зв'язування класів	230
5.4	Висновки до розділу	236
Розділ 6. Аналіз переваг застосування онтологічних моделей для формування сценаріїв та оцінка якості та ефективності моделювання сценаріїв аналітичної діяльності.....		238
6.1	Оцінка якості моделі онтології та бази знань на якість сценарію аналітичної діяльності.....	238

6.2 Оцінка якості та ефективності сценарію, побудованого на онтології 245

6.3 Дослідження впливу виразних можливостей мов SPARQL та DL – Query на ергономічність програмних засобів, які забезпечують моделювання сценарію АнД 266

6.4 Висновки до шостого розділу 271

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 277

ДОДАТОК А. Побудова та реалізація сценаріїв моделювання процесів функціонування вимірювальної гідроакустичної системи 300

A1. Створення онтології предметної області програмного комплексу моделювання функціонування вимірювальної системи в морському середовищі 302

A1.1 Формування підмножини атрибутів ПрО 310

A1.2 Формування екземплярів класів для наповнення бази знань 311

A1.3 Визначення та формування підмножини відносин та зв'язування класів 312

A1.4 Формування підмножини відносин у визначеній предметній області 313

A1.5 Утворення підмножини Axioms 314

A1.6 Формування відносин між окремими суб'єктами 316

A2. Методика моделювання функціонування ВС за сценарно-цільовим підходом 319

A3. Фрагмент сценарію ідентифікації об'єктів на основі мови запитів SPARQL-Query 326

ДОДАТОК Б. Комп'ютерне моделювання сценаріїв розподілу енергопостачання 332

B1. Концепція управління попитом на електроенергію з боку користувача 334

B2. Математична модель електроспоживання 336

B3. Побудова типового сценарію розподілу енергопостачання 337

Б4. Отримані результати планування енергоспоживання за використанням сценарію.....	344
ДОДАТОК В. Формування сценарію інтерактивної обробки багатовимірних даних	355
ДОДАТОК Г. Побудова та виконання сценаріїв аналітичної діяльності в задачах просторового моделювання.	371
Г.1 Формування типового сценарію GIS-аналітики при просторовому моделюванні наслідків природних явищ.....	373
Г.2 Виконання сценарію просторового моделювання наслідків надзвичайної ситуації.....	379
Г.3 Гідрологічне моделювання.....	381
Г.4 Просторове моделювання.....	382
ДОДАТОК Д Моніторинг фінансових показників виконання бюджетів всіх рівнів.....	388
Д1. Фрагмент програмного додатку конфігурації інтелектуальної системи «Моніторинг фінансових показників виконання бюджетів всіх рівнів»	388
Д2. Фрагмент складу базових модулів інтелектуальної системи «Моніторинг фінансових показників виконання бюджетів всіх рівнів»	391
Д3. Фрагмент програмного додатку формування сценаріїв аналітичної діяльності для системи «Моніторингу фінансових показників виконання бюджетів всіх рівнів»	392
Д4. Програмні засоби виконання типового сценарію автоматизованого вводу звітності до БД системи «Моніторингу фінансових показників виконання бюджетів всіх рівнів»	394
ДОДАТОК Е. Стандарти та засоби процесного моделювання.....	397
Е.1 Огляд сучасних засобів моделювання бізнес-процесів	397
Е.2 Платформа BizAgi, як засіб інтелектуального моделювання сценаріїв	403
Е.3 Основні елементи інтерфейсу користувача Process Modeler.....	407

ДОДАТОК Ж Побудова моделей обробки даних засобами програмної платформи RapidMiner	414
ДОДАТОК 3. Фрагмент програмної реалізації методу інтелектуальний метод визначення наступних подій сценарію аналітичної діяльності при формуванні наступного кроку сценарію	422
ДОДАТОК І Акти впровадження	427

ВСТУП

Актуальність теми.

Сьогодні ефективність прийняття управлінських рішень практично у всіх областях діяльності спеціалістів, у тому числі науковців та інженерів в процесі проектування складних технічних систем, на пряму залежить від наявності та якості інформації та знань, що аналізується. Але з кожним роком обсяг даних, інформації та знань, які потрібні для прийняття збалансованих рішень, збільшується в геометричній прогресії, причому процес аналізу цього обсягу даних все більш обмежується в часі, що відповідно змушує розробників програмних систем (ПС), які направлені на підтримку аналізу інформації, постійно вдосконалювати методи і технології отримання, структуризації та аналізу різноманітних даних.

З іншої сторони, згідно з дослідженнями всесвітньовідомої консалтингової компанії Gartner від 70% до 80% проектів, пов'язаних з впровадженням складних аналітичних рішень в організаціях та корпораціях, – провальні. Все це спонукає до подальшого пошуку шляхів удосконалення як наукових, так і технологічних підходів, які сприяють раціональній організації аналітичної діяльності (АнД) та підвищують якість інформації, що аналізується, та ефективність ПС в цілому. Починаючи з 2013 та ж компанія Gartner визначила дієву аналітику в десятці найбільш затребуваних інформаційних технологій. При цьому зазначається для того щоб аналітика була дієвою вона повинна мати змістовне наповнення, бути цілеспрямованою, зокрема надавати засоби моделювання та оптимізації сценаріїв аналізу інформації та розвитку ситуації або процесів.

Сучасний розвиток інформаційних технологій, в тому числі й тих, що вирішують задачі АнД, вирізняється рядом тенденцій, які насамперед обумовлені зрушенням від кастомізації, яка на тяжінні довгого часу була золотим стандартом, в сторону персоналізації. До таких тенденцій, які забезпечують цей зсув можна визначити накопичення та використання знань

про предметну область (ПрО) та накопичений попит вирішення завдань в ПС, інтелектуалізація як самих ПС так і інтелектуалізації за їх використанням процесів прийняття рішень [18], структурна алгоритмізація та як наслідок використання сервіс-орієнтованої архітектури при побудові ПС. Ці тенденції вимагають наукового осмислення та опрацювання, розробки нових архітектурних вирішень, включаючи й підходи до проектування, впровадження ПС, спрямованих на підтримку АНД.

Стійка тенденція до інтелектуалізації програмних компонентів, які реалізують певне коло аналітичних процесів, пояснюється наявністю як об'єктивних, так і суб'єктивних факторів. До об'єктивних факторів відноситься зростаюча складність самих процесів аналізу даних, і, як наслідок, сценаріїв їхньої обробки. Залежно від цілей і мети аналізу, а також природи даних (структуровані, неструктуровані, слабо структуровані, веб-дані, або навіть дані соціальних мереж), для аналізу можуть залучатися різноманітні методи обробки, наприклад, очищення, нормалізація, перетворення форматів, кластерний аналіз, навчання, моделювання, прогнозування, графовий аналіз, оцінка якості тощо. Різноманіття методів і технологій обробки само по собі потребує від аналітика як навичок їхнього застосування, так і спеціалізованих знань стосовно цільового використання. Крім того, для аналізу різних типів даних бажано використовувати саме ті сценарії аналітичної обробки, які відповідають як типу даних, так і цілям і меті аналізу. Суб'єктивні фактори асоціюються як з діями людини (експерта-аналітика) – кроками виконання процесів аналізу складної предметної області (ПрО), так і можливістю сприйняття та інтерпретації результатів аналітичного дослідження. Безумовно, як об'єктивні, так і суб'єктивні фактори впливають на ефективність методів аналізу та результати аналітичної діяльності (АНД) у цілому. Тому для певного кола аналітичних програмних систем (АНС) дуже важливо полегшити процеси моделювання та виконання складних сценаріїв за рахунок їхньої інтелектуалізації.

Одним із проявів інтелектуалізації є накопичення знань щодо особливостей функціонування АНС, включаючи і знання о діях аналітика при виконанні типових завдань. У цьому контексті сенс накопичення знань полягає в тому, щоб програмний засіб був у змозі самостійно класифікувати нові дані, одержувані при аналізі, і пропонувати користувачу найбільш доцільні кроки сценарію, виходячи із накопичених знань і ситуації, що склалася на попередніх кроках [2, 3]. Тобто бажано щоб сучасний аналітичний засіб мав можливість прогнозувати найбільш імовірний наступний крок складного сценарію та пропонувати його користувачеві, виходячи з наявних у системі знань та умов, що склалися в процесі виконання сценарію.

Сьогодні, аналітична діяльність – це сукупність дій на основі концепцій, методів, засобів, нормативно-методичних матеріалів для збору, накопичення, обробки і аналізу даних з метою обґрунтування та прийняття рішень, або генерації нових знань. Об'єктом АНД досить часто стають складні взаємопов'язані процеси, кожен з яких може характеризуватися значною кількістю параметрів, безперервністю змін, важкістю передбачення динаміки їхнього розвитку. Особливістю задач, які виникають при розв'язанні таких проблем, є відсутність чіткого алгоритму, який дозволяв би завжди досягти необхідного рішення.

Сценарій АНД, виходячи з парадигми інтелектуалізації програмних засобів, можна розглядати як певну структуру представлення знань, яка використовується для опису послідовності пов'язаних подій – у вигляді орграфа. В цьому підході орграф фактично визначає сукупність шляхів досягнення мети в конкретній стереотипній ситуації (pattern) для заданої ПрО, представленої у вигляді семантичної мережі, наприклад, онтології [10]. Тобто, для генерування сценарію АНД з елементами інтелектуалізації має формуватися орієнтований ациклічний граф можливих операцій сценарію (Directed Acyclic Graph – DAG). Слід нагадати, що DAG відображає припущення про взаємозв'язок між змінними – вузлами в контексті побудованого графу, в якому відсутні орієнтовані цикли [248]. При такому підході задача побудови сценарію

АнД фактично зосереджується на вирішенні відомої задачі пошуку найкоротшого шляху (Single Source Shortest Path) [249]. Отже, пошук найкоротшого шляху в оргграфі від витoku до його стоку з урахуванням накопичених знань про стереотипні дії користувача можна розглядати як основу для інтелектуалізації процесу формування сценарію АнД. Враховуючи тенденції щодо інтелектуалізації програмних засобів, така постановка задачі і досі є дуже актуальною, не тільки для інтелектуалізації процесу формування сценаріїв АнД, але й для вирішення широкого кола завдань, включаючи взаємодію аналітиків у межах систем організаційного управління.

Цілком зрозуміло, що сучасна АнД має ґрунтуватися на широкому застосуванні інформаційних технологій (ІТ). У контексті АнД технології можна розглядати як поєднання чотирьох типів інженерії – комп'ютерної, програмної, системної та інженерії даних і знань. Як відомо, інженерія (Engineering) – це творче застосування наукових принципів для розробки або розробки структур, машин, апаратів або виробничих процесів, або робіт [31].

Для АнС центральним об'єктом є база знань (БЗ), що формується відповідно до типових завдань ПрО, та враховує її особливості, що сприяє прийняттю обґрунтованих рішень. Сучасні методи інженерії знань (отримання знань від експерта, інтелектуальний аналіз даних, машинне навчання тощо) для побудови БЗ ґрунтуються на інженерії знань із застосуванням онтології – детальній формалізації знань ПрО за допомогою концептуальної схеми. Така схема, зазвичай, складається з ієрархічної структури даних, що містить усі релевантні класи об'єктів, їхніх зв'язків, теорем та обмежень, які прийняті в певній ПрО.

ІТ сценарного моделювання включають: визначення цілей побудови сценаріїв, множини допустимих станів сценарію, формування структури масивів вхідних і вихідних даних, методикку ітеративного процесу моделювання ПрО, онтології певної ПрО та оригінальне ПЗ, генерацію сценаріїв.

Аналіз публікацій у напрямку розробки та використання підходів до моделювання сценаріїв АнД свідчить про існування певних тенденцій, але

розв'язання проблеми в цілому не можна вважати завершеним. У межах зазначених напрямків слід виділити дослідження А.О. Морозова, О.Г. Додонова, Д.В. Ланде, Н.Г. Загоруйка, Е.А. Рабчевського, В.Ш. Рубашкина, Bart Kosko, Lotfi Aliasker Zadeh, T. Gruber, S. Staab, R. Studer, N.F. Noy, A. Gomez-Perez. Також методологія сценарного аналізу розглядається в роботах М.З. Згуровського, Т. Сааті, Ф. Цвікі, В.М. Одріна, С.С. Картавова, Т. Річі, Дж. Х. Лінстона та М.Турофа, Т. Гордона, О. Хелмера, Н. Долкі, Х. Сакмана та ін.

Які причини таких інтенсивних досліджень? Що дає використання онтології у складі АНС? Відомі дослідники в цій галузі зазначають – передовсім це формалізація знань за допомогою онтології, якої не вдавалось досягти за допомогою інших методів подання знань, інакше кажучи, онтології стали стандартом інженерії знань. По-друге, це використання спільного понятійного апарату як вченими, так і користувачами в усьому світі в певній ПрО. По-третє, це структуризація і організація знань.

Використання онтології при створенні сценаріїв АНД допомагає вирішити низку проблем методологічного та технологічного характеру, які виникають під час розроблення таких систем. Основні теоретичні засади формальних математичних моделей онтології розроблено у роботах Т. Грубера, Дж. Солтона, А. Гомес-Переса, які запропонували онтологію розглядати як тривимірний кортеж; у працях Н. Гуаріно, П. Фолтса, М. Шамбарда наведено методики побудови онтологій та їхні можливі шляхи розвитку; проблему автоматизованої побудови онтології висвітлено у роботах Д. Челюска, М. Варгас-Вера, Т. Ватсона, П. Льюїса, К. Блашке; Дж. Сова увів поняття концептуального графа, а М. Монтес-Гомес використав його для подання онтологій; використання онтології під час функціонування прикладних інформаційних систем описано в роботах Р. Кнаппе, К. Джонса, Е. Кауфмана, Е. Мена, М. Бориса, А. Каллі, І.П. Норенкова, М.Ю. Уварова, Ю.В. Рогушина; проблему побудови інтелектуальних систем на основі онтології розглянуто в роботах Т. Андреасена, Т. Бернерса-Лі, Д. Хендлера, О. Лазсіла, О.В. Палагіна, А.В. Анісімова, А.Я. Гладуна, зокрема опрацювання української природної

мови; використання онтології у структурі та під час функціонування інформаційних систем проаналізовано в роботах Джос де Брюїна, Д. Фенсела, С. Штаба, Р. Студера, Л. Холінка, М. Кокара, В. Лопеса, П.І. Андона, Л.А. Святогора, О.Г. Дубинського, І.Є. Кураленко, Т.А. Гаврилової, А.С. Клещева, І.Л. Артем'євої; проблему опрацювання природномовних текстів для автоматизованої побудови онтології досліджено в роботах С. Баррі, Р. Кахліна, Л.В. Найханової, Д.П. Ночевнова, І.В. Совпела, О.І. Боровікової, Ю.А. Загорулько. Аналізуючи роботи загалом, можна зробити висновок, що наукові дослідження в галузі розроблення та використання онтології під час побудови прикладних інформаційних систем активно розвиваються. Ці факти свідчать про актуальність проблематики побудови інтелектуальних систем підтримки прийняття рішень на основі онтології як предмету наукових досліджень.

Аналіз основних підходів, методів і засобів побудови інтелектуальних систем підтримки прийняття рішень і напрямів досліджень використання онтології показує, що у складі таких систем використовуються не всі можливості онтології, особливо під час моделювання функціональності сценаріїв, хоча переваги використання онтології порівняно з іншими методами побудови БЗ очевидні, оскільки саме онтології відображають об'єктивні знання та слугують стандартом інженерії знань. Зокрема, не вирішеними є такі завдання: моделювання процесів прийняття рішень і виведення нових знань на основі онтології; критерії наповнення онтології; оцінка новизни знань онтології тощо.

Підсумовуючи вищенаведене, основою верифікованих, повних, несуперечливих, цілісних інформаційних моделей сценаріїв АнД повинні бути об'єктивні знання про розвиток складного сценарію, подані не тільки у вигляді направлених оргграфів, але й у вигляді онтології. Це обумовлює розроблення та запровадження уніфікованих методів побудови інтелектуальних систем підтримки прийняття рішень з використанням онтологічного підходу з метою підвищення ефективності процесів (сценаріїв) функціонування аналітичних систем. У дисертаційній роботі подано вирішення цієї проблеми у вигляді

теоретично обґрунтованих моделей функціонування та методів побудови інтелектуальних систем формування сценаріїв Анд, поєднуючи процесне уявлення сценарію у вигляді оргграфів з онтологічною моделлю із залученням широкого спектра методів семантичного аналізу. Суть полягає в адаптації баз знань до специфіки задач формування сценаріїв у відповідної Про, а також оптимізації онтології і проведення семантичного аналізу коректності логічно-з'язаних фрагментів сценарію (особливо багаторівневих), використовуючи технології та інструментарій редакторів онтології.

Таким чином, можна стверджувати, що є актуальним і необхідним вирішення науково-прикладної проблеми дослідження, формалізації та розробки теоретичних основ та практичних засад комп'ютерного моделювання сценаріїв аналітичної діяльності в різних предметних областях, зокрема при проектуванні складних технічних систем.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконана в межах концепції програми інформатизації Національної академії наук України, визначеної пріоритетним напрямом згідно розпорядження № 146 від 27.02.2004 р., та за тематикою наукових досліджень Інституту проблем реєстрації інформації Національної академії наук України, зокрема в рамках науково-дослідних робіт: «Дослідження методів побудови аналітичної складової корпоративних інформаційно-аналітичних систем» (шифр «КІАС-2008»), № держреєстрації 0108U000260; «Дослідження та розробка технологій комп'ютерного моделювання інформаційно-аналітичних систем» (шифр «АСКІАС-2011»), № держреєстрації 0111U002089; «Дослідити та розробити технології комп'ютерного моделювання систем керування знаннями при проведенні аналітичної діяльності» (шифр «УЗОР-2014»), № держреєстрації 0114U002060; «Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності в системах організаційного управління» (шифр теми «СЦЕНАРІЙ») № держреєстрації 0119U001165.

Окрім того, у роботі використано результати, які отримав автор під час виконання науково-дослідних робіт у Національному технічному університеті

України «Київський політехнічний інститут імені Ігоря Сікорського», а саме: «Розробка алгоритмів (документації) та програмного забезпечення виявлення, визначення напрямку та класифікації морських об'єктів для існуючої у Інозамовника вимірювальної системи з дослідження сигналів та перешкод в світовому океані», № держреєстрації 0116U005555; «Дослідження та розробка математичних моделей сигналів та перешкод, розрахунок параметрів розповсюдження звукових хвиль у морському багат шаровому середовищі», № держреєстрації 0116U005598; «Моніторинг фінансових показників виконання бюджетів всіх рівнів», № держреєстрації 0111U004274.

Мета і задачі дослідження. Метою дисертаційної роботи є створення методів і засобів побудови комп'ютерних систем моделювання сценаріїв аналітичної діяльності.

Для досягнення поставленої мети в дисертаційній роботі вирішено такі задачі.

1. Аналіз сучасного стану і особливостей проблеми дослідження, створення та вдосконалення комп'ютерних систем моделювання багаторівневих сценаріїв АнД, вибір та обґрунтування перспективних підходів до її вирішення з метою визначення та зменшення логічно зв'язаних помилок та семантичної незв'язаності опису багаторівневих сценаріїв чи їх окремих гілок, що призводить до суттєвого зменшення часу моделювання сценаріїв в цілому.

2. Розробка теоретичних засад встановлення співвідношення графічних елементів опису сценарію АнД з описом онтології його предметної області з метою забезпечення можливості проводити семантичний аналіз коректності як логічно-зв'язаних фрагментів сценарію АнД, так і його верифікацію.

3. Дослідження можливостей та розвиток математичних моделей і методів побудови сценарію АнД з урахуванням можливих наступних кроків сценарію та на основі виведення по прецедентах, що вже накопичені в результаті попереднього моделювання сценаріїв АнД.

4. Розробка та дослідження метода виконання сценаріїв АнД на основі Web-сервісів, вибір яких засновано на механізмі порівняння параметрів метаописів Web-сервісів із параметрами функціональних завдань сценарію.

5. Розробка алгоритмів та програмних засобів комп'ютерного моделювання сценаріїв на основі розроблених та удосконалених моделей (сценарних моделей, моделей опису предметної області в умовах невизначеності, заданих обмежень) та методів побудови та виконання сценаріїв АнД.

6. Створення комп'ютерної моделюючої системи (інтегрованого програмного інструментального середовища) для тестування і оцінки запропонованого теоретичного підходу комп'ютерного моделювання сценаріїв АнД, яка ґрунтується на побудованих моделях, методах та алгоритмах моделювання.

Об'єктом дослідження є процеси функціонування комп'ютерних систем моделювання сценаріїв АнД.

Предметом дослідження є методи і засоби математичного та комп'ютерного моделювання процесів функціонування комп'ютерних систем моделювання сценаріїв АнД.

Методи дослідження. Для розв'язування поставлених задач у дисертації використовуються теоретико-множинні підходи; алгебра систем; теорія графів; методи семантичного моделювання, методи математичного моделювання переходів станів; апарат модельно-орієнтованих підходів до побудови комп'ютерних ПС, методи концептуального моделювання, методи організації засобів комп'ютерного моделювання динамічних процесів і систем, загальні архітектурні принципи побудови ПС, методи побудови сервісно-орієнтованих ПС, методи оцінювання тривалості виконання завдань проекту.

Наукова новизна одержаних результатів. В рамках розв'язання проблеми створення методів і засобів математичного та комп'ютерного моделювання багаторівневих сценаріїв АнД на основі поєднання переваг візуального моделювання і семантичного аналізу та подальшого автоматичного

перетворення метаданих моделі сценарію в виконавче програмне інструментальне середовище отримані наступні наукові результати.

Вперше запропоновано:

– теоретичний підхід до комп'ютерного моделювання багаторівневих сценаріїв АнД, якій відрізняється від відомих тим, що оснований на поєднанні переваг візуального моделювання – графічної нотації BPMN (Business Process Model and Notation) з керованою моделлю і семантичного аналізу на базі онтологічної моделі предметної області задач АнД, яка конвертується з BPMN-моделі, та подальшого автоматичного перетворення метаданих BPMN-моделі в виконавче програмне середовище (серіалізації), що зменшує кількість логічних розривів в комп'ютерній моделі побудови сценарію АнД, яка будується, та суттєво зменшує час моделювання в цілому;

– метод перетворення BPMN-моделі в модель уявлення про предметну область сценарію АнД мовою OWL (Web Ontology Language) - OWL-модель, якій відрізняється від відомих тим, що за рахунок забезпечення співвідношення категорії основних графічних примітивів нотації BPMN до відповідних класів онтології (сутностей) та зв'язування їх з властивостями та відношеннями між елементами реальної моделі вирішення задачі АнД в нотації BPMN дозволяє проводити семантичний аналіз коректності логічно-зв'язаних фрагментів сценарію вирішення задачі АнД та його верифікацію (особливо багаторівневих сценаріїв);

– інформаційна технологія конвертації BPMN-моделі в OWL-модель сценарію АнД, яка відрізняється від відомих тим, що дозволяє перетворювати метадані опису сценарію та його предметної області в базі знань, накопичуючи нові знання, що дає можливість побудови нових сценаріїв АнД на базі вже існуючих в базі знань. Для спрощення процесу комп'ютерного моделювання сценаріїв залучається графічна нотація BPMN 2, яка дозволяє не тільки моделювати сценарії АнД у вигляді BPMN-діаграм, але й серіалізувати його опис в XML-файл для реалізації сценарію в комп'ютерному середовищі із залученням стандартного програмного забезпечення BPM-систем.

Удосконалено:

– математична модель обчислення розподілу значень можливих подій сценарію та інтелектуальний метод визначення наступних подій сценарію АнД, який відрізняється від відомих тим, що на основі застосування методів машинного навчання, а саме розвитку методу навчання деревами класифікації і регресії (Classification and Regression Trees) з використанням комбінації метрик оцінок ефективності та накопичених знань про семантику предметної області для визначення кореневого вузлу графу, дозволяє виконувати формування дерева рішень для класифікації можливих наступних кроків сценарію. Це дає можливість значно підвищити точність розбиття графу складного сценарію на відповідні гілки при обчисленні ймовірності наступного кроку сценарію;

– метод реалізації сценаріїв АнД на основі Web-сервісів, який відрізняється від відомих тим, що вибір сервісів засновано на механізмі порівняння параметрів їх метаописів із параметрами функціональних завдань сценарію, які задаються його метаописом, що забезпечує можливість виконання завдань кінцевого користувача шляхом динамічного формування послідовності Web-сервісів.

Набули подальшого розвитку:

– процес автоматизованої побудови сценаріїв АнД на основі бібліотеки сценаріїв (прецедентів) для різних предметних областей. Це дало можливість в автоматизованому режимі будувати сценарії на основі експертного опису проблемних ситуацій, здійснювати деталізацію сценаріїв та розгалуження їх. У свою чергу, це суттєво зменшує часові та фінансові витрати на моделювання та реалізацію сценаріїв АнД.

– інтелектуальний програмний засіб, що навчається на діях користувача та на основі отриманих в процесі навчання знань представляє дії користувача у вигляді дерева рішень ймовірного сценарію аналітичного процесу, пропонуючи користувачу найбільш доцільні наступні кроки вирішення завдання АнД;

– трирівнева комп'ютерна моделююча система (інтегроване програмне інструментальне середовище) для тестування і оцінки запропонованого

теоретичного підходу (методології) моделювання сценаріїв АнД, яка ґрунтується на запропонованих моделях, методах та алгоритмах моделювання, що дало можливість реалізувати окремі програмні інтелектуальні засоби вирішення задач аналітики для різних предметних областей аналітичної діяльності, ядром баз знань яких є знання та накопичений досвід аналітика (експерта, проектувальника). Програмні інтелектуальні засоби були використані зокрема для вирішення задачі побудови та реалізації сценаріїв моделювання процесів функціонування вимірювальної гідроакустичної системи з метою удосконалення алгоритмів обробки гідроакустичних сигналів, та сценаріїв розподілу енергопостачання з метою оптимізації енерговитрат.

Практичне значення одержаних результатів. Запропоновані та створені засоби алгоритмічного та програмного забезпечення процесів моделювання сценаріїв АнД забезпечують комп'ютерну реалізацію та взаємодію отриманих видів моделей графічного представлення сценаріїв АнД, семантичних моделей предметної області задач АнД та знань про процес побудови та виконання сценаріїв АнД у вигляді адаптивної об'єктної моделі. За використанням розроблених інтелектуальних програмних засобів здійснюється автоматизована побудова сценарію АнД з урахуванням попередніх його кроків та на основі виведення по прецедентах, що вже накопичені в результаті попереднього моделювання сценаріїв АнД. Такій підхід забезпечує зменшення логічно-зв'язаних помилок та семантичної незв'язаності опису багаторівневих сценаріїв чи їх окремих гілок та тем самим призводить до суттєвого зменшення часу моделювання сценаріїв АнД. Це підтверджують результати проведених імітаційних експериментів на тестових прикладах, які показали, що в залежності від кількості рівнів (кількістю рівнів 4 та 10) графу багаторівневого сценарію та кількості вузлів в рівні (кількість вузлів на рівні 3,5,7,10 та 15) час моделювання мінімально зменшився на 10 % та максимально зменшився на 25 %.

Розроблені комп'ютерні програмні засоби призначені для дослідження та забезпечення процесів моделювання сценаріїв АнД як у лабораторних

дослідженнях, так і в умовах застосування в реальних системах із забезпеченням їх інтелектуалізації та адаптивності до змін предметної області завдань, які вирішує аналітична ПС.

Створені методи та засоби математичного та комп'ютерного моделювання сценаріїв АНД використано при розв'язуванні ряду практичних задач, зокрема, для вирішення задачі побудови та реалізації сценаріїв моделювання процесів функціонування вимірювальної гідроакустичної системи з метою удосконалення алгоритмів обробки гідроакустичних сигналів, та сценаріїв розподілу енергопостачання з метою оптимізації енерговитрат.

В роботі вирішено наступні прикладні задачі, що представлені у додатках: «Побудова та реалізація сценаріїв моделювання процесів функціонування вимірювальної гідроакустичної системи» (додаток А); «Комп'ютерне моделювання сценаріїв розподілу енергопостачання» (додаток Б); «Формування сценарію інтерактивної обробки багатовимірних даних» (додаток В); «Побудова та реалізація сценаріїв аналітичної діяльності в задачах просторового моделювання» (додаток Г); «Моніторинг фінансових показників виконання бюджетів всіх рівнів» (додаток Д). Основні результати роботи та основні положення, які виносяться на захист, пройшли апробацію при розв'язанні модельних і практичних задач.

Результати дисертаційного дослідження прийнято до впровадження в Дочірньому підприємстві «Національний фонд розвитку», ТОВ «Науково-виробниче підприємство «Символ», ТОВ «Інженерна логіка», Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського».

Особистий внесок здобувача. Усі наукові результати дисертаційної роботи отримані автором особисто. У друкованих працях, опублікованих у співавторстві, автору належать: [1] – формулювання та доведення методології семантичного моделювання побудови систем управління знаннями та алгоритмічних і технологічних засад моделювання аналітичної діяльності; [2] – постановка задачі, класифікація моделей, архітектура та принципи

функціювання програмних засобів побудови та реалізації сценаріїв аналітичної діяльності в інформаційно-аналітичних системах; [3] – ідея інтегрованого підходу до визначення якісної оцінки сценарію аналітичної діяльності, якій заснований на виявленні найкоротшого шляху у графі онтології; [4] – постановка задачі та застосування підходу до моделювання змінюваних сценаріїв аналітичної діяльності є залученням технологій BPM-систем з керованою моделлю; [5] – адаптивна мікросервісна архітектура інформаційно-аналітичної системи; [6] – постановка задачі та принципи формування багаторівневих сценаріїв GIS-аналітики за застосуванням BPMN-моделі в поєднанні з онтологічною моделлю ПрО; [7] – постановка та способи вирішення задачі побудови типових сценаріїв на основі аналізу онтології предметної області з використанням представлення її структури у вигляді графу; [8] – інтелектуальний метод побудови сценарію аналітичної діяльності, заснованого на розвитку методу навчання деревами класифікації і регресії з використанням комбінації метрик оцінок ефективності; [9] – застосування підходу до побудови сценаріїв проектування бізнес-процесів виконання інженерних завдань на порталах знань; [10] – застосування підходу використання оцінок ймовірностей для аналізу відповідності джерел інформації запиту пошуку відповідних документів у цих джерелах; [11] – принципи застосування концептуальної моделі сценарно-цільового підходу до аналізу об'єктів дієвої аналітики; [12] – постановка задачі та застосування підходу використання типових сценаріїв аналізу виконання проектів на основі типових задач за створенням репозитарію сценаріїв і репозитарію типових задач; [13] – принципи побудови комп'ютерної моделі системи інформаційного управління з дієвою аналітикою; [14] – принципи застосування адаптивної об'єктної сценарної моделі для побудови сценарію аналітичної діяльності; [15] – постановка задачі та застосування підходу до побудови системи моніторингу виконання держбюджету України, який ґрунтується на інформаційній моделі інформаційно-аналітичної системи, моделі прецедентів предметної області та типових сценаріях моніторингу й аналізу державного бюджету; [16] – постановка задачі та застосування підходу

до підтримки управління проектами, заснований на побудові та застосуванні сценаріїв аналізу змін бізнес-процесів та бази знань проекту для визначення оперативних завдань управління проектом; [17] – постановка задачі та застосування підходу до аналізу типових задач проектів, заснований на накопиченні знань про виконання всіх проектів з часовими параметрами, та схема послідовності обробки даних оцінки своєчасності виконання проектів; [18] – постановка задачі та принципи створення спеціалізованих програмних засобів аналізу проектної діяльності з урахуванням стандартних корпоративних політик безпеки; [20] – принципи застосування розподіленої сервіс-орієнтованої архітектури автоматизованої системи екстреної допомоги населенню за єдиним телефонним номером 112; [21] – принципи побудови комп'ютерної адаптивної моделі корпоративних інформаційно-аналітичних систем, яка базується на концептуальних засадах теорії складних адаптивних систем; [22] – принципи побудови та застосування мікросервісної архітектури інформаційно-аналітичної системи для оцінки рівня міжнародної діяльності, в якій кожен мікросервіс створюється як автономний контейнер; [23] – застосування методу спрощення інженерних завдань за допомогою специфічної алгебраїчної системи інженерних завдань, спрямованих на формування мінімізованого набору веб-сервісів порталу знань; [24] – формування етапів та параметрів розвитку онтологічної моделі предметної області при проектуванні програмної платформи аналізу інформації; [25] – методологія створення онтологічної моделі предметної області інтелектуальної системи дослідження гідроакустичних процесів та принципи побудови її узагальненої архітектури на основі знань, які накопичуються в процесі дослідження; [26] – метод формування сценарію завдань користувача, що дозволяє динамічне визначення впорядкованої послідовності веб-сервісів, необхідних для виконувати завдання користувача.

Апробація результатів дисертації. Основні результати дисертаційних досліджень неодноразово доповідалися на міжнародних наукових конференціях, зокрема на конференціях «3-я Міжнародна конференція з

передових інформаційних та комунікаційних технологій» (м. Львів, 2019), «VII, XVIII та XIX Міжнародна науково -практична конференція "Інформаційні технології та безпека" (м. Київ, 2017-2019), «IEEE Міжнародна Чорноморська конференція з питань комунікацій та мереж» (м. Батумі, Грузія, 2018), «Міжнародна конференція «Радіоелектроніка та інформаційні комунікації» (м. Київ, 2016, м. Одеса, 2017), «14-18 Міжнародна науково-технічна конференція «Системний аналіз та інформаційні технології» (м. Київ, 2012–2016), «VII та IX Міжнародна науково-практична конференція аспірантів, магістрів, студентів «Сучасні проблеми наукового забезпечення енергетики» (м. Київ, 2009, 2011), Науково-практична конференція с міжнародною участю «Системи підтримки прийняття рішень. Теорія і практика» (м. Київ, 2007), «Науково-практична конференція с міжнародною участю «Проблеми моніторингу та забезпечення економічної безпеки держави» (м. Київ, 2007).

Публікації. За результатами досліджень опубліковано 44 наукових роботи, з них: 2 монографії, 24 статті у наукових фахових виданнях України, що входять до переліку, затвердженого МОН України, 22 статі у виданнях, які включені до міжнародних наукометричних баз, 18 публікацій у працях і матеріалах наукових конференцій, 5 публікацій в закордонних виданнях, 8 публікацій у збірниках наукових праць, які включено до міжнародної наукометричної бази Scopus.

Структура та обсяг роботи. Дисертаційна робота складається з анотації, вступу, шести розділів, висновків, списку використаних джерел з 290 найменувань на 18 сторінках та 4 додатків на 23 сторінках. Загальний обсяг дисертацій 334 сторінки, з них 283 сторінки основного тексту, 75 рисунків, 17 таблиць.

РОЗДІЛ 1. ПРОБЛЕМИ ПОБУДОВИ КОМП'ЮТЕРНИХ СИСТЕМ МОДЕЛЮВАННЯ СЦЕНАРІЇВ АНАЛІТИЧНОЇ ДІЯЛЬНОСТІ

1.1 Особливості аналітичної діяльності для різних предметних областей

Сьогодні інформація та знання вважається унікальним нематеріальним активом, що визначає конкурентоспроможність організацій, глибину та обґрунтованість управлінських рішень і багато інших аспектів їхньої діяльності. Наявність або відсутність необхідної інформації і знань буває вирішальним фактором у ході прийняття збалансованих рішень. Але придбання або видобуток інформації є тільки першим кроком, за яким необхідно здійснити її аналіз і витяг необхідних даних з метою отримання знань. Саме тому сьогодні аналіз та Анд, яка з ним асоціюється, перетворюються на креативний вид управлінської діяльності, без якої неможливо представити існування сучасної конкурентоспроможної організації.

Функції аналізу інформації і знань поширюються на всі сфери людства, починаючи від економіки та маркетингу і закінчуючи сферою прийняття політичних і військових рішень. Проблема загострюється через значну невпорядкованість в аналітичній термінології. Тому не визиває здивування, що Анд, залежно від сфери діяльності людства, має різні тлумачення. Так, неоднозначно трактуються такі терміни, як «аналіз», «аналітика», «аналітична діяльність», «види аналізу» та інші. Загальновідомий «інформаційний підхід» трактує інформаційно-аналітичну діяльність як таку, що застосовує науково-інформаційні методи та використовує технології обробки інформації [172].

Якщо аналітику розглядати з точки зору аналітичної роботи, яка багатьма науковцями трактується як творча діяльність [169], то вона сприймається як сукупність процесів семантичної обробки даних із використанням методів і засобів «інтелектуальної аналітики» [239]. В результаті інтелектуального аналізу розрізнені дані перетворюються на закінчену продукцію – аналітичний документ.

Деякі науковці трактують Анд як сукупність дій на основі концепцій, методів, засобів, нормативно-методичних матеріалів для збору, накопичення, обробки і аналізу даних з метою обґрунтування та прийняття рішень, або синтезу нових знань [88, 100, 162, 220].

У сучасному бізнесі – Анд розглядається як діяльність, спрямована на аналітичну обробку даних, що об'єднує, аналізує та зберігає інформацію, видобуту як з БД, що містяться в інформаційних ресурсах організації, так і із зовнішніх джерел [30, 36, 173]. Однак, слід зазначити, що чітке розділення між різними аспектами бачення Анд є умовним і можливим, виключно, на рівні абстрагування в теоретичній площині.

Аналітика включає в себе підбір і систематизацію фактів стосовно процесів або явищ, які є предметом аналізу, їхню оцінку, відбір, тлумачення та підготовку звітних документів з відображенням результатів аналізу в зручному для сприйняття вигляді. Інтуїтивно зрозуміло, що головна мета будь-якої Анд полягає в отриманні максимальної користі від інформації і знань, які є в розпорядженні аналітика, для того щоб правильно зрозуміти і оцінити ситуацію, прогнозувати її розвиток та наслідки, а в кінцевому підсумку – успішно діяти.

Основні принципи організації аналітичного процесу сформульовані ще кілька десятиліть тому американським генералом Вашингтоном Плетом (фахівцем зі стратегічної розвідки), але вони й досі не втратили актуальність для сучасних аналітиків [228, 233, 246, 191]. Вони є такими:

Цілеспрямованість Анд – тобто визначення конкретних цілей і завдань, які мають бути досліджені у визначеній сфері.

Достовірність даних – постійне оновлення та актуалізація даних, потрібних для аналізу та дослідження проблем у визначеній сфері.

Комплексний аналіз проблем, що вирішуються – здійснюється з урахуванням їхнього місця, ролі та взаємозв'язків, виявлення суті та причини проблеми або явища.

Актуальність досліджень – повинна витікати з потреб практики та мати високу міру важливості для вирішення конкретної проблеми. Своєчасність отримання та видачі результатів у необхідні терміни, в зручному вигляді та формі для її сприйняття та осмислення керівництвом.

Прогнозування наслідків і негативних факторів, що можуть ускладнити ситуацію в досліджуваній сфері. Для забезпечення ефективності дослідження потрібне визначення динаміки розвитку ситуації, що вивчається, і передбачення можливих негативних наслідків, встановлення та пояснення закономірностей зміни показників, що характеризують ситуацію, розробка сценаріїв її розвитку та експертне прогнозування.

Формулювання пропозицій щодо способів їхнього рішення, тобто вироблення не лише оціночних результатів, але і конструктивних пропозицій, і рекомендацій щодо вирішення проблем.

Моніторинг обстановки – організація постійно діючого інформаційно-аналітичного моніторингу обстановки, своєчасно та із заданою мірою деталізації того, що відбиває основні зміни в досліджуваній ситуації.

Неупереджене відношення аналітика до дослідження і його результатів. Об'єктивне та всебічне обґрунтування висновків, оцінок, пропозицій на підставі аналізу достовірності даних. При цьому передбачається висунення та перевірка усіх можливих варіантів розвитку подій, версій про суть і причини явища, що вивчається, визначення закономірностей його розвитку.

Забезпечення умов альтернативності досліджень – наявність механізмів вільного (незалежного) дослідження та доведення до вищого за рангом керівництва альтернативних висновків за результатами проведеного дослідження.

У загальному сенсі, АнД є складовою пізнавальної діяльності, націленої на пошук і розкриття тих закономірностей (нових знань), які на початок досліджень невідомі. Іноді це може надати аналітичному дослідженню непередбачуваний результат. Але, якщо АнД (у методичному сенсі) організована коректно, то навіть помилкове рішення, якщо воно отримало

правильну оцінку, здатне надати нові знання, корисні для оцінки та вирішення проблеми в цілому. Призначення аналізу полягає в тому, щоб відокремити суттєве від несуттєвого, звести складне до більш простого.

Найбільш розвинутою формою АнД виступають аналітичні організації, центри, які за визначенням американського дослідника П. Діксона «...не створюють майже нічого речовинного, крім паперів. Головний їхній продукт – це теоретичні вишукування, звичайно вбрані у форму звітів або досліджень, що являють собою варіанти різних заходів, оцінки, проекти, теорії, рекомендації, попередження, перспективні плани, статистичні зведення, прогнози, описи методів, тести, аналізи або просто нові ідеї» [242].

Особливе значення АнД відіграє у сфері державного управління. АнД реалізується за двома класами суб'єктів: за рівнем організації і залежно від належності до сфери реалізації аналітики та використання результатів. Перший клас охоплює значний спектр суб'єктів – від спеціальних аналітичних інституцій до менеджерів організації. До другого класу входять аналітики залежно від основних напрямів її прояву, таких як державне управління, місцеве самоврядування, бізнес, громадянське суспільство.

Сфера державного управління характеризується значними потоками інформації, що акумулюються в різноманітних ІР, та мають перероблятися для прийняття критично важливих рішень для життєдіяльності суспільства. Специфіка АнД в органах державної влади полягає в тому, що вона «вбудована» в процес управління, до неї висуваються особливі вимоги щодо оперативності підготовки матеріалів, їхньої достовірності та обґрунтованості. Важливого значення набуває експертна діяльність, залучення фахівців з усіх напрямів.

Органи державного управління, аналізуючи дані, можуть вивчати громадську думку та впливати на її зміну, більш ефективно планувати свої програми та роботу з населенням, оцінювати вплив програми на цільовий сегмент, знаходити шляхи для задоволення потреб населення.

Розглядаючи АнД у сфері систем організаційного управління, слід нагадати поняття систем організаційного управління – *managerial control system*. У цьому

питанні серед науковців (як зарубіжних, так і вітчизняних) є суттєві розбіжності щодо цілісного погляду на сферу організаційного управління. Так, за Л.І. Лопатниковим «система організаційного управління (СОУ) – це система управління, об’єктом якої, на відміну від системи управління технологічними процесами, є не машини або інші технічні пристрої, а перш за все людина, людські колективи» [238]. Аналогічне визначення наведене і в роботі [5], яке визначає СОУ як «систему процесів, що надають організаційний вплив на групу людей і на організаційну систему в цілому».

Такі зарубіжні фахівці як Хорнгрин, Датар і Фостер визначають систему управління «як засіб збору та використання інформації для сприяння та координації процесу прийняття рішень щодо планування та контролю через організацію та для управління поведінкою її керівників і працівників. Метою системи організаційного управління є вдосконалення колективних рішень у рамках організації економічно доцільним способом» [236]. За їхнім визначенням СОУ може включати як формальну, так і неформальну систему управління. Формальна система управління вимагає, щоб організація мала чіткі правила, процедури, вказівки, плани, які стосуються різних аспектів управління структурою організації. Такі компоненти необхідні для керівництва компанії, спрямованості, мотивації керівників та інших працівників і координації їхніх зусиль для досягнення організаційних цілей.

Основними етапами розробки сценаріїв АнД є такі.

1. **Визначення цілей, мети та проблеми аналізу** починається з формулювання глобальної мети дослідження. Надалі вона конкретизується за рахунок визначення цілей та проблем аналізу. Для багатьох складних Про процес визначення цілей перетворюється в побудову дерева цілей і завдань (проблем), яке може відображатися у вигляді графа, що дозволяє на ранньому етапі аналізу виявити внутрішню суперечливість глобальної мети, що дуже важливо для виявлення формальних передумов досяжності мети в цілому. Для

визначення мети задається функція мети – F_{gl} , яка дозволяє визначити, чи може бути досягнута мета gl , якщо F_{gl} , знаходиться у певному стані t'_{BFC} :

$$F_{gl}(t'_{BFC}) = \begin{cases} true & | t'_{BFC} \in t_{GL} \\ false & | t'_{BFC} \notin t_{GL} \end{cases}.$$

Наявність максимально повної інформації про можливі варіанти досягнення мети дозволяє приймати рішення не тільки на основі експертних оцінок, але й формальними методами моделювання за рахунок порівняння і оцінки найбільш імовірних сценаріїв розвитку процесів і подій, їхньої стійкості на основі інформації про структурні особливості.

2. Аналіз наявних знань, інформації і прецедентів. На момент виникнення проблеми аналізу, зазвичай, у системі вже існують деякі знання, накопичений досвід, прецеденти та моделі інтерпретації проблеми, що вирішується. Питання полягає в тому, наскільки наявні знання можуть бути доцільні для вирішення даного конкретного завдання, чи можуть вони бути застосовані без модифікації, і які напрямки модифікації моделей і поповнення інформаційних ресурсів слід застосувати. Ресурсами, що залучаються для вирішення завдання, є різноманітні джерела інформації, які існують у системі архіви прецедентів, моделі їхнього вирішення, бази даних і знання експертів.

3. Збір даних, інформації, знань, пов'язаних з проблеми аналізу. Дані, потрібні для аналізу проблеми, можуть: мати різноманітні форми подання; бути як забезпечені, так і не забезпечені моделями їхньої інтерпретації; бути розподілені в IP, різних за своєю фізичною природою, часової і просторової локалізації джерел; джерела даних можуть бути в різному ступені доступності; інформація може дублюватися в різних джерелах; ступень достовірності інформації може значно відрізнятися. Збір даних, інформації, знань, як правило, багато ітераційний процес, тому оцінка достатності отриманого матеріалу або експрес-аналіз даних є невід'ємною частиною цього процесу. Завдання експрес-аналізу полягає у визначенні важливості, вірогідності та значимості даних, а також перевірки інформації на помилковість. Крім того, в результаті аналізу над

наявними масивом даних здійснюються певні маніпуляції, у результаті яких з'являється спеціалізований масив каталогізованих і, можливо, анотованих даних, необхідних для проведення подальшого дослідження проблеми.

4. **Інтерпретація зібраних факторів** – складається з таких кроків:

- фільтрація – відбір інформації і наявних знань, які вже існують в системі, та їхнє сортування за встановленими критеріями;
- класифікація завдань з метою порівняння з прецедентами, які мали місце в минулому, та визначення класу, до якого можна віднести нове завдання;
- аналіз досвіду вирішення прецедентів і знаходження відповідних даних, а також моделей вирішення проблеми, асоційованих з певним класом завдань;
- відбір з отриманого масиву даних і моделей тих, що релевантні новому завданню;
- встановлення відмінностей даного конкретного завдання від тих, що раніше розв'язувалися;
- визначення нового класу завдання, якщо подібні завдання ще не вирішувалися;
- встановлення тих блоків даних і компонентів моделей, які не можуть бути застосовані для вирішення даного завдання.

5. **Розробка робочої гіпотези для дослідження проблеми.** На основі інтерпретації зібраних факторів формується робоча гіпотеза для дослідження проблеми. Робоча гіпотеза – це *припущення про зв'язок між наявними факторами та причинами, які викликали наявні фактори або події*. Причому, висунете припущення обов'язково потребує або доведення, або спростування. У методологічному плані це означає, що не підтверджена гіпотеза ні є помилкою, а свідчить лише про те, що висунете припущення не має підтвердження та потребує корегування **робочої гіпотези**. Для успішного аналізу проблеми робоча гіпотеза має відповідати наступним вимогам:

- гіпотеза має бути конкретною – відповідати цілям і меті дослідження;

– достовірною, тобто гіпотеза повинна створюватися безпосередньо на підставі проблеми та мати тлумачення можливих наслідків, а також умов та обставин, за яких ці наслідки обов'язково настають;

– можливості перевірки припущень на фактичному матеріалі або на порівнянні з прецедентами (наявними знаннями о фактах і подіях у минулому);

– здатністю до формалізації, тобто можливості формального опису гіпотези за допомогою математичних або семантичних методів і логічних висновків.

6. **Визначення критеріїв оцінки** коректності робочої гіпотези. Залежно від складності робочої гіпотези та особливостей ПрО може використовуватись або як однокритеріальний, або як багатокритеріальний метод оцінки її коректності. За критерій, як правило, обирається деяке значення або діапазон значень параметрів, що є наважливими для робочій гіпотезі. Також визначаються критерії оцінки досяжності цілей і мети, пов'язані з проблемою аналізу.

7. **Аналіз, перевірка гіпотези і оцінка її коректності.** На цьому етапі складається план застосування різних методів та інструментальних засобів для дослідження гіпотези і оцінки її коректності. Для **цього** використовується широкий спектр методів, включаючи методи експертних оцінок, інтелектуальної обробки інформації, аналізу даних, а також методи витягу знань, отриманих із застосуванням продукційних правил на основі онтології ПрО. Різні методи аналізу володіють різною «чутливістю» щодо різних аспектів функціонування об'єктів дослідження.

8. **Процес синтезу нових знань.** Перевірена, обґрунтована, підтверджена дослідженнями, фактами, аргументами, гіпотеза перетворюється в достовірне знання, яке використовується для формування сценарію розвитку подій, на основі отриманих даних та скритих закономірностей.

9. **Оформлення аналітичних матеріалів.** Це заключна частина виконання типового сценарію, що містить висновки з проведеного дослідження і рекомендації щодо реалізації його результатів. У разі необхідності для

досягнення максимального ефекту можна використовувати таблиці та діаграми, а також мультимедійні презентації.

На рисунку 1.1. показано загальну структуру аналітичної діяльності з формуванням сценаріїв [196].

Для отримання знань з інформації використовується широкий спектр методів, включаючи методи експертних оцінок, інтелектуальної обробки інформації, аналізу даних, а також методи витягу знань, отриманих із застосуванням продукційних правил на основі онтології предметної області (таблиця 1.1).

Аналітична діяльність у складі АНС полягає в отриманні нової інформації і знань про об'єкт дослідження відповідно до цілей аналізу на основі побудови адекватної інформаційної моделі об'єкта та вивчення стану методами комп'ютерного моделювання можливих варіантів зміни його стану залежно від керуючих впливів і зміни зовнішнього середовища.

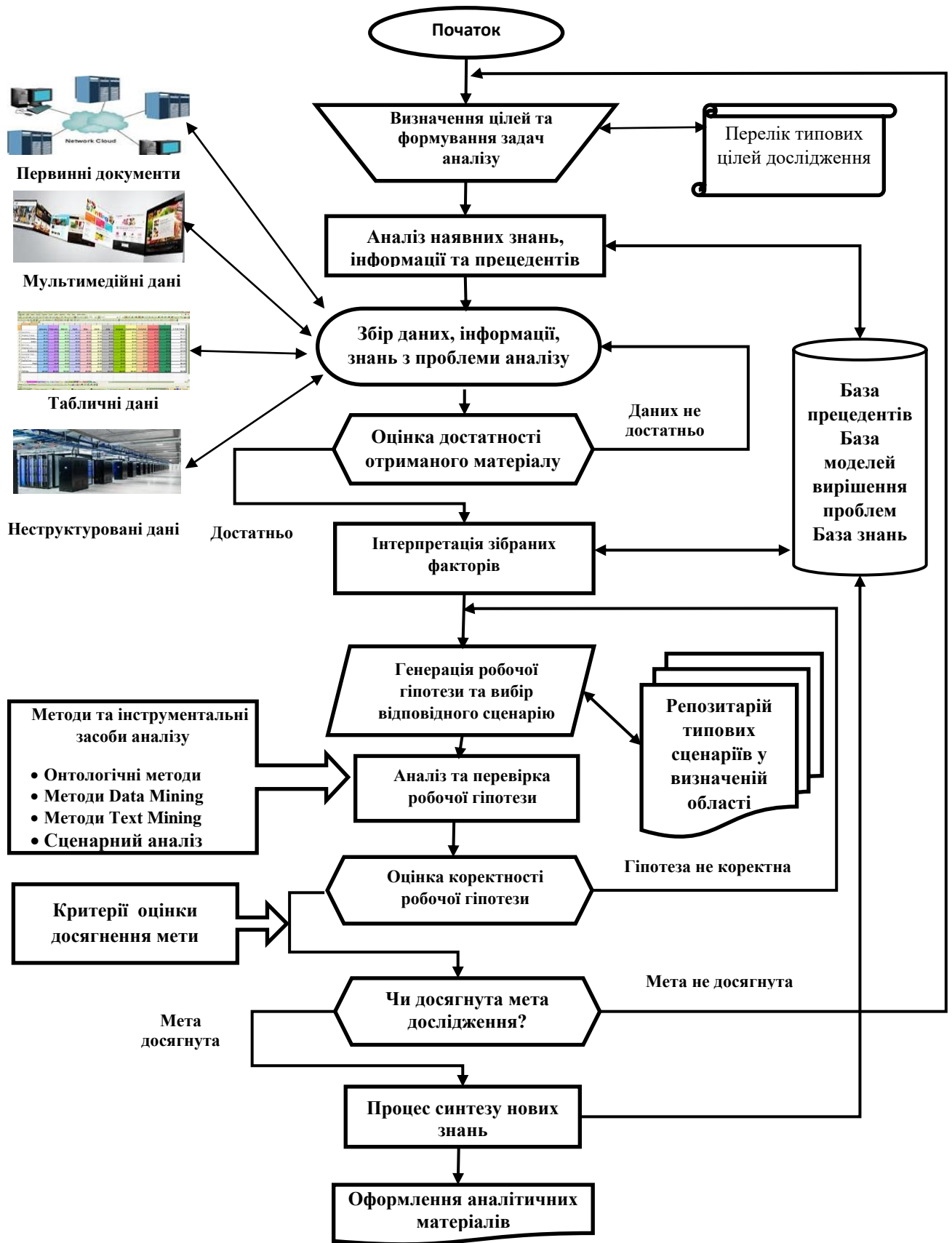


Рисунок 1.1 – Аналітична діяльність

	Методи аналізу	Характерні риси методу аналізу
1.	Графічний аналіз	Полягає в пошуку на графіку певних закономірностей
2.	Метод Дельфі	Delphi technique – один з методів прогнозування, який ґрунтується на припущенні, що на основі думок експертів можна збудувати адекватну модель майбутнього розвитку об'єкта прогнозування
3.	Методи експертних оцінок	Це спосіб прогнозування та оцінки майбутніх результатів дій на основі прогнозів фахівців. Дозволяє отримати об'єктивну оцінку на основі певної сукупності індивідуальних думок експертів
4.	Факторний аналіз	Застосовується для вивчення взаємозв'язків між значеннями показників, що впливають на досліджувані процеси
5.	Аналіз временних рядів	Рішення завдання прогнозу поведінки часового ряду у вигляді поліноміальної апроксимації трендів відхилень
6.	Варіаційний аналіз	Аналіз відхилень – (variance analysis) наприклад, між кошторисними показниками витрат, доходу або прибутку і фактичними показниками
7.	Порівняльний аналіз	Встановлює закономірності, вдаючись до порівняння стану дослідження об'єктів у різні часи, порівняння їх якостей у минулому з такими самими сучасними якостями для встановлення змін або тенденцій розвитку
8.	Метод головних компонент	Виявлення скритих, об'єктивно існуючих закономірностей у зміні явищ. 1. Аналіз причинно-наслідкових взаємозв'язків показників і встановлення їхнього стохастичного зв'язку з головними компонентами. 2. Виділення узагальнюючих економічних показників. 3. Ранжирування результатів спостережень по головних компонентах 4. Класифікація об'єктів спостереження. 5. Список вихідної інформації. 6. Побудова рівнянь регресії за узагальнюючими економічними показниками.
9.	Причинно-наслідковий аналіз	Виявлення об'єкта та формулювання проблеми, що створюють негативні наслідки, які необхідно усунути.

10.	Системно-структурний аналіз	<p>Метод, заснований на принципі системного підходу, що складається з декількох етапів:</p> <ul style="list-style-type: none"> • уточнення того, який феномен береться для аналізу як ціле; • виявлення можливо більшого числа елементів цілого; • групування елементів у необхідне і достатнє число підструктур з узгодженням їх з наявними фактами; встановлення різних зв'язків і відносин між елементами, підструктурами та цілим.
11.	Ресурсний аналіз	Збір інформації про наявні ресурси, ресурсопотоки; побудова організаційної моделі управління ресурсами
12.	Кореляційний аналіз	<p>Головні завдання кореляційного аналізу:</p> <ul style="list-style-type: none"> • оцінка за вибірковими даними коефіцієнтів кореляції; • перевірка значущості вибіркових коефіцієнтів кореляції або кореляційного відношення; • оцінка близькості виявленого зв'язку до лінійного; побудова довірчого інтервалу для коефіцієнтів кореляції.
13.	Дисперсійний аналіз	Analysis of variance – являє собою статистичний метод аналізу результатів, які залежать від якісних ознак. Кожен фактор може бути дискретною чи неперервною випадковою змінною, яку розділяють на декілька сталих рівнів (градацій, інтервалів)
14.	Кластерний аналіз	Кластеризація – технологія, що дозволяє розподіляти вхідні дані на класи – групи однотипних екземплярів вибірки, або групування екземплярів вибірки у просторі ознак
15.	Ретроспективний аналіз	Аналіз спрямований на вивчення сформованих у минулому тенденцій
16.	Семантичний аналіз	Метод обробки інформації природною мовою, що дозволяє проаналізувати взаємозв'язок між колекцією документів і термінами, які в них зустрічаються
17.	Фінансово-економічний аналіз	Орієнтований на оцінювання фінансового стану, фінансових результатів та ефективності діяльності підприємства
18.	Контент-аналіз	<p>Content analysis – формалізований метод виявлення смислу текстової і графічної інформації. Яка міститься в документі. До основних процедур відносять:</p> <ul style="list-style-type: none"> • виявлення термінів у тексті • теми тексту • назви тексту • виявлення події, фактів основного змісту

1.2 Визначення сценарію як елемента аналітичної діяльності

Під сценарієм Анд розуміється сукупність методів опису та організації Анд (для багатофакторних Про при жорстко заданих обмеженнях), застосування яких сприяє генерації різних сценаріїв досягнення цілей аналітичного дослідження з урахуванням багатофакторності і обмежень [200]. На рисунку 1.2 наведено графічне відображення складного сценарію у вигляді можливих варіантів дій, які, залежно від настання внутрішніх і зовнішніх факторів, дають різні варіанти для кожного кроку (n).

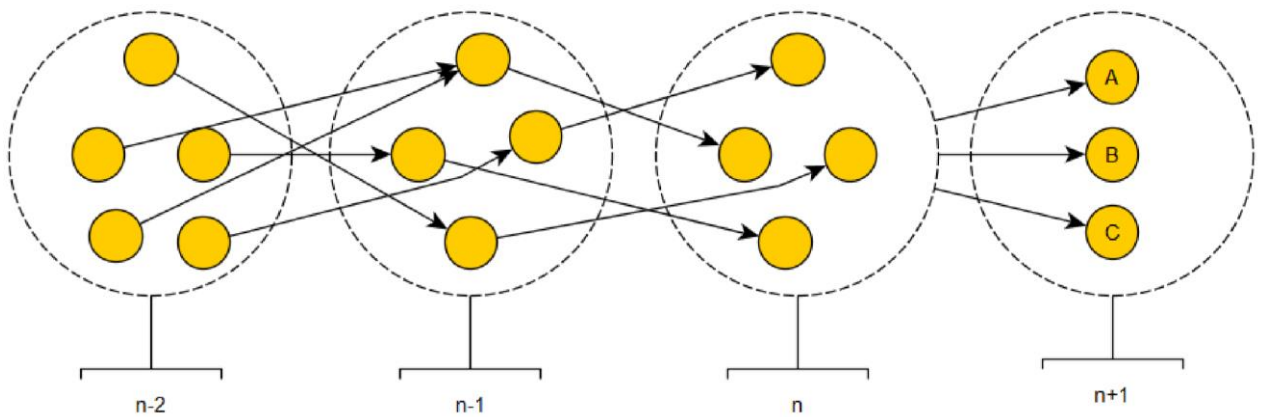


Рисунок 1.2 – Гіпотетична модель розгалуженого сценарію аналітичної діяльності

Формально аналітичний процес можна представити у вигляді вершин орієнтованого графа – $Gr(Sc)$, з'єднаних ребрами (стрілками), який відображує послідовності дій аналітика щодо досягнення мети дослідження. Для більшої реалістичності будемо вважати, що деякі дані можуть бути або втрачені, або спотворені. Спотворені дані зображені у вигляді порожнього кола (рисунок 1.2).

В контексті сценарного аналізу сценарії – це спосіб аналізу складного середовища, в якому є безліч значимих і взаємопов'язаних тенденцій і подій [167]. Відносно до галузі дослідження та класу систем сценарії поведінки системи можна розділити на три категорії.

1. Сценарії розвитку ситуації, до яких відносяться: сценарії розвитку соціально-економічної системи, сценарії розвитку технологій, сценарії розвитку регіону та ін.

2. Сценарії взаємодії користувача з системою (use-cases).

3. Сценарії «Що-якщо».

Сценарій – можливий набір подій, що визначає розвиток тих чи інших факторів, які впливають на результат діяльності.

1.2.1. Сценарії розвитку ситуації. Дослідженням методів побудови сценаріїв розвитку систем займалися вчені В.В. Кульба [258] та С.А. Юдицький, М.З. Згуровський [200, 201], К.В. Переверза [223] та П.І. Бідюк [179].

Проаналізувавши підходи до сценарного аналізу, можна виділити основні етапи побудови сценаріїв розвитку ситуації:

- визначення цілей та об'єктів дослідження, часових рамок;
- опис об'єкта/системи, визначення зовнішніх і внутрішніх трендів, встановлення факторів невизначеності;
- визначення зацікавлених сторін – агентів;
- визначення множини можливих припущень про майбутнє, що базуються на ключових змінних,
- формулювання сценаріїв;
- перевірка на протиріччя елементів сценаріїв;
- створення кількісних моделей сценаріїв;
- конструювання альтернативних стратегій, вироблення плану дій.

Дослідження показують, що в зв'язку із багатофакторністю предметних областей, дуже важко запропонувати єдину методологію сценарного аналізу, яка б чітко описувала всі етапи побудови моделі прогнозу. З методологічної точки зору це означає, що для досягнення мети, на кожному окремому етапі сценарного аналізу можуть використовуватися різні методи аналізу (рисунок 1.3.), наприклад, морфологічний і SWOT – аналіз для виявлення сильних і

слабких сторін досліджуваної системи, а також перспектив і загроз для неї у майбутньому [223].

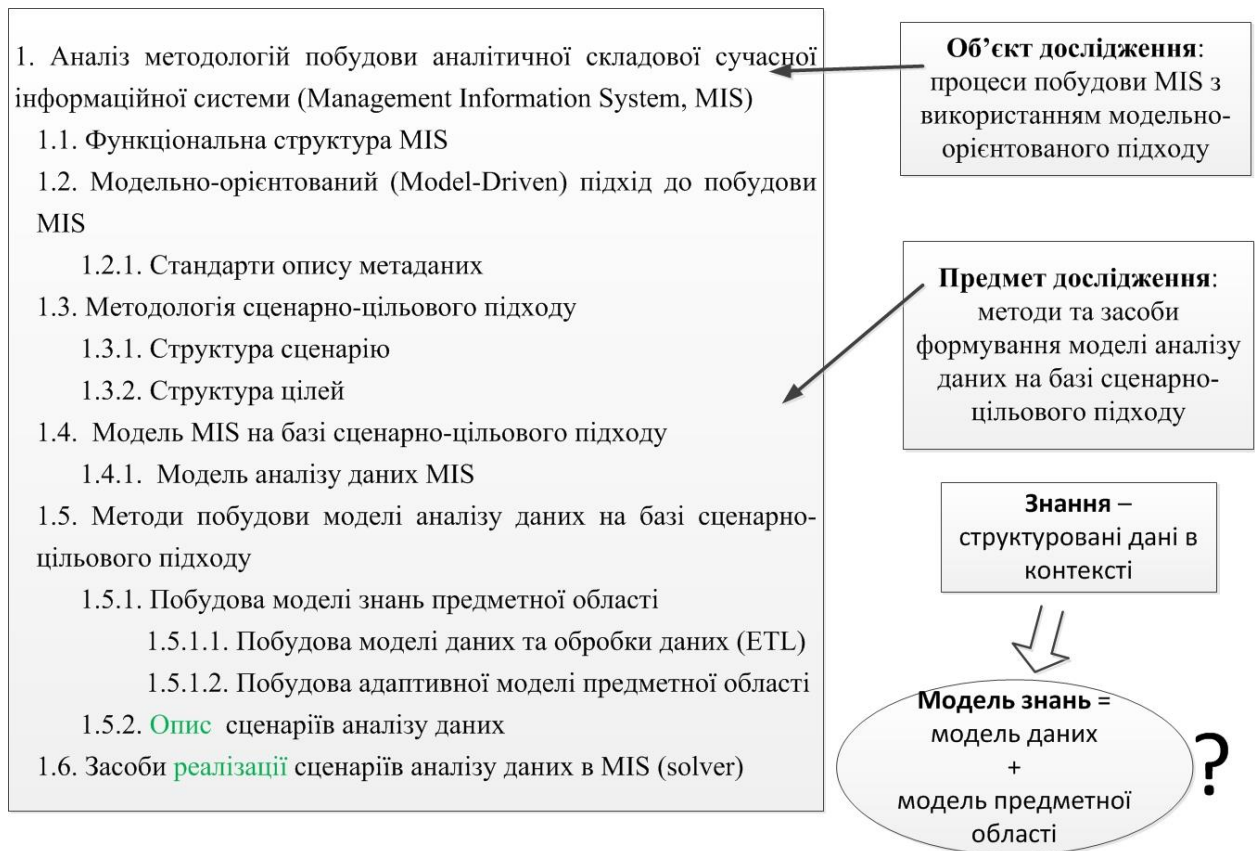


Рисунок 1.3 – Методології сценарного аналізу

Сценарний аналіз [179] поєднує логічні та евристичні методи з формальними методами інформаційної економіки, наприклад, для побудови сценаріїв розвитку ситуації у регіоні. Виходячи з особливостей ПрО, сценарій розвитку ситуації визначають як послідовний опис альтернативних гіпотетично можливих варіантів розвитку подій, як відображення різних точок зору на минуле, теперішнє та майбутнє, що може слугувати базисом для подальшого планування дій.

Такий підхід не відноситься до ІТ, методи та моделі не можуть бути використані для проектування СІУ, але з іншого боку, розробка інформаційної технології розширеної аналітики може надати інструментальні засоби побудови сценарних моделей розвитку ситуації. Сценарій розвитку ситуації не має

формалізованої структури та визначається лінгвістичним формулюванням гіпотез щодо розвитку ситуації на базі висновків експертів з різних галузей.

Моделювання сценаріїв розвитку багатоагентних тріадних мереж, що описаний С.А. Юдицьким, базується на побудові структури мережі у вигляді множини функціональних одиниць, що можуть взаємодіяти одна з одною. Дослідження С.А. Юдицького розглядається як приклад застосування сценарно-цільового підходу до формування мережі агентів. Сценарії Анд оперують інформацією про об'єкти, що моделюються, що є аналогічним до взаємодії агентів між собою шляхом передачі ресурсу або інформаційного повідомлення. Даний підхід передбачає проведення декомпозиції системи на складові частини та визначення наступних складових тріадної моделі агенту:

- склад і структура цілей, що поставлені перед агентом і причинно-наслідкові зв'язки на множині цілей;
- склад і порядок виконання дій (сценарій);
- ключові показники, що характеризують роботу агенту;
- зв'язки між цілями, діями та показниками.

Формальний апарат моделювання цілей базується на графах, позиції якого відповідають цілям c_i , модель якого представлена на рисунку 1.4.

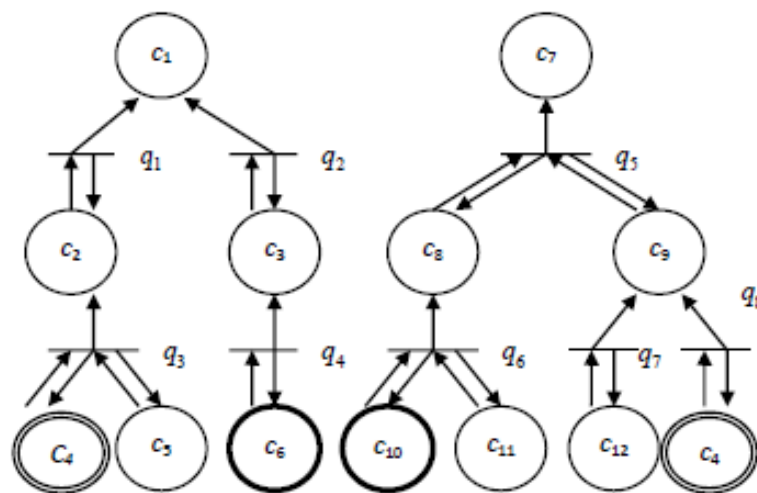


Рисунок 1.4 – Граф цілей

Для моделювання сценарію розвитку ситуації використовується граф дій, що називаються графами операцій. Приклад графа дій наведено на рисунку 1.5.

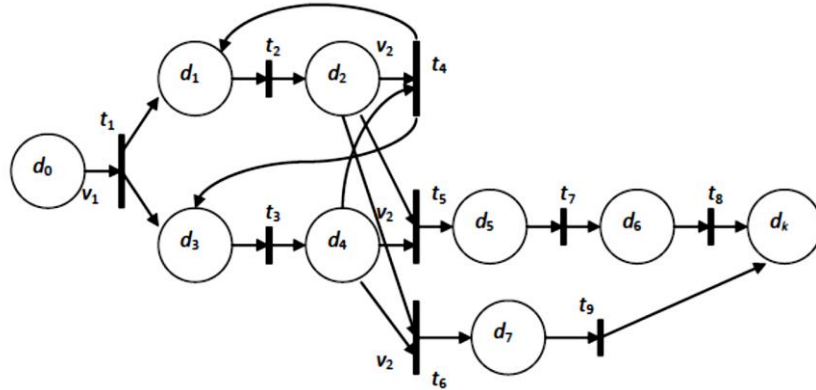


Рисунок 1.5 – Граф дій

Взаємовплив показників діяльності агента моделюється «зваженим» орієнтованим графом, вершини якого відповідають показникам, а дуги відповідають впливу показника попередника на показник послідовник, причому дуга помічається «вагою» (рисунок 1.6.).

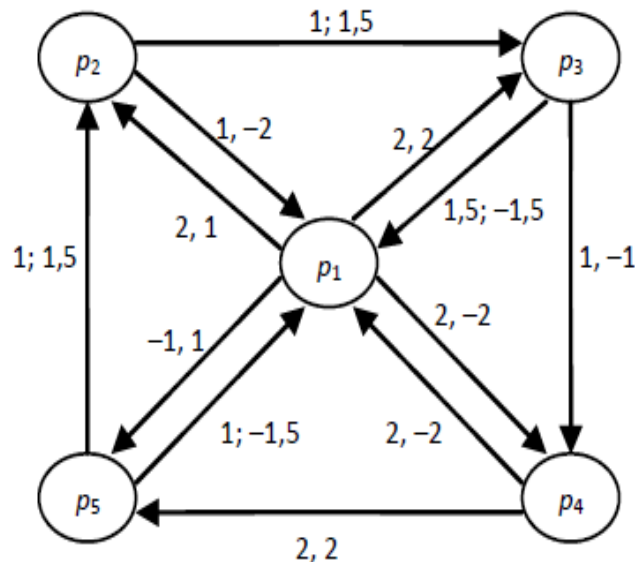


Рисунок 1.6 – Частковий випадок графа показників

Часові та ресурсні характеристики побудованої системи, конфліктні ситуації і «вузькі місця» визначаються методом імітаційного моделювання.

Для деталізації сценарію та відображення зв'язку з діаграмою цілей і показників використовується діаграма переходів (рисунок 1.7) [29], що представляє собою орієнтований граф, вершини якого відповідають переходам, а дуги помічені виконуваними діями (в квадратних лапках) і часовим проміжком між моментом виконання дій. Над вершиною переходу проставляються ініційовані ним початкові цілі, під вершиною – сформовані після переходу нові значення відхилення показників від норми.

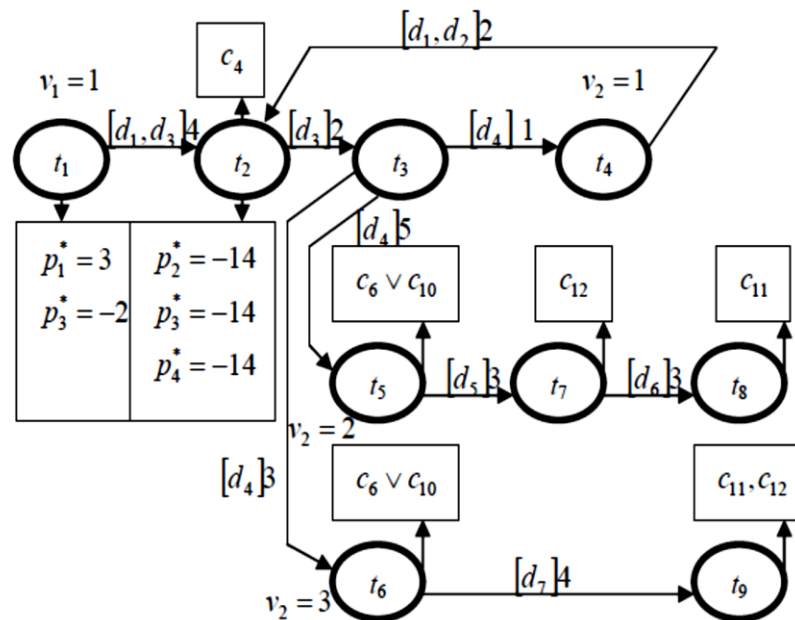


Рисунок 1.7 – Діаграма переходів

При моделюванні сценаріїв Анд більшу увагу необхідно приділити на опис, на базі якого надалі формуються сценарії, але даний підхід є функціонально-орієнтованим без можливості врахування особливостей ПрО. Основними принципами побудови моделі предметного середовища є:

- використання єдиного елемента для конструювання моделі – функціонального елемента;
- моделі описуються у вигляді графічних причинно-наслідкових структур:

– елементи моделі перераховується ітераційно.

За функціональний елемент виступає оператор присвоєння. Комп'ютерна реалізація додатку на основі принципів модельного програмування зводиться до побудови його моделі у вигляді послідовно-функціональних елементів. Запропонований підхід до конструювання допомагає неспеціалістам у програмуванні спростити алгоритмічне представлення предметної області.

Недоліки даного підходу полягають у наступному:

- в описаному підході надається можливість створювати та змінювати функціональні моделі відповідно до графічної специфікації, але нема програмного середовища для імітації та верифікації сценарних моделей, що є важливою умовою вирішення поставленої задачі;
- недостатньо використовувати лише один функціональний елемент для опису всієї множини процесів аналітичної діяльності та опису моделі предметної області.

1.2.2. Сценарії аналізу даних. Сценарій аналітичної обробки даних є моделлю обчислювального процесу та може розглядатись як поглинаючий ланцюг Маркова з дискретними станами та автоматним часом [222].

Вважається, що в такій моделі перехід із одного стану до іншого здійснюється миттєво. Тому моменти часу, в які здійснюється перехід, пов'язують з подіями, а проміжки часу між подіями – з активністю (процесом). Відповідно до цього, сценарій може бути описаний як послідовністю подій, так і послідовністю процесів. Слід зауважити, що в моделі, орієнтованій на опис подій, припускається наявність формального апарату опису процесів, а в моделі, орієнтованій на опис процесів, навпаки, – певна формалізація подій. Використання обмеженого алфавіту подій з чітко визначеною в межах протоколу взаємодії процесів семантикою забезпечує підхід до опису сценаріїв аналітичної обробки даних, орієнтованих на процеси, переваги, пов'язані з тим, що модель подається як сукупність взаємодіючих процесів і більш адекватно

(ніж сукупність подій) відображає структуру перетворень даних, які необхідно здійснити в сценарії.

Таким чином, сценарій аналітичної обробки даних є сукупність взаємодіючих процесів, кожен з яких може розглядатися як «чорна скринька» [229].

1.2.3. Сценарії «Що-якщо». Основні дослідження в напрямку аналізу «що-якщо» спрямовані на пошук адекватної формалізації для концептуального відображення імітаційної моделі розвитку складної системи.

Вивченням методології аналізу «що-якщо» та методів формування сценаріїв «що-якщо» займалися такі вчені як Ina S. Markham та Susan W. Palocsay, Marco Gavanelli, Matteo Golfarelli, Rizzi, S.

Аналіз «що-якщо» – це моделювання на великих об’ємах даних, метою якого є відстеження поведінки складної системи (підприємство чи його частина) за допомогою сформованих гіпотез, що називаються сценаріями. Аналіз «що-якщо» визначає, як зміни в наборі незалежних змінних вплинуть на набір залежних змінних в імітаційній моделі, що є спрощеним відображенням бізнес-процесів, і спроектована для відображення важливих властивостей на базі історичних даних підприємства [189].

На рисунку 1.8 зображено піраміду, в якій представлено перехід від даних до знань через бізнес-аналітику. Аналіз «що-якщо» відноситься до аналізу, що працює з та надає знання для подальшої підтримки прийняття рішень.

Методологія аналізу «що-якщо» складається з наступних етапів [15].

1. Аналіз цілей має за мету визначення бізнес-феномену та його характеристик, для якого будується імітаційна модель.

2. Бізнес-моделювання – будується спрощена модель предметної області для того, щоб допомогти проектувальнику зрозуміти бізнес-феномен, дозволяючи йому покращувати сценарії.

3. Аналіз джерел даних має за мету зрозуміти, яка інформація є доступною для виконання моделювання, як вона структурована та як вона фізично розгорнута, з необхідною увагою до історичних даних.

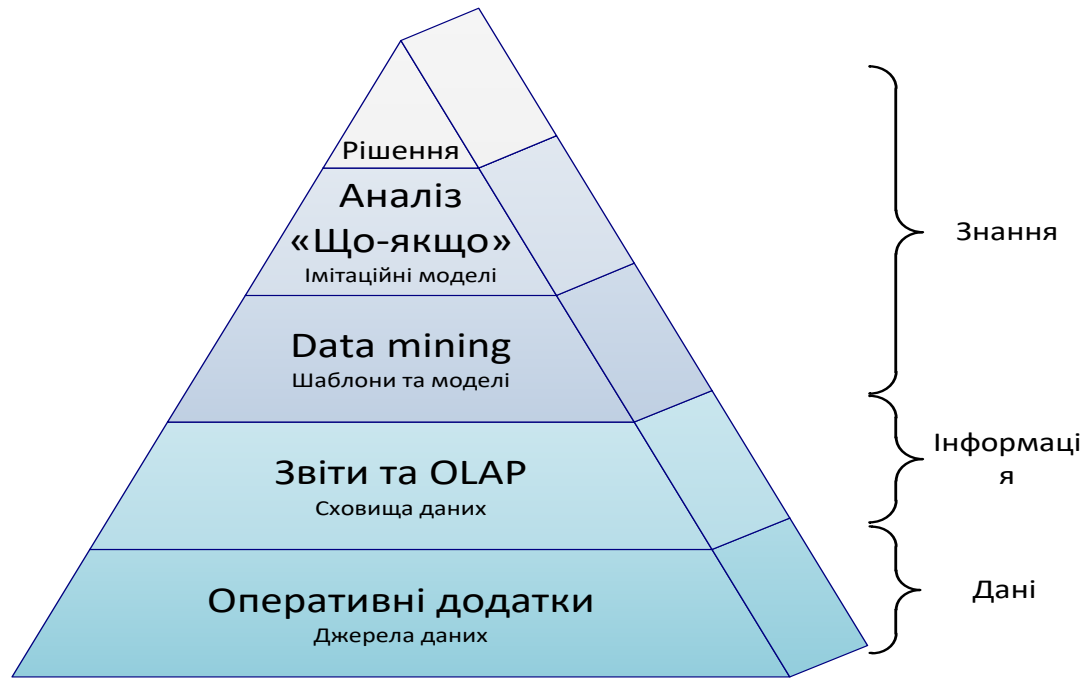


Рисунок 1.8 – Піраміда бізнес аналітики

4. Багатовимірне моделювання структурно описує передбачення, беручи до уваги статичну частину бізнес-моделі, що розроблена на другому етапі з урахуванням вимог, описаних на першому етапі.

5. Імітаційне моделювання визначає, базуючись на бізнес-моделі, імітаційну модель, що дозволяє створити передбачення для кожного сценарію, з джерел даних, що є доступними.

6. Проектування даних і виконання має за мету виконання передбачення та імітаційної моделі на платформі для створення прототипу тестування.

7. Ратифікація оцінює разом з користувачами оцінює імітаційну модель на достовірність і надійність.

У контексті наукових досліджень сценарії можуть представляти набагато складніші компоненти, які включають взаємодію множини змінних,

що впливають на хід сценарію. Для окремих сфер сценарії можуть посилатися, з одного боку, на тексти (з різним ступенем вичерпності та деталізації), а з іншого боку, термін сценарій може також стосуватися модуляцій кількісної моделі. Множинність визначення терміну сценарію безпосередньо пов'язана з множиною самих існуючих методів формування сценарію. Розглянемо основні підходи щодо формування та опису сценаріїв, які представляються доцільними при моделюванні розгалужених сценаріїв аналітичної діяльності.

Сценарна модель характеризується більш високим ступенем формалізації щодо вербальної і зачіпає переважно логічний аспект розвитку подій [35,36].

Розглянемо *концептуальну модель* сценарно-цільового підходу, яка є формалізованим описом, що задає основні *поняття* цього методу та їхні взаємозв'язки, з визначенням за кожним із них основних найбільш значущих *факторів* (рисунок 1.9). Кожен з *факторів* є визначальною характеристикою відповідного *поняття* та має своє значення – як визначення конкретного об'єкта/суб'єкта, деякого значення характеристики, функціональної або процедурної реалізації.

Концептуальна модель СЦП є основою для побудови об'єктних моделей відповідних понять з більш глибокою їхньою деталізацією. Об'єктні моделі, в свою чергу, слугують як база для проектування та програмної реалізації методології СЦП у комп'ютерних моделях ІАС.

Сценарно-цільовий підхід визначається наступними поняттями.

1. *Ціль* – загальна ціль аналітичних досліджень, що визначає кордони дослідження об'єкта, та задає критерії оцінки результатів таких досліджень.

1. Метаопис – опис предметної області досліджень у вигляді об'єктної моделі з визначенням метаданих для її представлення.

2. Сценарій – задає порядок і зміст здійснення аналітичної діяльності на функціональному рівні.

3. *Знання* – визначають фактографічний матеріал, необхідний для визначення попередніх понять.

Розглянемо більш докладно зміст і функціональну взаємодію елементів концептуальної схеми сценарно-цільового підходу.

1. Ціль визначає два основних параметри аналітичних досліджень:

1) кордони галузі дослідження ширину та глибину опису предметної області, тобто об'єм і точність метаопису об'єкта;

2) цілі окремих аналітичних досліджень визначають порядок і зміст сценаріїв аналітичної діяльності.

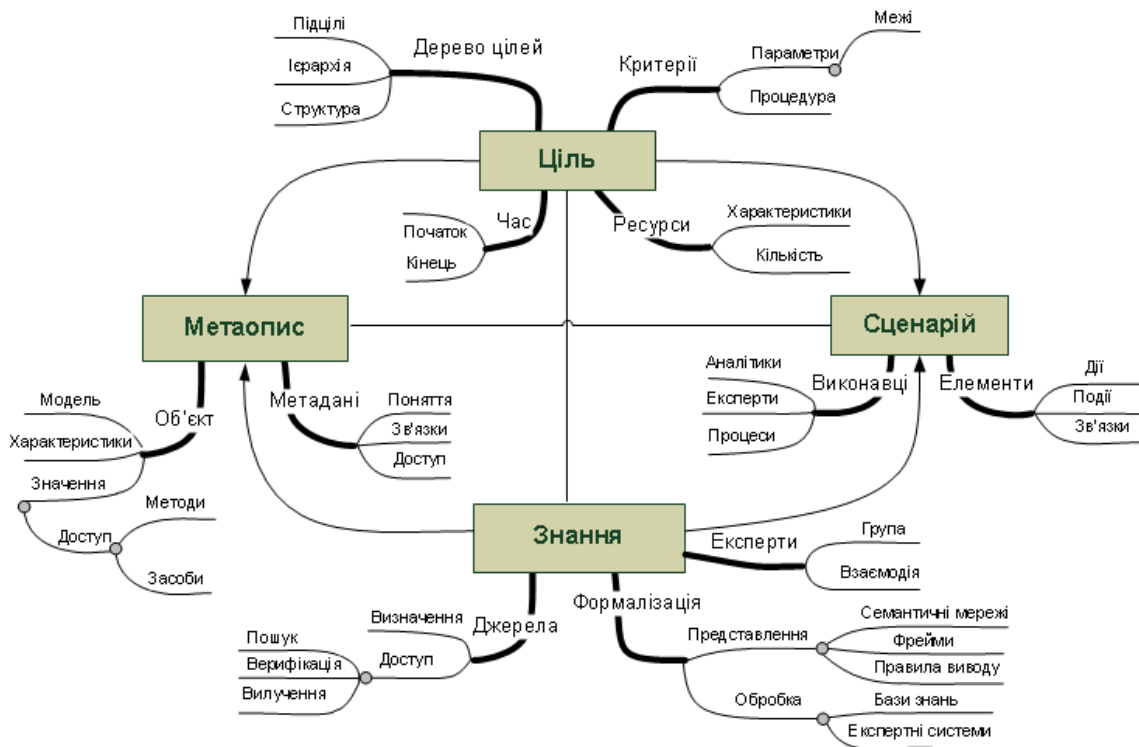


Рисунок 1.9 – Концептуальна модель сценарно-цільового підходу

Ціль задається як *дерево цілей*, з визначенням *підцілей*, що мають свою *ієрархію* та утворюють деяку *структуру*. Досягнення *цілі* визначається через *критерії*, які задаються *параметрами*, що мають свої *граничні значення*, та *процедурами* оцінки досягнення цих *критеріїв*.

Досягнення визначених *цілей* відбувається в заданих параметрах *часу* - *початку* та *кінця* періоду досягнення цієї *цілі*. Такі обмеження викликані необхідністю оперативного отримання результатів аналізу для подальшого своєчасного прийняття рішення.

Як додаткові обмеження для досягнення цілі виступають наявні *ресурси*, які визначаються своїми *характеристиками* (типом, можливостями і т. д.) і їхньою *кількістю*.

2. **Метаопис** – опис предметної області відповідно до визначених цілей аналітичних досліджень. *Метаопис* визначається *об'єктом* дослідження, який має свої *характеристики* (найбільш значущі властивості) з їхніми відповідними *значеннями*. Для представлення *об'єкта* в ІАС будується *модель*, що відображає взаємозв'язок основних *характеристик*. Така *модель* визначає концептуально-логічну структуру *БД предметної області* ІАС для опису *об'єкта*, в якій зберігаються фактографічні *значення* всіх основних *характеристик*. Для пошуку та доступу до цих *значень* використовуються відповідні методи та засоби *доступу*.

3. **Сценарій** задає послідовність здійснення аналітичного дослідження. *Сценарій* виконується *виконавцями*, якими є *аналітики*, *експерти* у відповідній *ПрО*, а також автоматизовані *процеси*, що виконуються аналітичною складовою ІАС. Опис *сценарію* складається з розгалуженої направленої послідовності *елементів*, об'єднаних між собою через *зв'язки*, і утворюють собою структуру у вигляді направленого графа. Кожен *елемент* є або деякою *дією*, що реалізується *виконавцем сценарію*, або деякою *подією*, що впливає на подальше виконання *сценарію*. *Подія* може бути спричинена як змінами зовнішнього середовища, так і зміною стану самого *об'єкта* дослідження. Кожна *дія* може бути описана як *функція*, що повертає деякий результат, або як *процедура*, що може реалізувати будь-яку послідовність операцій (із заданим рівнем їхньої деталізації).

4. **Знання** – забезпечують необхідною інформацією для формування та реалізації усіх інших *понять* СЦП. Необхідні *знання* визначаються *експертами*, які можуть об'єднуватись у *групи* для спільної *співпраці*. *Експерти* визначають можливі *джерела* необхідних для аналітичного дослідження *знань*, виконують їхній *пошук*, *виймання* знань з масиву отриманої інформації, *верифікацію* їхньої достовірності.

Математичний апарат теорії графів і теорії відносин є ефективним інструментом для аналізу проблем варіантності й активно використовується для побудови сценаріїв. Апарат теорії графів застосовується, як правило, на *перших* етапах дослідження для розробки його загальних підстав.

Для подальшого використання і автоматизованої обробки отриманих *знань* необхідно виконати їхню *формалізацію*, для чого виконується їхнє формалізоване *представлення* – опис у вигляді *семантичних мереж, фреймів, правил виводу*. Таке формалізоване *представлення знань* завантажується у *БЗ* для подальшої автоматизованої *обробки*.

Застосування сценарно-цільового підходу при побудові АНС дозволяє реалізувати моделювання АНД для досягнення заданих цілей аналізу. Сценарії аналітичної діяльності задають як послідовність дій аналітиків, так і можливу динаміку зміни об'єкта в часі через послідовність подій з урахуванням різних зовнішніх і внутрішніх умов. Таке сценарне моделювання дозволяє не тільки досліджувати об'єкт за рахунок визначення його основних характеристик і їхніх залежностей, але й прогнозувати процес їхньої зміни в майбутньому залежно від тих чи інших подій і умов.

Аналітична діяльність за використанням інформаційно-аналітичної системи – це отримання нової інформації і знань про об'єкт дослідження відповідно до цілей аналізу на основі побудови адекватної інформаційної моделі об'єкта та вивчення стану методами комп'ютерного моделювання можливих варіантів зміни його стану залежно від керуючих впливів і зміни зовнішнього середовища.

1.3 Методологічні засади моделювання сценаріїв аналітичної діяльності

В основі поширеної сьогодні модельної методології розробки ПС, що реалізують аналітичну діяльність, лежить структурно-функціональний підхід, в якому первинним є архітектура ПС, що визначається організаційно-структурною моделлю об'єкта автоматизації (предметної області). При цьому

зміна організаційно-структурної моделі об'єкта автоматизації викликає необхідність реінжинірингу ПС, що є суттєвим недоліком цього підходу. Адже для сучасної дієвої аналітики первинним фактором є сукупність сценаріїв аналізу даних, а не її організаційна структура (склад її компонентів та інформаційних потоків, що пов'язують компоненти між собою та з її оточенням). На етапі проектування сучасної аналітичної системи виникає необхідність «створення» організаційної структури ПС «під сценарій».

Найбільш близькою методологією до MDD у вітчизняних працях є модельне конструювання. Вивченням методів модельного конструювання програмних додатків займалися такі вчені як В.Д. Самойлов [232] та А.П. Писаренко [179]. Так, В.Д. Самойлов визначає модельне конструювання як інформаційну технологію конструювання моделей на базі структур з причинно-наслідковими зв'язками (функціональні моделі). Сценарій у даній технології задає процес взаємодії користувача із системою та передбачає заздалегідь заплановану реакцію на дії користувача. Модель предметного середовища, як складовий елемент модельного проектування, описує алгебраїчні структури та обчислення та є відображенням основної функціональності комп'ютерної системи.

Як функціональний елемент виступає оператор присвоєння. Комп'ютерна реалізація додатку на основі принципів модельного проектування зводиться до побудови його моделі у вигляді послідовно-функціональних елементів. Отже, модельне проектування допомагає неспеціалістам у програмуванні спростити алгоритмічне представлення предметної області.

Недоліком підходу, заснованому на модельному проектуванні, по-перше, є те що, незважаючи на можливість створювати та змінювати функціональні моделі відповідно до графічної специфікації, відсутнє програмне середовище для імітації і верифікації сценарних моделей, а, по-друге, не враховується інформаційна складова дієвої аналітики – знання та метаопис, у той час як функціональна складова представлена лише оператором присвоєння, що є

недостатнім для розробки програмного забезпечення, яке реалізує дієву аналітику.

Інший підхід до розробки ПС, заснований на *framework-for-model-driven-SOA* [6], зосереджується на виконанні таких базових вимог.

По-перше, кожна *model-driven-SOA* модель створюваної ПС повинна бути формалізованою, тобто описаною добре відомою мовою та обов'язково включати наступні компоненти:

- організаційну модель ПС – бажано в *Business Process Model and Notation* BPMN;
- модель сервісів – таблицю з властивостями (специфікація сервісів);
- модель компонент – SCA (Компонентна Архітектура Сервісів);
- модель реалізації компонент – набір з'єднаних DCL (предметно-орієнтована мова).

По-друге, кожна модель ПС повинна взаємодіяти інформаційно з попередніми моделями.

Враховуючи вищенаведене, узагальнену модель компонентів ПС, орієнтованої на Анд можна представити у наступному вигляді [45]:

$$MI^{MIS} = \bigcup_k \left\{ F_k^{MIS}, S_t^{MIS}, I_t^{nU}, Rg_k, V_k^{In}, V_k^{Out} \right\},$$

де F_k^{MIS} – набір функцій k -ої задачі інформаційного управління;

S_t^{MIS} – структурні елементи АнС;

I_k^{nU} – набір початкових налаштувань k -ої задачі інформаційного управління;

Rg_k – регламент виконання k -ої задачі інформаційного управління;

V_k^{In} – внутрішній інформаційний вплив на хід виконання k -ої задачі;

V_k^{Out} – зовнішній інформаційний вплив на хід виконання k -ої задачі

Слід підкреслити, що з позиції вимог, які висуваються до дієвої аналітики, узагальнена модель компонентів ПС не враховує наявність або необхідність

побудови сценаріїв розвитку бізнес-процесів аналізу інформації. Це не дозволяє ефективно побудувати сценарії аналізу інформації, базуючись на технологічній (інформаційної) потужності організації. Така модель більш відповідає структурно-функціональному підходу до проектування аналітичних додатків [181], який не враховує знання як про дії користувачів, так і особливості роботи сервісів.

Тому, враховуючи особливості сучасної Анд, заснованої не тільки на обробці різноманітних даних, але й на знаннях, в якості концептуальної моделі запропоновано інтегровану модель аналітичної діяльності, яка враховує не тільки часові обмеження, що накладаються на систему, а й динамічно змінювану в часі компоненту – знання про предметну область:

$$M^{MIS} : \left\{ \begin{array}{l} Kn^M, D_i^M, Pr_i^M, Sc_i^M, C_i^M, Dv^M \\ i = \overline{1, N} \end{array} \right\},$$

де Kn^M — модель знань, що визначає дві категорії знань:

- знання про технології АнС – сервіси, інформаційні технології, модулі, програмні засоби, БД, БЗ, сховища даних, великі дані;
- знання про організаційну структуру – виконавці сценаріїв, бізнес-правила);

D_i^M – об’єктна модель i -ої предметної області, що відповідає за опис понять і метаданих для доступу до даних;

Pr_i^M – модель прецедентів використання, що є декларативним описом функціональних вимог до системи та логіки взаємодії користувача з елементами графічного інтерфейсу;

Sc_i^M – сценарна модель i -ої задачі аналізу інформації, що формалізує процес аналізу інформації і виражається в логічно-впорядкованій послідовності подій у процесі аналізу даних з метою отримання необхідної інформації для прийняття обґрунтованих рішень;

C_i^M – компонентна модель АнС, як сукупність компонентів, необхідних для забезпечення функціональних та інших вимог, яка побудована та функціонує відповідно до певних правил створення компонентних конфігурацій і компонентної взаємодії [15];

Dv^M – модель реалізації компонент, що також включає опис інформаційної взаємодії об'єктів та елементів системи.

На рисунку 1.10 показано багаторівневу інтегровану модель ПС, зокрема за використанням якої здійснюється АнД з урахуванням впливу динамічних змін. На першому етапі будується модель знань Kl^M (включаючи модель онтології). Модель включає в себе знання про технологічну базу в організації, необхідну для формування реального сценарію АнД: програмні модулі, комп'ютерні системи та інформаційні технології, що є в наявності, знання про виконавців (кадрів) та знання про методи обробки і аналізу даних.

Такі знання надалі використовуються для побудови сценаріїв аналізу інформації, визначення елементів сценарію, для яких необхідно розроблювати додаткові програмні засоби, сервіси та для яких необхідно залучити експерта з певної проблеми.

На другому етапі будується об'єктна модель ПрО, що представляє собою набір понять спеціалізованої області дослідження [45]:

$$D^M = \bigcup_i \{En_i, R_{ij} \mid i = \overline{1, N}; j = \overline{1, N}; i \neq j\},$$

де En_i – деяке поняття предметної області; R_{ij} – зв'язок між i -им та j -им поняттями предметної області.

Згідно з концепцією побудови адаптивних об'єктних моделей [188] атрибути об'єктів і типи атрибутів є окремими від об'єкта елементами:

$$En = \bigcup_k \{At_k^{En}, Atp^{At}, Vl^{At} \mid k = \overline{1, N}\},$$

де At_k^{En} – атрибут поняття, що відноситься до визначеної ПрО;

Atp^{At} – тип відповідного атрибуту;

V^{At} – значення відповідного атрибуту (через засоби доступу до даних).

Сценарна модель відображає порядок і зміст проведення інформаційного управління на функціональному рівні.

$$Sc^M = \bigcup_i \{TSc_i, ASC_i, Ex_i, G_i \mid i = \overline{1, N}\},$$

де TSc_i – типовий сценарій виконання i -го завдання аналізу інформації;



Рисунок 1.10 – Багаторівнева інтегрована модель програмних систем

ASC_i – розширений (вторинний) сценарій виконання i -го завдання аналізу інформації;

G_i – ціль сценарію виконання i -го завдання аналізу інформації, що має такі характеристики як: критерії, час і ресурси.

Сценарій аналізу інформації має загальний вигляд:

$$Sc = \bigcup_i \left\{ El_i, Ex_i, F^{El}(null | Ob_1, \dots, Ob_n) \mid \right. \\ \left. i = \overline{1, N}; j = \overline{1, N}; i \neq j \right\}$$

де El_i – елемент сценарію аналізу інформації, що може бути дією, подією або зв'язком;

Ex_i – виконавець елемента сценарію, який може бути бізнес-аналітиком, аналітиком даних, експертом, працівником організації та процесом;

$F^{El}(null | Ob_1, \dots, Ob_n)$ – функція k -ої задачі аналізу інформації, що може приймати як вхідні значення поняття предметної області або пусте значення.

Модель прецедентів відповідає за взаємодію користувача з графічним інтерфейсом при виконанні задач аналізу інформації. Модель прецедентів застосовується для виявлення вимог до поведінки системи, відомих також як функціональні вимоги:

$$Pr^M = \bigcup \left\{ Ac_i, P_i^{Ac}, Rl_{ij}^{Ac}, I_k^{nU}, F_i^P \mid \right. \\ \left. i = \overline{1, N}, j = \overline{1, N} \right\},$$

де Ac_i – актор, який відноситься до учасників сценарію певної предметної області: $Ac_i \leq Ex_j$;

P_i^{Ac} – прецедент використання, що відповідає за послідовність дій користувачів (акторів) АНС і реалізується певним сценарієм виконання i -го завдання контрольно-управлінського аналізу інформації: $P_i^{Ac} \geq Sc_j$;

F_i^P – екранні форми користувачів для взаємодії з АНС;

Rl_{ij}^{Ac} – відношення між прецедентами використання:

$$Rl_{ij}^{Ac} = \{include, extend, generalize, normal\}$$

де *normal* – комунікаційний зв'язок між прецедентом та актором, який в ньому бере участь;

extend – включення додаткової поведінки в базовий прецедент, який про це не знає;

include – включення додаткової поведінки в базовий прецедент, що детально описує дане включення;

generalization – взаємозв'язок між загальним прецедентом і більш специфічними прецедентами, що наслідують та додають властивості до загального прецеденту;

Компонентна модель застосовується при визначенні архітектури реалізації сценаріїв у розподіленому середовищі й побудові комп'ютерної моделі системи реалізації сценаріїв і визначається як сукупність вимог до компонентів УІС та правил взаємодії і побудови компонентних конфігурацій.

Модель реалізації компонент Dv^M включає опис інформаційної взаємодії об'єктів та елементів системи та відображається у фабрику програмного забезпечення (software factory [45]):

$$Dv^M = \bigcup_i \{St_k^{MIS}, I_k^{nU}, V_i^{In}, V_i^{Out} \mid i = \overline{1, N}\},$$

де St_k^{MIS} – структурні елементи АНС;

I_k^{nU} – набір початкових налаштувань k -ої задачі аналізу інформації;

V_k^{In} - сценарії взаємодії компонентів АНС;

V_k^{Out} – сценарії взаємодії АНС з зовнішніми середовищами.

Відповідно до другого принципу модельної розробки необхідно передбачити та визначити інформаційну взаємодію між моделями. Пропонується три критерії перевірки комп'ютерної моделі АНС.

1. Цілісність – висловлює інтегрованість, самодостатність та автономність об'єктів системи, їхнє протипоставлення оточенню, що пов'язане з їхньою внутрішньою активністю. Цілісність характеризує якісну своєрідність об'єктів системи, що обумовлено притаманним їм специфічним закономірностям функціонування та розвитку.

2. Повнота – характеристика, що визначає кількість інформації, необхідної і достатньої для прийняття правильних рішень.

3. Несуперечність визначається передбачає неможливість виведення з однієї сценарної моделі двох результатів, що виключають один одного.

Відповідно до критеріїв верифікації запропоновано наступні етапи перевірки сценарних моделей:

1. Перевірка синтаксису сценарію, а саме перевірка:

- повноти інформації по кожному елементу сценарію;
- наявності більш ніж одного елемента сценарію;
- взаємозв'язків між елементами;
- участі актору хоча б в одному елементі сценарію;
- того, що кожен елемент сценарію використовує ресурс (як ресурс може

виступати інформаційний ресурс або кількісне значення);

- наявності необхідних ресурсів для виконання елемента сценарію;

2. Перевірка семантики сценарію, а саме перевірка:

- відповідності сценарію загальній цілі та елементу сценарію - підцілі;
- того, що виконані попередні умови для виконання елемента сценарію;
- того, що кожному елементу сценарію відповідає певна функція (модуль

або сервіс);

3. Перевірка посилань на коректність.

1.4 Аналіз сучасного стану підходів до моделювання процесів обробки даних

Розробка програмних засобів моделювання сценаріїв аналітики в корні відрізняється від традиційного програмування аналітичних застосунків. Це обумовлено підвищенням вимог до інтелектуалізації самих методів обробки різноманітних типів даних. Природно, що для прискорення розробки та зниження кінцевої вартості спеціалізованого ПО АНС розробник вимушений використовувати різноманітні ІЗ, які довели свою ефективність. В останні роки від провідних виробників, таких як IBM Cloud service models, Microsoft Azure, Google Cloud, Oracle Cloud, Amazon Web Services (AWS) спостерігається тенденція надання інтегрованих засобів у вигляді «хмарних» Cloud-сервісів. Ці

сервіси орієнтовані на широке коло користувачів, включаючи і розробників програмних систем та аналітичних додатків. Розглянемо найвідоміші Cloud-сервіси, їхні специфікації, переваги та особливості застосування в напрямку створення спеціалізованих застосунків, які мають входити до складу сучасних багатофункціональних АНС.

1. Платформа IBM Cloud service models [124] пропонує три класи сервісів, які можуть використовуватися як базові технології для побудови АНС [127]:

- Infrastructure as a service (IaaS) – в якій постачальник надає користувачам доступ до обчислювальних ресурсів (сервери, сховища та мережі) замість розгортання власної платформи та системного забезпечення;

- Platform as a service (PaaS) – забезпечує інструментами для тестування, розробки та розміщення додатків в одному середовищі;

- Software as a service (SaaS) – надає користувачам доступ до вже існуючих у вигляді бібліотек спеціалізованих IBM Cloud сервісів через Інтернет або через API. При цьому потрібні конфігурації сервісів не встановлюються на пристроях користувачів, а існують у хмарному просторі.

Сучасні тенденції залучення методів штучного інтелекту (ШІ) для перетворення даних у знання, потрібні для прийняття рішень, все більш опановують в багатьох сферах аналітики. Одним із лідерів у цьому напрямку є IBM Watson Studio, яка класифікується як множина послуг типу PaaS [56]. IBM Watson® Studio надає аналітикам і фахівцям з обробки даних цілу низку технологічних інструментів для підготовки даних і створення моделей, а саме:

- створення моделей процесів підготовки та обробки структурованих даних за допомогою візуального редактора IBM SPSS® Modeler;

- розробки моделей слабо структурованих даних на основі образів за допомогою IBM Watson Visual Recognition і текстів за допомогою IBM Watson Natural Language Classifier;

- підтримки процесів навчання та виконання моделей обробки даних за допомогою засобів IBM Watson Machine Learning.

Крім того, такі сервіси як IBM Cloud Pak™ for Data (уніфікована платформа даних) надають послуги в організації, зберіганні та підтримці даних для аналізу на всіх етапах життєвого циклу. На рисунку 1.11 показано загальний сценарій взаємодії процесів збору, моделювання, машинного навчання та обробки даних за технологією IBM Watson Studio.

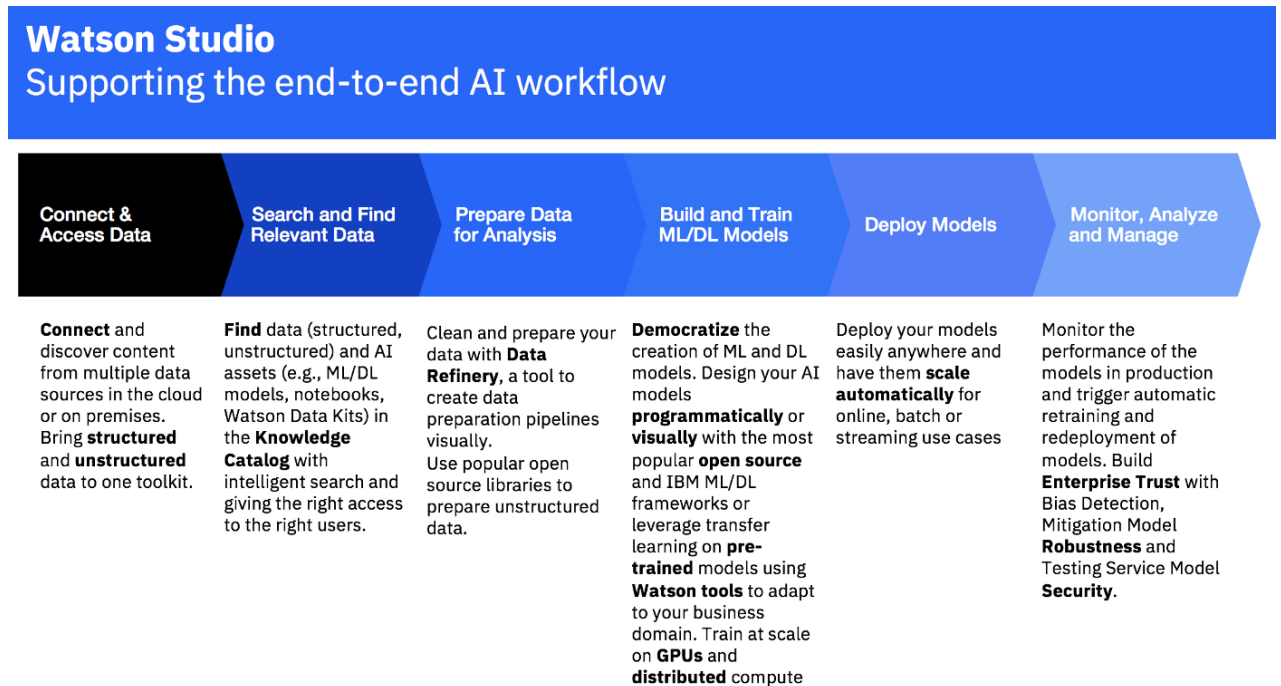


Рисунок 1.11 – Загальний сценарій взаємодії процесів збору, моделювання, машинного навчання та обробки даних за технологією Watson Studio

Тобто раціональне використання мультіхмарної архітектури Watson Studio (в процесі розробки рішень для АНС) може суттєво скоротити витрати коштів і часу, а також підвищити якість аналітики за рахунок залучення вже відпрацьованих і перевірених рішень.

2. Платформа Microsoft Azure [148] теж підтримує всі три класи хмарних сервісів: IaaS, PaaS та SaaS:

– Модель IaaS – інфраструктура як сервіс, підтримується через портал Azure <https://azure.microsoft.com> у вигляді WebApps та WebJobs, Node.js або Python і розгортати їх з використанням FTP, Git, Mercurial, Team Foundation Server; віртуальних машин, зміст яких контролюється користувачем;

– PaaS – платформа як сервіс для розгортання сервісів зберігання (Storage Services) Table Service, для збереження неструктурованих текстів у вигляді BLOB-об'єктів - Blob Service, до яких можна отримати доступ по HTTP, файлова служба (File Service), безкомп'ютерні обчислювальні середовища.

– Модель SaaS пропонує готові програмні рішення через Інтернет, які можна налаштувати за вимогами користувача. Послуги надаються за запитом, залежно від моделі підписки та розміщених служб, сервісу хостингу веб-сайтів, який дозволяє розробникам створювати сайти з використанням .NET, PHP.

Платформа Microsoft Azure особу увагу приділяє сервісам, пов'язаним з розширеною аналітикою та машинними методами глибокого навчання. Це програмні сервіси Azure ML та VS Code Azure Machine Learning (AML), які підтримують мову Python. Крім того, окремим напрямом, який є дуже цікавим для аналітиків – є процес обробки і аналізу даних групи (Team Data Science Process – TDSP) [156]. Microsoft позиціонує цей напрям як ітеративну методику обробки і аналізу даних для надання рішень для прогнозу аналітики й інтелектуальних додатків з метою використання переваг уже відпрацьованих програм аналітики. Як стверджує Microsoft, процес TDSP надає «систематичний підхід для спільної роботи груп над створенням інтелектуальних додатків» [156]. Більш того, враховуючи багатоетапність і складність самих процесів обробки даних, Microsoft розвиває напрям формування різноманітних сценаріїв. У цьому контексті сценарій трактується не як скрипт виконання аналітичної процедури, а значно глибше – як варіант робочого процесу обробки даних на основі характеристик даних, особливостей архітектури та цільових репозиторіїв Azure. Ці сценарії також включають цикл машинного навчання для підвищення якості методів аналізу даних. Отже, в публікації від Microsoft Azure за 2020 рік [148] наведено 7 типів сценаріїв обробки структурованих даних, які потребують попереднього навчання на платформі Azure Machine Learning. На рисунку 1.12 показано приклад подання сценарію роботи з великим набором даних в локальній БД SQL Server, на віртуальній машині Azure та AML в умовних позначеннях Microsoft Azure.



Рисунок 1.12 – Сценарій роботи з великим набором даних в локальній БД SQL Server

– Google Cloud Platform [98] — набір хмарних служб, що виконуються на Google інфраструктурі: Google Search та YouTube. Платформа Google Cloud, як і вищенаведені IBM Cloud та Microsoft Azure, теж надає низку сервісів, які трактуються як: платформа як сервіс – PaaS та безкомп’ютерні обчислювальні середовища. Cloud Functions – безсерверні функції, які запускаються при певних хмарних подіях; інфраструктура як сервіс – IaaS у вигляді App Engine для хостингу застосунків, Storage – доступ до файлів і вмісту СД, BigQuery – аналітика для БД, Bigtable – масштабованої NoSQL бази даних, Compute Engine, Kubernetes Engine — система автоматичного розгортання, масштабування та управління та багато інших.

Платформа Google Cloud органічно зв’язана з Cloud AutoML [64] — платформою машинного навчання, яка дозволяє аналітикам з обмеженим досвідом використовувати технології навчання та створення нейронних мереж. Cloud AutoML заснований на новітній технології трансферного навчання і пошукової нейронної архітектури Google. [161].

Другою платформою машинного навчання є TensorFlow – відкрита програмна бібліотека, від компанії Google з підтримкою мови Python [161]. TensorFlow від Google найвідоміша бібліотека глибокого навчання. За допомогою якої можна побудувати аналітичний застосунок на моделі нейронної мережі з можливостями навчання. Архітектура Tensorflow складається з трьох компонент: засоби попередньої обробки даних, формування моделі, тренування моделі на достовірних даних та оцінка якості моделі. Всі операції в TensorFlow виконуються послідовно та представляються у вигляді графа, де кожна операція (операційний вузол) і пов’язана між собою. Граф

збирає та описує всі обчислення тестової серії, зроблені під час навчання та використовує їх при аналізі даних.

На рисунку 1.13 показано виконання графа в умовних позначеннях TensorFlow на прикладі сценарію «Розробка машинного навчання: безперервний цикл».

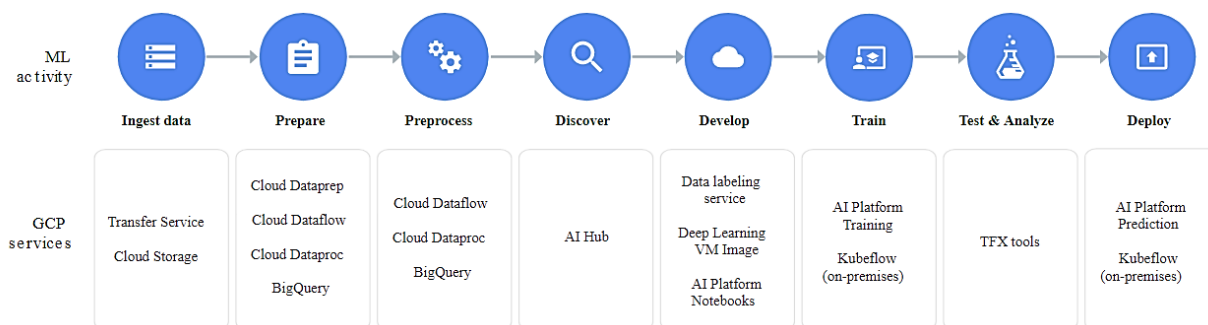


Рисунок 1.13 – Приклад відображення сценарію «Розробка машинного навчання: безперервний цикл» в умовних позначеннях TensorFlow

Доцільно нагадати, що, як правило, рішення, закладені в реалізацію Cloud-сервісів, потребують дотримання певних сценаріїв – послідовності дій для отримання бажаних результатів.

3. Платформа Amazon Web Services (AWS) – це комерційна публічна хмара, яка підтримується та розвивається компанією Amazon з 2006 року [1, 254]. AWS надає в оренду приватним особам, компаніям та урядам свої сервіси на основі платної підписки.

– Amazon Elastic Compute Cloud (EC2) – це сервіс IaaS, що надає в користування віртуальні сервери, які контролюються API, основані на гіпервізорі Xen. Еквівалентні віддалені сервіси включають Microsoft Azure, Google Compute Engine та Rackspace, а також рішення, які встановлюються на локальні сервери OpenStack або Eucalyptus.

– Amazon Lambda (AWS Lambda) – платформа безсерверних обчислень. Запускає код у відповідь на внутрішні або зовнішні події, такі як http-запит, відкрито надаючи необхідні ресурси [1]. Lambda глибоко інтегрована з AWS, але подібні сервіси, такі як Google Cloud Functions та відкриті рішення типу OpenWhisk, набирають популярність.

Аналітика в AWS представлена такими спеціалізованими застосунками як: Amazon Athena — інтерактивна служба запитів, яка полегшує аналіз даних, використовуючи стандартний SQL; Amazon Elastic MapReduce (EMR) — надає PaaS-сервіс для розміщення хостингу програм, який являє собою каркас Hadoop для запуску запитів MapReduce та інфраструктури EC2 та Amazon S3; Amazon Machine Learning — допомагає розробникам усіх рівнів використовувати технологію машинного навчання; Amazon QuickSight — інструмент бізнес-аналізу, аналітики та візуалізації.

Отже, аналіз відомих платформ IBM Cloud, Microsoft Azure, Google Cloud, Oracle Cloud або Amazon Web Services, показує, що для розробника АНС найбільш цікавою є множина послуг типу PaaS. Підхід на основі PaaS надає середовище розробки, яке розглядається в як частина віртуальної «платформи» програмування, де зберігання програмного коду, компіляція та, зазвичай, виконання програми здійснюються на серверній стороні платформи. Розробник АНС, без зайвих витрат, отримує доступ до ОС, СУБД, зв'язного ПЗ, складних методів аналізу даних, засобів розробки та тестування ПЗ. При такому підході користувачами PaaS-послуг є програмісти, які на клієнтській стороні працюють з візуальними редакторами або веб-інтерфейсами, розміщеними у хмарних провайдерах. Візуальне програмування здійснюється через інтерфейс, в якому моделі процесів обробки даних створюються шляхом об'єднання типових віджетів (операторів обробки даних) або їхнього власного набору.

Іншим шляхом прискорення розробки та зниження вартості аналітичних застосунків є залучення спеціалізованих платформ інтелектуального аналізу, в яких реалізовані методи Data Mining, Text Mining, Web Mining [103, 164]. Найбільш відомими платформами, які, до речі, дотримуються парадигми візуального програмування, є Rapid Miner, WEKA, Google Analytics, SimilarWeb, Majestic, Scrapy, Orange, Oracle Data Mining та інші, які теж можна класифікувати як послуги типу PaaS. Більшість із цих ІЗ позиціонуються як ліцензійні, але є такі як Free and Open Source, хоча вони пропонують доволі повну функціональність. Саме ці ІЗ викликають особливий інтерес, оскільки, ці

продукти цілком можливо застосовувати при моделюванні та відпрацюванні типових сценаріїв обробки даних з метою перевірки рішень для створюваної АНС.

1. **RapidMiner** [121] є програмною платформою Technical University of Dortmund, яка забезпечує інтегроване середовище візуального моделювання процесів аналізу даних, машинного навчання, ШІ, аналізу текстів, прогнозу аналітики і бізнес-аналітики. Крім того, **RapidMiner** спрямований на швидку розробку додатків і підтримує всі етапи процесу видобутку даних, включаючи результати візуалізації, перевірки та оптимізації. **RapidMiner** написана мовою Java і в даний час поширюється за ліцензією GNU Affero General Public License. До всіх основних функцій RapidMiner забезпечує доступ через Java API і версію програми для командного рядка (а не тільки через загальний користувацький інтерфейс) [121].

RapidMiner реалізує більше 500 різноманітних операторів (віджетів) обробки як структурованих, так і неструктурованих даних, включаючи текстові дані та веб-ресурси [121]. Вихідні дані та результати аналізу можуть бути представлені різними форматами CSV, Excel, XML, SAS, Access, AML, ARFF, XRFF, Database, SPSS, Stata, Sparse, dBase, C4.5, BibTeX, DasyLab, URL, включаючи роботу з хмарними сховищами і управління процесами та групами об'єктів. Оператори розділені на функціональні групи: Data Access, Blending (маніпулювання), Cleansing (очищення), Modeling (моделювання), Scoring (оцінювання), Validation (перевірка коректності моделі) та інші.

Процес аналізу даних представляється у вигляді інтерактивного графа операторів обробки даних. RapidMiner надає можливість завантаження даних із БД, CSV, Excel, XML, або хмарних сховищ (Amazon S3, Azure Blob, Dropbox). Основним робочим елементом є процес (<Process>, в основі якого налагодження операторів (віджетів) (рисунок 1.14.).

RapidMiner може використовуватися і як окремий додаток, і як «інтелектуальний двигонок», вбудований в інтегроване середовище для вирішення завдань аналізу великих даних. Отже, завданнями RapidMiner

можуть бути як дослідні застосування – для моделювання процесів обробки різноманітних за структурою даних (включаючи й великі дані), так і прикладні. До недоліків Rapid Miner можна віднести наступні:

- розмір даних не може бути більшим ніж 1 гігабайт;
- неможливість роботи із технологією Hadoop (інтеграції файлової системи Hadoop у Rapid Miner).

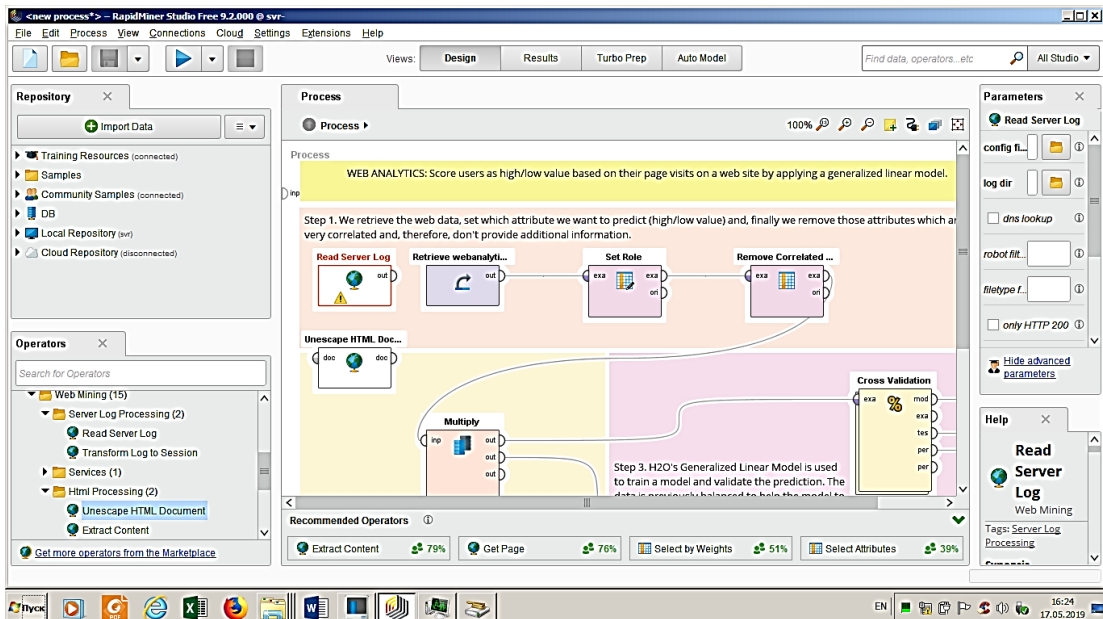


Рисунок 1.14 – Формування моделі процесу обробки даних в RapidMiner

2. KNIME Analytics Platform – вільно поширюваний програмний пакет [74] з графічним інтерфейсом, що підтримує цикл інтелектуального аналізу даних, а саме: доступ до різних форматів даних; трансформацію даних; аналітичні функції; візуалізація даних, підготовку звітів. Ідеологічною основою KNIME є поняття потоку робіт – workflow який графічно відображує процеси перетворення і обробки вихідних даних (рисунок 1.15). Графічне відображення складається з вузлів (віджетів KNIME), з'єднаних стрілками. Після того як workflow створений – він запускається на виконання. У базовому сценарії workflow відпрацьовуються, починаючи з першого. Якщо в ході виконання вузла сталася помилка, то виконання всієї гілки наступної за ним припиняється. Вузол інкапсулює деяку операцію над даними, стрілки показують шлях даних.

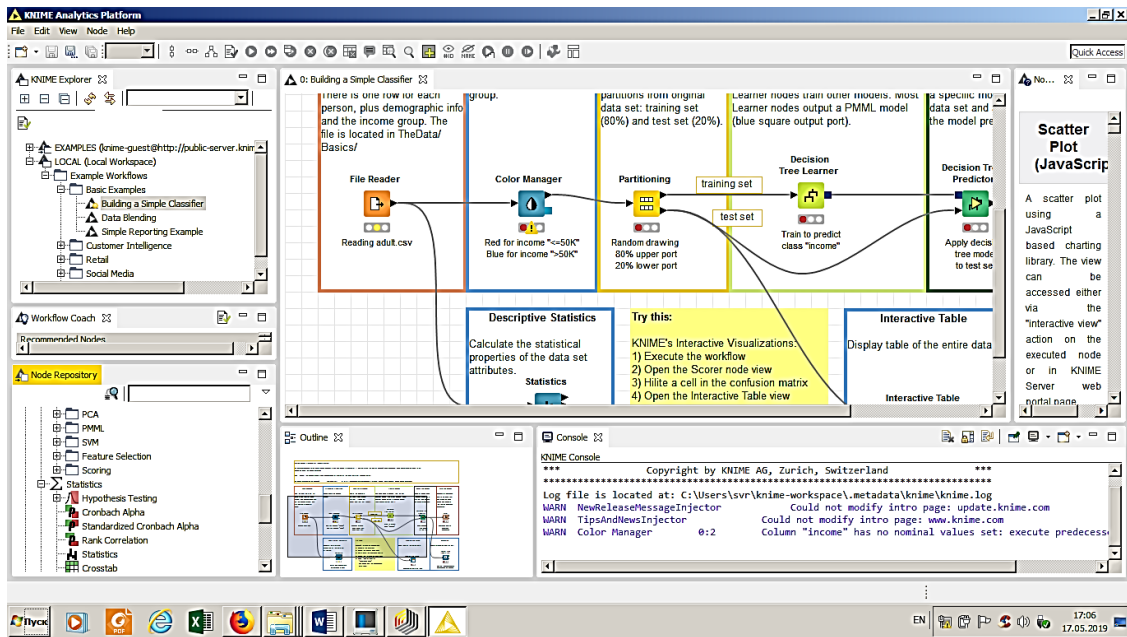


Рисунок 1.15 – Процес формування моделі обробки даних на платформі KNIME

Платформа KNIME містить плагін для обробки тексту KNIME Text Processing. Плагін, як і вся система KNIME, має відкритий код і підтримує багатоступеневий процес обробки тексту – від читання та розбору через розпізнавання сутностей, фільтрації і маніпуляції до підрахунку кількості слів, виділення ключових понять і, нарешті, візуалізації результатів аналізу тексту.

Як і в інших програмах інтелектуального аналізу даних, у KNIME Analytics Platform присутній візуальний редактор. Окремі елементи, які виконують роботу виконують одне завдання – node. Для імплементації функціонала KNIME Analytics Platform у додаток можна використовувати REST API. KNIME Analytics Platform надає більш 200 віджетів розподіленими за функціональними категоріями: аналітика (інтелектуальний аналіз даних та статистику); робота с БД; робота з файлами; маніпуляція із таблицями тощо.

KNIME Analytics Platform надає гнучкий підхід до побудови алгоритмів аналізу, перетворення і візуалізації даних, але на складних workflow можна зіткнутися з наступними проблемами:

- недостатньо гнучка поведінка вузлів – оскільки існують нюанси їх налаштування (якщо не аналізувати програмний код);

– програмування складних workflow може вилитися в довгу і складну діаграму з циклами, що призводить до складності читання і налагодження. Тому в окремих випадках для розв’язання таких конфліктів потребується написання коду на R або Python. Функціональність можна розширити за допомогою великої кількості існуючих plugin. Зокрема, конструктор звітів дозволить експортувати внутрішні документи в формати DOC, PPT, XLS, PDF і ін.

3. **Orange** – це кросплатформене ПЗ для ШІ і машинного навчання з відкритим вихідним кодом, [114]. Версії Orange 3.0 включають у себе основні компоненти на C++ з обгортками в Python. Починаючи з версії 3.0, Orange використовує загальні Python-бібліотеки з відкритим вихідним кодом для наукових обчислень, такі як Numpy, SciPy і scikit-learn, у той час як його графічний користувацький інтерфейс працює завдяки крос-платформній Qt framework. Моделювання процесів обробки даних в Orange виконується за допомогою візуального редактора (рисунок 1.16). Візуальне програмування здійснюється через інтерфейс, в якому моделі процесів обробки даних створюються шляхом об’єднання стандартних віджетів. Для чого широко застосовує віджети з бібліотеки Python. Це позиціонує Orange як потужний інструментарій візуального програмування процесів обробки даних. За функціональною ознакою віджети в Orange поділені на наступні категорії:

- робота з даними (31 віджет);
- візуалізація (17 віджетів);
- моделювання (15 віджетів);
- оцінювання (6 віджетів);
- навчання без вчителя (14 віджетів).

Функціонал Orange імплементується до аналітичного застосунку за допомогою механізмів, реалізованих в API Orange. Оскільки платформа Orange написана мовою Python, це надає декілька важливих переваг. По-перше, мова Python є однією із найпоширеніших мов у сфері обробки великих даних, тобто можна використовувати вже готові рішення простим підключенням до аналітичного додатку існуючих віджетів. По-друге, розроблюючи свій власний

додаток мовою Python, неважливо веб-додаток чи десктопне рішення, функціонал Orange можна викликати як звичайний модуль наступним текстом: «import Orange».

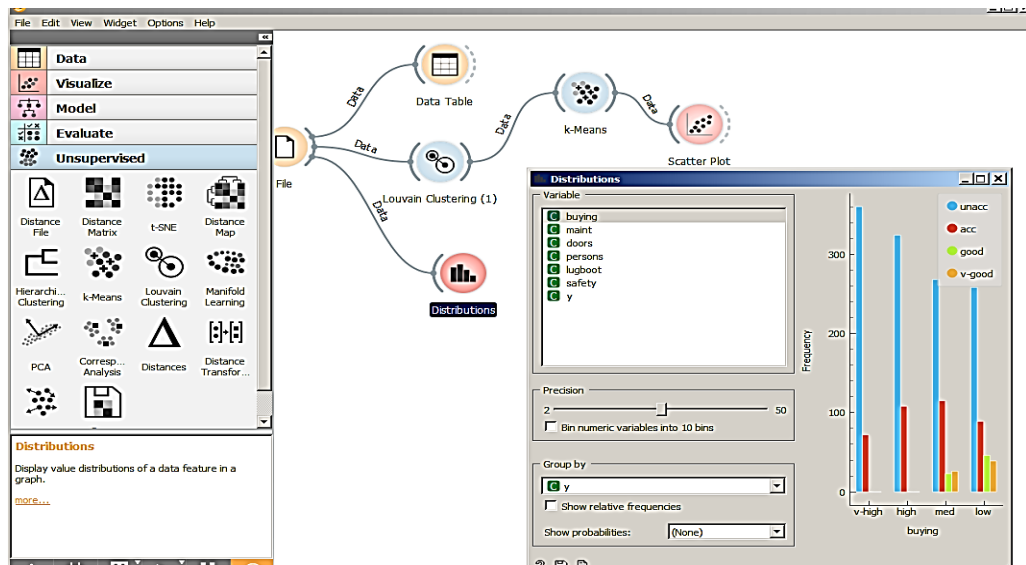


Рисунок 1.16 – Формування сценарію обробки даних на платформі Orange

Серед недоліків Orange, порівняно з іншими програмами інтелектуального аналізу даних, можна визначити те, що платформа має меншу кількість імплементованих алгоритмів класичної статистики.

4. **Weka 3.8.2** (Waikato Environment for Knowledge Analysis) – це платформа інтелектуального аналізу даних і машинного навчання, розроблена в Університеті Вайкато (Нова Зеландія) [166]. Основним інтерфейсом є Explorer. Інтерфейс Explorer містить панелі: імпорту даних з БД, текстових файлів у форматі CSV, опрацювання за допомогою алгоритмів (фільтрів). Ці фільтри використовуються для трансформування даних; класифікації та регресійного аналізу до обраного набору даних, візуалізації та оцінки результатів кластеризації, яка містить різноманітні методи кластеризації, наприклад, кластеризації методом k -середніх, EM-алгоритм тощо;

- вибору атрибутів, що найбільш впливає на якість прогнозування.

Як і в інших платформах інтелектуального аналізу даних, платформа Weka 3.8.2 (Рисунок 1.17) теж містить візуальний редактор для моделювання

процесів обробки даних за допомогою бібліотеки віджетів, представлених у вигляді .jar-файлів. Така перевага надає можливість імплементувати безпосередньо в аналітичний додаток програмний код процесу обробки даних. Алгоритми можна застосувати як, безпосередньо, до набору даних, так і викликати з коду Java.

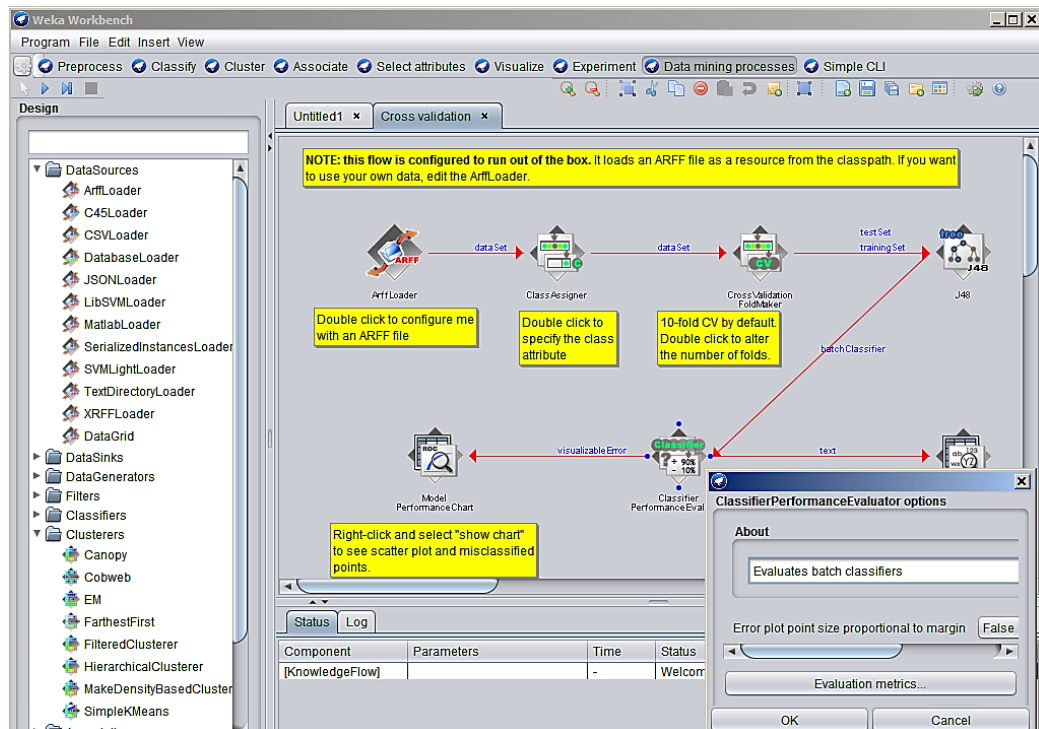


Рисунок 1.17 – Формування сценарію обробки даних на платформі Weka

Серед недоліків програмного продукту Weka можна відзначити, що алгоритми класифікації Weka 3.8.2 характеризуються меншим рівнем точності по відношенню до інших платформ.

1.5 Невирішені проблеми підвищення якості формування сценаріїв аналітичної діяльності

Однією із найбільших завад у напрямку підвищення ефективності АНС є занадто високий рівень складності технологічних компонентів, з яких вона складається, а також значні фінансові витрати на розробку, тестування, модифікацію та адміністрування. Крім того, значна шкода породжується від програмних помилок, що виявляються в процесі експлуатації АНС. Так, за оцінками національного інституту стандартів і технологій (National Institute of

Standards and Technology – NIST) щорічний збиток від дефектів програмного забезпечення для економіки США оцінюється у 59,6 млрд доларів, при цьому тільки третина цих дефектів може бути усунена шляхом застосування традиційних методів контролю якості програмного забезпечення, таких як тестування [49]. Як вважає більшість дослідників, на сьогодні проблема складності не тільки не вирішена, але й загострюється. Так Едгар Дійкстра пояснює це становище тим, що «більшість систем мають рівень складності, більший за припустимий, і відрізняються хаотичністю, невпорядкованістю, що призводить до збільшення скритих дефектів у програмних системах» [50]. Високий рівень складності призводить до виникнення ряду недоліків і вузьких місць, якими характеризуються існуючі архітектури програмних систем і підходи до проектування.

Це цілком стосується й спеціалізованих застосунків АНС, а саме: зростаючими вимогами до сценаріїв АнД, реалізація яких потребує обробки широкого спектра даних; різноманітністю методів та ІТ-обробки даних; потребою застосування інструментарію штучного інтелекту для підвищення якості аналізу. Також важливим фактором є взаємодія з чисельними компонентами АНС, які теж можуть мати скриті помилки або дефекти, що на жаль виявляються тільки при сумісному використанні компонентів АНС при вирішенні низки конкретних задач.

Отже і досі невирішеними проблеми у напрямку підвищення якості формування сценаріїв АнД можна вважати наступні.

Проектування АНС і сценаріїв АнД (як користувацького шару аналітичної системи) базується на фіксованому наборі вимог, які, особливо на початковому етапі, визначені нечітко. Тому за результатами такого проектування з'являються жорсткі сценарії АнД, які погано реагують на зміну вимог і зовнішніх чинників, а програмні застосунки, які їх реалізують, дорогі у підтримці та експлуатації, і в яких нерідко трапляються збої та аварії [51]. Зміна вимог вимагає перепроектування системи.

Застосування традиційної методології проектування АНС найчастіше виглядає як циклічне повторення таких робіт, а саме: Визначення вимог → Проектування архітектури → Придбання обладнання → Кодування програмного забезпечення → Тестування → Системна інтеграція → Впровадження та експлуатація. Реалізація подібного підходу зазвичай займає багато часу і не дозволяє сценаріям АнД гнучко реагувати на зміни технологій обробки даних, обсяги яких зростають у геометричній прогресії, а також вимог до методів інтелектуальної обробки даних і представлення результатів.

В існуючих АНС правила і алгоритми функціонування сценаріїв АнД жорстко зашиті в програмному коді аналітичних додатків. Будь-яка зміна алгоритму та правил вимагає реінжинірингу перекодування та тестування, що займає багато часу та потребує додаткового фінансування.

Під час проектування та керування системою превалює клаптиковий підхід – підсистеми та задачі розглядають окремо одна від іншої, у результаті чого вирішення однієї проблеми в одній підсистемі створює проблеми в інших підсистемах [53].

Керування системою (навіть прості рутинні задачі) виконується на низькому рівні за участю спеціалістів – адміністраторів, що дуже дорого та неефективно.

Знання про систему розрізнені, значна кількість їх недокументована та легко втрачається [54].

Відрив системи бізнес-процесів, які описують сценарії АнД, від програмних застосунків та ІТ (включаючи й Cloud), які забезпечують їхні реалізацію та функціонування. В результаті зміна у бізнес-системі не може бути оперативно відображена у програмному застосунку та, як наслідок, не змінює відповідній сценарій АнД.

Для вирішення зазначених проблем пропонуються різноманітні підходи та технології, які стосуються як архітектурних, так і методологічних принципів проектування *сценаріїв АнД*, що суттєво підвищить якість аналітики в цілому. Зокрема, з метою збільшення ефективності розробки сценаріїв АнД в умовах

змінності вимог як до спектра даних, що аналізуються, так і до їхньої кількості, запропоновано методологію семантичного проектування із застосуванням знань, що міститься в даних та ІТ як в явному, так і неявному вигляді, а також методів штучного інтелекту та машинного навчання.

1.6 Постановка задач дослідження

Як зазначалося раніше, найбільш важливою проблемою широкого впровадження аналітики у сучасному суспільстві є занадто високий рівень складності її компонентів, що призводить до її несприйняття більшістю рядових «аналітиків», на «плечі» яких саме й покладаються функції аналізу в організаціях і корпораціях. Це стосується як зайвої функціональності та специфічності існуючих аналітичних методів та операцій, інструментальних засобів пошуку, обробки та інтеграції різноформатних даних, так і високої вартості їхнього адміністрування, розробки та модифікації.

Отже, загальноприйнятим підходом до побудови програмних компонентів ефективних сценаріїв, які сприймаються аналітиками та придатні до модифікації, є аналіз та моделювання бізнес-процесів (включаючи багаторівневу взаємодію), які супроводжують галузь аналітики та виступають джерелом вимог до програмної системи. Для досягнення мети в роботі вирішуються наступні задачі.

1. Аналіз сучасного стану і особливостей проблеми дослідження, створення та вдосконалення комп'ютерних систем моделювання багаторівневих сценаріїв АнД, вибір та обґрунтування перспективних підходів до її вирішення з метою визначення та зменшення логічно зв'язаних помилок та семантичної незв'язаності опису багаторівневих сценаріїв чи їх окремих гілок, що призводить до суттєвого зменшення часу моделювання сценаріїв в цілому.

2. Розробка теоретичних засад встановлення співвідношення графічних елементів опису сценарію АнД з описом онтології його предметної області з метою забезпечення можливості проводити семантичний аналіз коректності як логічно-зв'язаних фрагментів сценарію АнД, так і його верифікацію.

3. Дослідження можливостей та розвиток математичних моделей і методів побудови сценарію АнД з урахуванням можливих наступних кроків сценарію та на основі виведення по прецедентах, що вже накопичені в результаті попереднього моделювання сценаріїв АнД.

4. Розробка та дослідження метода виконання сценаріїв АнД на основі Web-сервісів, вибір яких засновано на механізмі порівняння параметрів метаописів Web-сервісів із параметрами функціональних завдань сценарію.

5. Розробка алгоритмів та програмних засобів комп'ютерного моделювання сценаріїв на основі розроблених та удосконалених моделей (сценарних моделей, моделей опису предметної області в умовах невизначеності, заданих обмежень) та методів побудови та виконання сценаріїв АнД.

6. Створення комп'ютерної моделюючої системи (інтегрованого програмного інструментального середовища) для тестування і оцінки запропонованого теоретичного підходу комп'ютерного моделювання сценаріїв АнД, яка ґрунтується на побудованих моделях, методах та алгоритмах моделювання.

1.7 Висновки до першого розділу

1. Показано сутність аналітичної діяльності та її основна мета, що полягає в забезпеченні особи, яка приймає рішення, необхідними знаннями для прийняття правильного управлінського рішення в умовах непередбаченості та кризових явищ.

2. Надано основні визначення дослідження, проаналізовано багаторівневі моделі аналітичних систем і виділено вузькі місця як резерви для підвищення ефективності функціонування та проектування таких систем.

3. Проаналізовано підходи та методи моделювання та керування бізнес-процесами, які є джерелом вимог до формування сценаріїв аналітичної діяльності.

4. Доведено, що зростання функціональної складності програмних систем створює умови для розробки нових підходів щодо організації АнС,

оскільки традиційні підходи до проектування, розробки та адміністрування систем не відповідають вимогам сьогодення.

5. На підставі проведеного аналізу підходів, технологій і ПС моделювання сценаріїв аналізу даних визначено особливості зазначених ПС, які заважають здійснювати за їх використанням комп'ютерне моделювання сценаріїв Анд, особливо при зміні вимог та відповідно опису сценарію, а саме показано:

- в якості засобу опису сценарію процесів обробки даних відомі ПС використовують графічні нотації, що відповідає вимогам критерію функціональної оптимізації та дозволяє формувати підмножину вузлів обробки даних, необхідних для побудови сценаріїв;

- в зв'язку з цим побудова сценарію базується лише на візуальному представленні моделі процесів обробки даних, як наслідок, фіксованому наборі вимог, які, особливо на початковому етапі, визначаються нечітко, без формалізації та урахування семантичного наповнення процесів, що призводить до виникнення на етапі виконання сценаріїв логічно-зв'язаних помилок та необхідності повернення до початку процесу моделювання, що ускладнює процес моделювання та збільшує його час;

- правила логічно-зв'язаних переходів виконання сценаріїв Анд жорстко зашиті у програмному коді аналітичних програмних додатків, що не дозволяє використовувати ці сценарії при зміні цілей, початкових умов та обмежень Анд, тобто не враховується семантична складова Анд;

- ПС більш зосереджені на опису безпосередньо процесів обробки даних та не враховують їх взаємодію в межах багаторівневого сценарію, який виконується аналітиками з різними повноваженнями та компетенціями;

- ПС не використовують сучасні інтелектуальні методи, насамперед машинного навчання, в процесі виконання сценаріїв моделювання процесів обробки даних, тем самим не забезпечують можливості навчання ПС на власному досвіді та адаптування до незапрограмованих змін опису сценаріїв;

– в результаті зміна опису сценарію не може бути оперативно відображена у програмному додатку та, як наслідок, не дає можливості користувачеві, спираючись на свої знання чи на підказки ПС, які формуються з урахування вже накопиченого досвіду моделювання, скорегувати при необхідності сценарій в процесі здійснення аналітичної діяльності, що значно скоротило б час її здійснення.

До того ж, будь-яка зміна умов моделювання сценарію АнД вимагає реінжинірингу, перекодування та тестування програмних додатків моделювання, що займає багато часу та потребує додаткового фінансування.

6. З урахуванням цих особливостей з метою дослідження та розробки методів та засобів, які б забезпечували можливість адаптуватися до змін вимог до АнД, запропонований у відповідності до модельно-орієнтованого підходу до проектування ПС багаторівневий процес моделювання ПЗ моделювання сценаріїв АнД.

7. Виконано постановку задач дослідження.

РОЗДІЛ 2. ТЕОРИТИЧНІ ЗАСАДИ МОДЕЛЮВАННЯ СЦЕНАРІЇВ АНАЛІТИЧНОЇ ДІЯЛЬНОСТІ НА ОСНОВІ ОНТОЛОГІЧНИХ ТА ПРОЦЕСНИХ МОДЕЛЕЙ

2.1 Адаптація концепції процесного моделювання до побудови сценаріїв аналітичної діяльності

Дослідження багатьох проектів зі створення та впровадження складних АНС або ВІ-рішень показують, що традиційним підходом є або розробка спеціалізованого ПЗ, або придбання та адаптація готового ПЗ від відомих виробників [1, 166, 153]. Однак аналітичні додатки та ІТ-рішення можуть автоматизувати тільки певну частину процесів, що потрібні для функціонування організаційної системи в цілому. Крім того, традиційні методи проектування SADT, IDEF, UML, MDA, DSDM орієнтовані на використання чіткого набору статичних вимог [70, 128, 187, 188, 209]. Тобто у разі внесення навіть невеликих змін в сценарії АнД, виникає необхідність перепрограмування виконуваного коду, що призводить до значних інтелектуальних зусиль, витрат часу та залучення додаткових фінансів. В результаті ІТ-рішення все одно не встигають оновлюватися в тому темпі, який потрібен організаційній системі для реагування на виклики часу.

Шляхом вирішення проблем актуалізації сценаріїв АнД з мінімальними витратами на виклики часу є залучення підходів та технології, які притаманні BPM-системам (Business Process Management) з так званою «керованою моделлю» (model driven) [3, 9, 6]. Відмітна особливість процесно-орієнтованих систем з керованою моделлю полягає в тому, що розробка систем ведеться в термінах ПрО, а не комп'ютерного середовища (технологій) для їх виконання. Таким чином, розробник стає захищеним від складнощів безпосереднього програмування виконуваного коду і зосереджується на вдосконаленні моделі сценарію АнД (з позиції людського фактору), який є графічним описом послідовності робіт або станів об'єкту дослідження.

Суть залучення BPM-рішень для моделювання сценаріїв АнД (особливо багаторівневих) полягає в тому, що по аналогії з бізнес-процесами, сценарії АнД можуть бути представлені сукупністю процесних моделей (типових процесів), кожна з яких має початок і кінець та які взаємодіють між собою в межах загального сценарію. Для спрощення самого процесу формування сценаріїв залучаються графічні методології та відповідні засоби моделювання, зокрема, нотація BPMN (Business Process Model and Notation). Таке рішення дозволяє не тільки моделювати сценарії АнД у вигляді BPMN-діаграм, але й серіалізувати в XML-файл для відпрацювання в комп'ютерному середовищі із залученням стандартного ПЗ BPM-систем.

Але формальні методи графічного моделювання, незважаючи на їх безумовні переваги, мають суттєвий недолік – вони не пропонують механізмів змістовного (семантичного) опису та верифікації сценаріїв. Для подолання цього недоліку пропонується модель сценарію АнД, представлена в графічній нотації, доповнювати його онтологічною моделлю із залученням Web-технологій. Технологічно це означає, що BPMN-модель має бути конвертована в OWL-файл (у форматі XML), що включає такий набір сутностей онтології, який співвідноситься з усіма графічними елементами діаграми сценарію в нотації BPMN. Завдяки технології конвертації BPMN-моделі в онтологію, OWL-модель сценарію АнД, фактично, перетворюється в базу знань організації, яка може розширюватися за рахунок додавання нових сценаріїв АнД.

OWL-модель конвертованого сценарію може досліджуватися в інструментальному середовищі будь-якого зі стандартних редакторів онтології, зокрема платформою Protégé 5. Залучаючи Web-технології, розробник сценарію АнД має можливість проводити семантичний аналіз коректності логічно-зв'язаних фрагментів сценарію (особливо багаторівневих) використовуючи інструментарій редакторів онтології. Крім того, завдяки залученню OWL технологій, розробник також може формувати запити для аналізу логічно-зв'язаних фрагментів сценарію на природній мові. Для цього OWL технології пропонують низку сучасних мов запитів до бази знань – DL Query, SPARQL

Semantic Web Rule Language, які можуть оброблюватися стандартними машинами виводу – Reasoners.

Розроблений в нотації BPMN сценарій АнД та перевірений засобами OWL в онтології з мінімальним програмуванням серіалізується в формат BPEL (Business Process Execution Language), який виконується в середовищі BPM-систем. Якщо в подальшому виникає необхідність внесення змін в логіку роботи АС, модифікації піддається тільки модель сценарію, яка автоматично серіалізується в BPEL.

Таким чином, запропонований підхід до побудови моделі сценарію на базі «керованої моделі», яка визначає логіку роботи АНС, дає низку переваг – починаючи з суттєвого спрощення самого процесу моделювання та валідації сценаріїв АнД в інтегрованому середовищі BPMN–OWL і, закінчуючи, реалізацією моделі в середовищі стандартних BPM-систем.

Це також створює умови для інтелектуалізації безпосередньо процесу моделювання та верифікації сценаріїв АнД за рахунок поєднання переваг графічної нотації з описом знань у вигляді OWL-моделей.

2.1.1 Мови та нотації моделювання процесів

Концепція процесного управління організацією BPM (Business Process Management) переживає сьогодні друге народження. В цілісному вигляді BPM була сформульована в 2003 році в роботі Г. Сміта і П. Файнгар «Управління бізнес-процесами (BPM): третя хвиля». Ця концепція поєднує процесну методологію, яка, по-перше, зосереджується на операціях, стратегії, вимогах клієнта, проблемах крос-функціональної взаємодії, а, по-друге, спирається на сучасні ІТ, які підтримують графічне моделювання, процесний движок, функціонал моніторингу і аналітики.

Відомо, що діаграми передають інформацію значне точніше та краще сприймаються людиною ніж звичайна мова [12]. Це пояснюється не тільки мовними особливостями представлення та передачі інформації, але й когнітивними можливостями людини щодо сприйняття інформації та отримання нових знань. Відповідно до теорії подвійного каналу [107] людський

розум має окремі системи для обробки образотворчого і текстового матеріалу. Візуальний матеріал (графічні зображення, включаючи й діаграми) обробляються паралельно візуальної системою людини, а текстовий матеріал обробляються послідовно мовною системою людини. Тільки схематичне подання інформації (у вигляді діаграм) може передавати складні відносини Про відразу, використовуючи графічне та текстове представлення матеріалу одночасно. Відомий дослідник Moody у своїй роботі «The 'physics' of notations: Toward a scientific basis for constructing visual notations in software engineering» дуже чітко окреслив можливості та переваги візуальної (графічної) нотації:

«...візуальна нотація (або візуальна мова, графічна нотація) складається з набору графічних символів (візуального словнику), набору композиційних правил (візуальної граматики) і визначення значення кожного символу (візуальної семантики)».

Візуальний словник і візуальна граматика разом утворюють візуальний синтаксис. Графічні символи використовуються для символізації (перцептивного уявлення – тобто здатності людини до сприйняття матеріалу) семантичних конструкцій, які, зазвичай, визначаються як мета-модель. Значення графічних символів визначається шляхом зіставлення їх з конструкціями, які вони представляють. Допустимий вираз в візуальній нотації називається візуальною пропозицією або діаграмою. Діаграми складаються з екземплярів символів (токенів), розташованих відповідно до правил візуальної граматики.

Найбільш відомими графічними мовами для створення мета-моделей вважаються Event Process driven Chain (EPC) та Business Process Modelling Notation (BPMN). У порівнянні цих графічних засобів, нотація BPMN пропонує більш детальне уявлення і логіку виконання для елементів мета-моделі, ніж EPC. Слід підкреслити, що подання інформації про процеси в графічній формі ще не гарантує, що така форма цілком сприймається розробником ПЗ та коректно описує процеси Про [154]. Оскільки діаграми UML не завжди передають всі особливості застосування графічних конструкцій в кожному

окремому проєкті, що в цілому ускладнює сприйняття та семантичну інтерпретацію UML-діаграм розробником ПЗ. Те ж саме можна сказати і про інші відомі мови (нотації) графічного моделювання бізнес-процесів ARIS, IDEF0, IDEF3.

На базі концепції BPM з'явився новий клас програмного забезпечення – BPMS – «Business Process Management Software» або «Business Process Management Suite» [155]. Сьогодні системи класу BPMS пропонують відомі компанії, такі як IBM, Oracle, SAP, Software AG, і десятки вузьких фахівців (pure-play vendors).

Розвитком концепції BPM та BPMS є поява графічної нотації BPMN (Business Process Model and Notation), яка з 2013 року прийнята як стандарт моделювання ISO/IEC 19510 у галузі інформаційних технологій. Основна цінність цієї нотації в тому, що вона дозволяє вирішувати не тільки традиційні задачі моделювання для регламентації та оптимізації як нотації EPC або UML, але й дозволяє безпосередньо виконувати процеси. BPMN – підтримується мовами BPMML і BPEL та дозволяє не тільки будувати діаграми процесів в графічній формі, але й перетворювати їх у виконувані XML-файли, які в подальшому реалізуються в комп'ютерному середовищі BPMS-IT системи підприємства.

Стандарт увібрав в себе всі кращі ідеї, концепції і досвід інших нотацій моделювання процесів, таких як ML Activity Diagram, UML EDOC Business Processes, IDEF, ebXML BPSS, Activity-Decision Flow (ADF) Diagram, RosettaNet, LOVeM, and Event-Process Chains.

Концептуально нотація BPMN 2.0 фактично визначає специфікації для мета-моделі, яка описує процеси Про. Тобто мета-модель містить формальну специфікацію примітивів нотації (графічних елементів), а також взаємозв'язки між елементами, а її компоненти визначаються як класи об'єктів з певними атрибутами. Деякі класи (Root Element, Base Element) є чисто абстрактними і не використовуються безпосередньо в моделях BPMN.

Стандарт BPMN використовується для опису процесів нижнього рівня та розглядає три основних типи підмоделей в рамках наскрізній моделі BPMN:

1. Процес (Оркестровка – Orchestration) розділяється на:
 - приватні (внутрішні) бізнес-процеси, що не виконуються;
 - приватні (внутрішні) бізнес-процеси, що виконуються;
 - публічні процеси.
2. Хореографія (Choreographies)

Оркестровка – це опис потоку взаємодії між внутрішніми і зовнішніми елементами бізнес-процесу із залученням веб-сервісів для їх виконання [260]. При оркестровці вважається, що існує центральний процес (як частина від бізнес-процесу вищого рівня), який управляє викликаними веб-служб (веб-сервісів – елементарних операції). Оркестровка відрізняється явними описами операцій і порядком виклику веб-сервісів.

Хореографія – це визначення послідовності умов, при дотриманні яких кілька незалежних учасників обмінюються повідомленнями з метою виконання загального бізнес-завдання [260]. В хореографії відсутня роль центрального координатора, оскільки кожна веб-служба, залучена в хореографію, знає точно, коли і з ким потрібно взаємодіяти. Кожен учасник хореографії повинен знати про всі виконуваних бізнес-процеси, операції, повідомлення та налаштування для їх обміну.

Межі між оркестровкою і хореографією досить розмиті. Так, мова BPEL підтримує і оркестровку, і хореографію через поняття абстрактних і виконуваних бізнес-процесів, хоча безпосередньо для опису хореографії призначений, наприклад, WS-CDL. Нотація BPMN дозволяє почати з розробки аналітичної моделі, яка дає загальне уявлення про характер виконання бізнес-процесу. У міру зростання розуміння того, як повинен виконуватися бізнес-процес, модель розширюється, уточнюється і поглиблюється. Результатом моделювання стає виконувана модель бізнес-процесу.

Мова моделювання бізнес-процесів (Business Process Modeling Language – BPMML) – це мета-мова на основі XML, призначена для моделювання бізнес-

процесів, які описують процесні аспекти діяльності корпоративних систем [BPM]. Ініціатива в створенні мови BPMML належить корпорації BPMI (Business Process Management Initiative). З червня 2005 р. BPMI спільно з OMG (Object Management Group) сформували робочу групу з моделювання та інтеграції бізнесу (BPMI DTF). [262]. OMG, яка є фундатором мови UML, застосувала мову BPMML для розширення виразності нотації UML. Мова BPMML визначає операції різного рівня складності, транзакції і компенсації, управління даними, паралелізм, обробку винятків і операційну семантику. Граматика мови BPMML оформлюється у вигляді XML-схеми, що забезпечує сталість визначень і їх обмін між різнорідними системами та інструментами моделювання – Frameworks.

Мова BPMML доповнює мову реалізації бізнес-процесів BPEL (Business Process Execution Language), яка відома також як Web Services Business Process Execution Language або xlang версія MS [265]. Тобто BPEL – це XML-мова виконання бізнес-процесів. Мова BPMML використовується для визначення детальних процесів, які виконуються при виклику кожного web-сервісу (взаємодія яких описується мовними конструкціями BPEL).

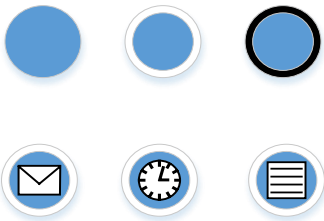
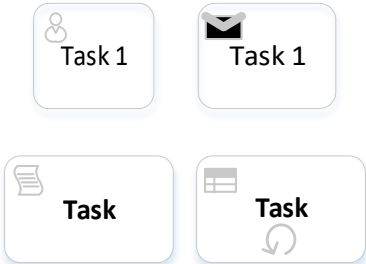
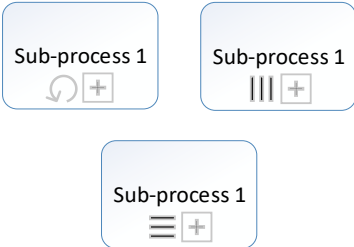
XPDL (XML Process Definition Language) – мова та формат обміну даними між BPM-системами. XPDL служить для опису інтерфейсів Web-служб та використовується для моделювання доступних операцій, включаючи адреси їх виклику. Формат обміну даними XPDL запропонований як стандарт для імпорту/експорту описів бізнес-процесів у вертикально інтегрованих інструментальних платформах типу BPwin, ARIS, Oracle Hyperion, Bizagi тощо.


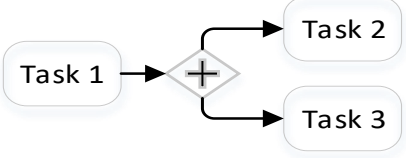
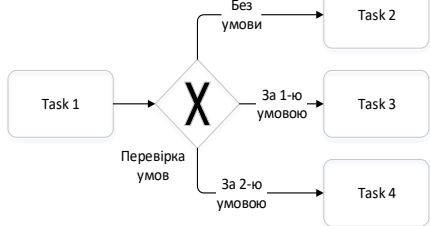
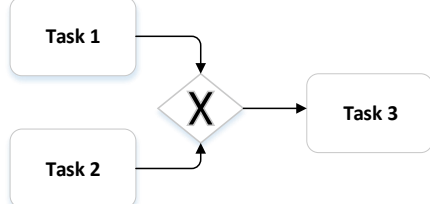
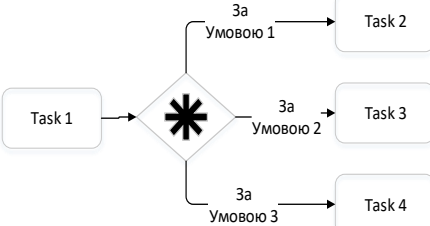
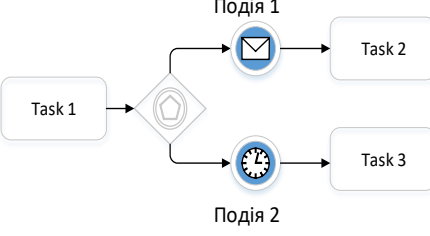

BPMIS-IT – система побудована за певними правилами, заданими в методології BPM та являє собою інтегрований набір програмного забезпечення, який визначає архітектуру додатків та технологію інфраструктури для роботи та виконання програм в нотації BPMN 2.0. У процесі виконання модель бізнес-процесу в нотації BPMN 2.0 транслюється в опис процесу на мові BPEL, який потім завантажується в движок BPMIS-IT-системи.





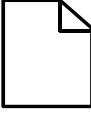

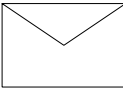

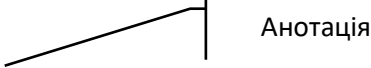
2.1.2 Базові положення нотації BPMN 2.0

Нотація BPMN 2.0 використовує базовий набір інтуїтивно зрозумілих елементів (специфікацію), які дозволяють визначати складні семантичні конструкції моделі процесу у графічному вигляді. Крім того, специфікація BPMN 2.0 визначає, яким чином діаграми, що описують процес, можуть бути трансформовані в виконувани моделі на мові BPEL (Business Process Execution Language). BPEL є стандартною виконуваною мовою OASIS для визначення дій в бізнес-процесах з веб-службами. Основні елементи нотації BPMN 2.0, позначення та пояснення їх сенсу наведені в таблиці 2.1

Таблиця 2.1

Назва елементу нотації	Опис та пояснення сенсу елементу нотації BPMN	Графічне позначення
Event – Подія	<p>Event – Подія, що «відбувається» під час процесу. Існує три типи подій: Початок, Проміжний і Кінець.</p> <p>Події впливають на стан виконання процесу, та визначають умови (причину) виконання через механізм тригерів або впливів на результат (таймер, помилка, компенсація, умови тощо)</p>	
Activity – Діяльність	<p>Activity – характеризує різні види Діяльності, що виконуються в моделі Діяльність поділяються на:</p> <p>Послідовність елементарних дії;</p> <p>Завдання – (задача – Task) елементарна дія (або операція), яка не має подальшої декомпозиції в рамках даного процесу. В BPMN задачі розрізняються за типом (Task Type) та позначаються відповідним знаком: сервіс, прийняти, надіслати, встановити роль та інші</p>	
	<p>Підпроцес – (Sub-process) більш детально описаний процес, який на діаграмі включено до складу основного процесу. На діаграмі підпроцес позначається блоком зі знаком "плюс" в центрі нижньої частини фігури, які в свою чергу підпроцеси розрізняють на <i>стандартні</i>, <i>паралельні</i> та <i>послідовні</i></p>	

Назва елементу нотації	Опис та пояснення сенсу елементу нотації BPMN	Графічне позначення
Gateway – Шлюз (логічні оператори)	Gateway – шлюз вузол (логічний оператор) для позначення розгалуження процесу при виникненні певних умов:	
	<ul style="list-style-type: none"> Логічний оператор AND - "+" (Паралельний шлюз) використовується для позначення злиття / розгалуження потоків управління в рамках процесу. Задаються за допомогою умовних потоків управління, що виходять з шлюзу. Після виконання Task 1 запусниться виконання і Task 2 і Task 3. 	
	<ul style="list-style-type: none"> Логічний оператор XOR – диз'юнкція. Ексклюзивний шлюз використовується для розгалуження потоку на кілька альтернативних при настанні умов: наприклад, якщо Умова 1 вірно – виконується Task 3; якщо Умова 2 – Task 4; якщо ні Умова 1, ні Умова 2 – Task 2. 	
	<ul style="list-style-type: none"> Логічний оператор OR – диз'юнкція. Ексклюзивний шлюз OR використовується також для злиття потоків управління. В цьому випадку шлюз просто пропускає через себе всі потоки управління без синхронізації. 	
	<ul style="list-style-type: none"> Комплексний шлюз використовується для розгалуження потоку управління на кілька потоків, коли виконання процесу залежить від виконання умов. Наприклад, якщо Умова 1 - виконується Task 2, якщо Умова 2 – виконується Task 3, якщо Умова 3 – Task 4 	
<ul style="list-style-type: none"> Ексклюзивний шлюз за подіями використовується для розгалуження потоку управління на кілька альтернативних потоків, коли подальше виконання процесу залежить від виникнення деякої події-умови наступного після шлюзу. 		
Sequence flow – Потік управління	Sequence flow – Потік управління. Стрілка показує зв'язок елементів потоку BPMN (подій, процесів, шлюзів) та відображає хід виконання процесу. При необхідності потік може бути іменованим.	

Назва елементу нотації	Опис та пояснення сенсу елементу нотації BPMN	Графічне позначення
Message Flow – потік повідомлення	Message Flow – повідомлення, якими обмінюються учасники процесу. Потік показує передачу повідомлень або об'єктів з одного процесу в інший	
Association – Асоціація	Association – використовується для зв'язку інформації та артефактів з графічними елементами BPMN 2.0	
Pool – Пул	Pool – це графічний елемент, який описує один процес на діаграмі. На одній діаграмі може бути кілька пулів. Pool може містити внутрішні деталі процесу. Зазвичай з пулом асоціюється Роль.	
Lane – Доріжка	Доріжка – розділяє процесі всередині пулу та поширюється по всій довжині Процесу вертикально або горизонтально. Доріжки використовуються для категоризації заходів та позначають виконавців процесів, що прописуються всередині доріжки	
Data Object – Об'єкт даних	Об'єкт даних - надають інформацію про те, які дані потрібні для виконання певних дії Об'єкти даних можуть представляти. особливий об'єкт або колекцію об'єктів.	
Data Store – База даних	Використовується для відображення на діаграмі бази даних, що забезпечує виконання процесів	
Message – повідомлення	Message – використовується для відображення вмісту спілкування між двома Учасниками процесу	
Group – Група	Група – сукупність графічних елементів, віднесених до однієї категорії. Назва категорії відображається на діаграмі як мітка групи. Використовується для цілей підготовки документів або аналізу	
Text Annotation – Текстові анотації	Текстові анотації – механізм надання додаткової текстової інформації для пояснення діаграм в нотації BPMN.	

До числа переваг застосування нотації BPMN 2.0 відносяться:

- Можливість максимально деталізувати події, дії, задачі, логічні оператори, підпроцеси, умови виникнення подій, тригери, а також дані та коментарі, необхідні для отримання результату;
- Графічні нотації – наочні, що дозволяє зрозуміти фахівцям та керівникам особливості процесів і побачити їх слабкі місця;
- Специфікація BPMN 2.0 є виконуваною, тобто процес сформований в одному редакторі від одного виробника, може бути виконаний на движку (Engine) абсолютно іншого виробника, за умови, якщо BPMS-IT – система підприємства підтримує нотацію BPMN 2.0.

Незважаючи на свою промислову зрілість, нотація BPMN 2.0 не підтримує уявлення деяких специфікацій процесу, таких як ресурси, необхідні для виконання даного завдання, або робочої станція, на якій повинна бути виконана дана задача. Ці специфікації необхідні для повного опису процесу, як зазначено в (Falbo і Bertollo, 2009): «Процес повинен бути визначений з урахуванням: дій, які необхідно виконати, необхідних ресурсів, артефактів введення і виведення, прийнятих процедур (методів), шаблонів тощо) і моделі життєвого циклу, яка буде використовуватися».

2.1.3 Методологія побудови сценаріїв аналітичної діяльності на основі OWL та нотації BPMN

Моделювання сценаріїв АнД ґрунтується на розумінні сценарію, як сукупності логічного, послідовного, взаємозалежного набору подій, та аналітичних процедур, над даними, IP та знаннями з метою отримання кінцевого результату, що відповідає висунутим вимогам. Під методологією (нотацією) створення моделі сценарію АнД, розуміється сукупність методів та ІТ, за допомогою яких об'єкти та суб'єкти АнД, а також зв'язки між ними представляються у вигляді моделі, яка сприймається людиною, та яку можна досліджувати у різних вимірах [247].

На сьогодні спектр методів моделювання досить широкий: від найпростіших графічних нотацій, що використовуються для побудови блок-схем та алгоритмів, і таких математичних апаратів, як мережі Петрі, до об'єктно-орієнтованих мов моделювання UML (Unified Modeling Language) та спеціально розроблених мов моделювання бізнес-систем, наприклад, XPDL

(XML Process Definition Language) і BPEL (Business Process Execution Language). За критерієм вартість/ефективність найбільш привабливою для побудови сценаріїв Анд є нотація BPMN 2.0., яка увібрала в себе кращі ідеї [268], що реалізовані в таких відомих нотаціях і методологіях моделювання як:

1. IDEF (SADT) structured analysis and design technique – [69, 106]: «...методологія SADT являє собою сукупність методів, правил і процедур, призначених для побудови функціональної моделі об'єкта будь-якої Про». Функціональна модель SADT відображає структуру об'єкта (процесу), тобто вироблені їм дії та зв'язки між цими діями

2. UML 20 (Unified Modeling Language) – що дотримується парадигми об'єктно-орієнтованого програмування, є відкритим стандартом, що використовує графічні позначення для створення абстрактної моделі системи:

- Activity Diagram – візуальне представлення діаграм діяльності;
- Interaction Overview Diagram – візуальне представлення діаграм Огляду взаємодії;
- EDOC (Enterprise Distributed Object Computing, корпоративна розподілена обробка об'єктів) – Business Processes;

3. ebXML (Electronic Business eXtensible Markup Language на основі стандартів OASIS) – забезпечує відкритий протокол, заснований на XML-інфраструктурі для здійснення електронного бізнесу;

4. ebXML BPSS (Business Process Specification Schema) – схеми специфікацій бізнес-процесів).

5. ADF Diagram (Activity-Decision Flow) – потік «діяльність-результат», якій використовуються для візуалізації послідовності дій та умов для координації поведінки нижчого рівня при опису процесів.

6. RosettaNet – стандарт, що визначає як документи електронної комерції і протоколи обміну функціонують в просторі електронного обміну даними EDI (Electronic data interchange). Стандарт RosettaNet заснований на XML і визначає рекомендації щодо повідомлень, інтерфейси для бізнес-процесів та протоколи взаємодій між компаніями.

7. LOVeM (Line of Visibility Engineering Methodology) - відома як "Методика інженерії видимості IBM" – візуальна методологія проектування, що використовує базові позначення блок-схеми з горизонтальними смугами swim lanes. Верхня смуга завжди використовується для представлення замовника або кінцевого споживача.

8. EPC (Event-Driven Process Chain) - подієвий ланцюжок процесів, являє собою спрямований граф з подій і функцій, а також початкових та кінцевих подій, використовуючи логічні оператори OR, AND, XOR. EPC розроблений в рамках архітектури інтегрованих інформаційних систем ARIS (Architecture of Integrated Information Systems).

Отже, методологія проектування моделей сценаріїв АнД в нотації BPMN ґрунтуються на наступних положеннях:

1. *Line of Visibility Engineering Methodology* – візуальній методології проектування (графічна мова), здатний наочно представляти широкий спектр ділових, виробничих та інших процесів і операцій підприємства на будь-якому рівні деталізації.

2. Представленні моделі у вигляді спрямованого графу (діаграми), в якому діаграма відображає функціональну взаємодію графічних елементів BPMN. Взаємодія елементів описуються спрямованими ребрами, а умови переходів за допомогою шлюзів, що виражають "обмеження", які визначають, коли і яким чином функції виконуються й управляються.

3. Представлені сценарію як сукупності взаємопов'язаних і взаємодіючих елементів, спрямованих на досягнення кінцевого результату. Елементами сценарію можуть бути будь-які комбінації різноманітних сутностей (завдання, процеси, підпроцеси, виконавці, ролі, інформація, дані, ПЗ, логічні оператори, умови логічних переходів тощо). Сценарії описує: 1) що має відбуватися в системі (організації) для досягнення визначеної мети; 2) які управляючи дії потрібні для її досягнення; 3) які сутності він перетворює; (4) які ресурси використовує для виконання функцій та завдань, (5) що є кінцевим результатом виконання сценарію.

4. Об'єднанні в BPMN різних типів моделювання процесів: внутрішніх (Private Process) для опису потоку операцій в межах окремого процесу; публічних (Public Process) для відображення взаємодії між приватними

процесами; процесах взаємодії між двома або більше учасниками (акторами) (Choreography). Це дозволяє створювати безперервні (end-to-end) процеси.

5. Точності виконання правил опису процесів, графічних елементів та їх взаємодії, без внесення в той же час надмірних обмежень на дії аналітика.

6. Підтримці концепції ітеративного моделювання, де розробка моделі процесу являє собою ітеративну процедуру. На кожному кроці ітерації аналітик пропонує варіант моделі процесу, який піддають обговоренню і подальшому редагуванню, після чого цикл повторюється.

7. Серіалізації візуальних моделей сценаріїв АнД у форматі мов XML та BPMML для їх подальшої комп'ютерної обробки та виконання в BPMS-IT-системах.

В нотації BPMN 2.0 сценарій АнД відображає хід процесу у вигляді послідовності окремих логічно-пов'язаних завдань, підпроцесів, початкових, проміжних та кінцевих подій, логічних операторів, а також умов які їх викликають.

Іншою базовою концепцією побудови сценаріїв АнД є методологія семантичного моделювання, заснована на знаннях. Ця методологія застосовує відкриту модель подання знань – Open Knowledge Base Connectivity (OKBC) [162] та мову Web Ontology Language (OWL). За рівнем виразності (повноти опису онтології) мова OWL2 вважається як така, що володіє найбільшою семантичною виразністю та інтегрує можливості RDFS та мовні концепції, засновані на дескрипційній логіці, що дозволяє формування складних запитів до моделі знань. OKBC дуже чітко визначає процес формування електронної моделі знань, якій складається з таких глобальних етапів:

- створення загальної термінології Про (таксономії), для спільного використання і розуміння всіма користувачами та розробниками АнС;
- формування точного і несуперечливого визначення значення кожного терміну таксономії;

– завдання семантики для кожного поняття ПрО (терміну таксономії) за допомогою аксіом, які автоматично дозволяють отримувати відповіді на основі стандартних мов запитів DL Query, SPARQL Semantic Web Rule Language та машин виводу – Reasoners.

Мова OWL2 містить такі елементи як: класи (Classes), властивості (Properties) і екземпляри класів (Individuals) [6]. Поняття ПрО діляться на класи, підкласи, екземпляри (примірники). Класи описуються тегом, наприклад,

```
<owl:Class rdf:about="http://untitled-ontology-9#Task">
```

Властивості в мові OWL2 поділяються на два види: DatatypeProperty (характеристики) і ObjectProperty (зв'язки або відношення). DatatypeProperty описуються тегом вигляду

```
<owl:DatatypeProperty rdf:about="http://untitled-ontology-9#Name">
```

і призначені для опису характеристик класів. Як значення властивостей-характеристик виступають дані, що характеризують описуваний клас.

ObjectProperty – це відносини, які можуть поєднувати між собою класи або екземпляри класів. ObjectProperty описуються тегом вигляду

```
<owl:ObjectProperty rdf:about="http://untitled-ontology-9#Take_task">
```

Екземпляри описуються фреймом і можуть бути представленні як класом, так і підкласом, що дає можливість описувати складні предметні області.

Отже, методологія семантичного моделювання із застосуванням уніфікованої моделі ОКВС та з використанням мови OWL2 складається з наступних етапів [93]:

1. Визначення концептів, тобто базових понять – класів, сутностей, категорії (<Active Ontology>, <Entities>, <Classes>), які характеризують ПрО.
2. Визначення множини властивостей, що описують концепти ПрО із застосуванням механізмів визначення атрибутів та ролей (<Data Properties>).

3. Встановлення відносин між концептами ПрО та їх властивостями із застосуванням механізмів формування предикатів (<Object Properties>) типу «вирішує задачу», або «входить до складу» та інші.

4. Створення екземплярів окремих класів онтології із застосуванням механізму (<Individual by class>) для наповнення бази знань ПрО.

5. Встановлення числових або логічних обмежень (<Valiu>) при опису властивостей екземплярів (<Individuals>) для формування аксіом (<Axiom>) ПрО. Наприклад, значення максимальної швидкості для наземних об'єктів обмежується певною величиною.

6. Сериалізації моделей онтології у форматі мови XML, яка сумісна з описом моделі процесу в нотації BPMN для подальшої комп'ютерної обробки в BPMS-IT-системі.

7. Розробка типових шаблонів запитів до бази знань з використанням різних мов запитів DL Query, SPARQL Semantic Web Rule Language та машин виводу – Reasoners.

8. Перевірка коректності функціонування онтологічної моделі ПрО з точки зору її відповідності вихідним цілям і завдання та знаходження розривів в онтології. Оцінка здійснюється на основі аналізу результатів тестування різними машинами виводу (Reasoners) та складання різноманітних типів запитів.

9. Ітеративна та поетапна розробка онтології з можливістю вдосконалення онтологічної моделі ПрО та проведення відповідних доробок.

Враховуючи викладене, запропонована методологія побудови моделі сценарію на базі «керованої моделі», які притаманні BPM-системам, та семантичного моделювання із застосуванням уніфікованої моделі ОКВС. Така методологія дозволяє не тільки моделювати сценарії АнД у вигляді BPMN-діаграм (XML-схеми), але й серіалізувати його у вигляді OWL-файлу для відпрацювання в комп'ютерному середовищі із залученням мов запитів до бази знань – DL Query, SPARQL Semantic Web Rule Language та машин виводу – Reasoners. У той час як XML-схеми через свою технічну специфікацію

можуть виконувати тільки обмежену перевірку XML-елементів, онтологія забезпечує більш високий ступінь деталізації завдяки своїй структурі.

Методологія суттєво спрощує сам процес моделювання за рахунок більшої наочності сприйняття моделі та дає низку переваг – починаючи з суттєвого спрощення самого процесу моделювання та валідації сценаріїв АнД в інтегрованому середовищі BPMN–OWL і, закінчуючи, реалізацією моделі в середовищі стандартних BPM-систем.

2.2 Теоретичні засади семантичного моделювання сценаріїв аналітичної діяльності

Застосування тільки графічних методів моделювання сценаріїв (навіть за стандартом ISO/IEC 19510 [261]) не виключає появи семантичних помилок, пов'язаних з моделюванням складних зав'язків між учасниками сценарію. Для усунення такої невідповідності пропонується інтегрована технологія моделювання сценаріїв АнД, заснована на графічних методах процесного моделювання (нотація BPMN) та онтологічному (семантичному) аналізі сценаріїв, сформованих в нотації BPMN.

Для здійснення процесного моделювання сценаріїв вживаються певні категорії понять нотації BPMN, як дозволяють доволі повно визначити сценарії АнД через її графічні елементи. До таких категорій відносяться [264]:

- Категорія «об'єкти потоку» (flow objects) – містить такі графічні елементи як подія (Event), завдання (Task), підпроцес (Sub-Process), шлюз (Gateway);
- Категорія «з'єднуючі об'єкти» (connecting objects) - потік послідовності (Sequence Flow), потік повідомлень (Message Flow), асоціація (Association);
- Категорія «доріжки» (swimlanes) – пул (Pool), смуга (Lane);
- Категорія «артефакти» (artifacts) - об'єкт даних (Data Object), текстова анотація (Text Annotation), група (Group).

Кінцевим результатом моделювання сценарію АнД є діаграма в нотації BPMN – графічна модель сценарію, яка може бути серіалізована в XPDL-файл

(XML Process Definition Language). У математичному сенсі діаграма сценарію – це спрямований граф, який складається з елементів нотації BPMN та описується кортежем [270]

$$Gr(Bp_N) = \{ V(Bp_N), E(Gr), L(v), Id(v), Rul^{(Bp)}, p(v) \},$$

Де $\{V(Bp_N)\}$ – множина вузлів спрямованого графа $Gr(Bp_N)$, причому множина вузлів графа має вигляд $V(Bp_N) = \{St \cup Tk \cup In \cup Gt \cup Fn\}$, де

$\{St\}$ – підмножина вершин, що представляють собою стартові події (Start Event) в нотації BPMN;

$\{Tk\}$ – підмножина вершин, що представляють задачі (Task, Sub-Process) в нотації BPMN;

$\{In\}$ – підмножина вершин, що представляють собою проміжні події процесу (Intermediate events);

$\{Gt\}$ – підмножина вершин, що представляють собою шлюзи (Gateway) – логічні оператори;

$\{Fn\}$ – підмножина вершин, що представляють кінцеві події (End Event);

$\{E(Gr)\}$ – множина ребер, які зв'язують вершини в напрямлені ребра $(v_i, v_{i+1}) \in E(Gr)$ спрямованого графа. Зв'язування вершин здійснюється через категорія з'єднуючи об'єкти (connecting objects). Де матриця суміжності ребер графа має вигляд $E(Gr) \subseteq |V(Bp_N) \times V(Bp_N)|$ для n вершин графа;

$\{L(v)\}$ – множина міток вершин, які дозволяють однозначно визначати кожен вузол $V(Bp_N)_i$ в спрямованому графі $Gr(Bp_N)$;

$\{Id(v)\}$ – функція розмітки вершин в спрямованому графі для надання унікальних імен, яка має вигляд $Id(v): V(Bp_N) \rightarrow L(v)$;

$\{Rul^{(Bp)}\}$ – множина правил зв'язування графічних елементів нотації, які враховують семантичні обмеження та умови розгалуження потоків в шлюзах моделі;

$p(v): E(Gr) \rightarrow [0,1]$ – булева функція – зіставляє значення ймовірності ребрам, які зв'язують вершини спрямованого графа.

Для виконання аналізу і подальшого вдосконалення моделі сценарію в нотації BPMN, спрямований граф $Gr(Bp_N)$ має бути представлений як в форматі XML (для взаємодії з OWL редакторами), так і мовою XPDЛ для обміну моделями процесів в середовищі виконання BPM-систем.

Відмітна особливість процесно-орієнтованих BPM-систем з керованої моделлю полягає в тому, що на відміну від об'єктно-орієнтованого підходу, побудова сценарію АнД ведеться в термінах ПрО, а не комп'ютерного середовища та технологій для їх виконання.

Але будь-які формальні методи, включаючи й методи процесного моделювання, незважаючи на їх безумовні переваги у сенсі формалізації, мають суттєвий недолік – вони не пропонують механізмів змістовного (семантичного) опису та верифікації розгалужених сценаріїв. Для подолання цього недоліку з метою формування коректної моделі складного процесу моделювання сценаріїв АнД запропоновано доповнювати концептуальну модель сценарію АнД, представлену в графічній нотації, його концептуальною онтологічною моделлю, яка включає категорії ПрО, які співвідносяться з основними категоріями нотації BPMN – це підмножини класів, відношень, аксіом, правила логічного висновку, множина екземплярів.

Для реалізації зазначеного пропонується розроблений метод перетворення BPMN-моделі в OWL-модель сценарію, яка додає семантичне пояснення та інтерпретацію аналітичних процесів. Суть метода в тому, що на першому етапі BPMN-модель має бути конвертована в OWL-файл (у форматі XML), якій включає такий набір сутностей онтології, що співвідноситься з усіма графічними елементами діаграми сценарію в нотації BPMN. Завдяки технології конвертації BPMN-моделі в онтологію, OWL-модель сценарію АнД, фактично, перетворюється в базу знань, яка може розширюватися за рахунок додавання нових сценаріїв АнД.

Слід нагадати, що у наслідок суб'єктивного сприйняття аналітиком специфікацій ПрО, завжди виникають помилки моделювання процесів. Ці помилки пов'язані як з компетентністю аналітика, так і з недоліками суто

формального опису процесів, які притаманні безпосередньо **нотації** BPMN [261]. Автори, які проводили дослідження типових помилок, що допускають аналітики при проектуванні процесів у нотації BPMN визначають шість типів помилок [265]. До найбільш вагомих можна віднести такі помилки, пов'язані з семантичною неоднозначністю елементів нотації BPMN:

- недостатньо повно визначено умови настання кінцевих подій (End Event) F_n окремих підпроцесів складного сценарію;
- недостатньо чітко визначені умови настання проміжних подій сценарію, що призводить до виключення подій з потоку (Process Flow) або неправильного виконання процесу;
- недостатньо пов'язані завдання або підпроцеси сценарію - оскільки не враховані особливості різних об'єктів потоку і правила їх з'єднання;
- нечітко визначені складні (багатофакторні) умови поділу потоку шлюзами (Gateway) на кілька маршрутів. Оскільки в нотації BPMN ребра вершини графа описуються булеві змінними $p(v): E(Gr) \rightarrow [0,1]$, це суттєво ускладнює коректне формування багатофакторних умов із залученням механізму тригерів подій, що призводить до виключення подій з потоку (Process Flow) або неправильного виконання процесу.

Отже, для формування коректної моделі складного процесу АнД бажано мати додатковий формальний опис – семантичну модель ПрО мовою XML, яку сприймає нотація BPMN. Найбільш прийнятним описом, що може доповнювати та пояснювати графічні діаграми BPMN та сприймається засобами моделювання, вважається онтологічна модель, серіалізована у вигляді OWL-файлу. Тоді в якості технології взаємодій OWL-моделей з XML-файлами може бути запропонований онтологічний аналіз [259]. Цей аналіз базується на положенні двостороннього відображення між XML-файлом нотації BPMN (у вигляді спрямованого графу – $Gr(Bp_N)$) і онтологією ***Ont(SD)***_{BP}, яка додає семантичне пояснення та інтерпретацію процесів. Це означає створення реальної онтології ПрО, яка серіалізована у вигляді OWL-файлу. Така онтологія має містити в собі всі класи OWL, які співвідносяться з класами BPMN-нотації,

а також обмеження та анотації. На основі реальної онтології ПрО можна провести декілька типів аналізів BPMN-моделі, наприклад, з метою перевірки виконання умов настання кінцевих або проміжних подій, а також коректності формування багатофакторних умов поділу потоку процесу шлюзами моделі сценарію.

Формально концептуальну модель онтології ПрО – $Ont(SD)_{BP}$, що описує сценарій Анд та співвідноситься з нотацією BPMN, можна представити у вигляді математичного виразу

$$Ont(SD)_{BP} = \langle C^{(E)}, An^{(C)}, Rel^{(H)}, T^{(A)}, Ax^{(s)}, Rul^{(S)}, Ex^{(C)} \rangle$$

де: $\{C^{(E)}\}$ – підмножина класів (<Classes>) основних концептів (<Entities>) - понять ПрО, які співвідносяться з основними класами нотації BPMN ;

$\{An^{(C)}\}$ – підмножина «властивість анотації» (<Annotation Property>), що використовується для надання концептам (класам, сутностям, екземплярам) онтології додаткової інформації при опису бізнес-процесів;

$\{Rel^{(H)}\}$ – підмножина відношень (<Object properties>) між класами $\{C^{(E)}\}$ та властивостями $\{T^{(A)}\}$, що задаються предикатами $\{Rel^{(H)}\}$ – дієсловами через які описуються відношення (типи зв'язків) у визначеній ПрО та співвідносяться з основними класами нотації BPMN;

$\{T^{(A)}\}$ – підмножина атрибутів (<Data properties>), що описує властивості класів $\{C^{(E)}\}$ (назви, їх типи, області значень даних тощо);

$\{Ax^{(s)}\}$ – підмножина аксіом (<Axioms>), що визначають основні поняття ПрО, які завжди для неї істинні. Аксіоми записуються предикатами першого порядку і задають правила виводу для встановлення співвідношення між підмножиною класів (<Classes>) онтології ПрО та основними класами нотації BPMN;

$\{Rul^{(S)}\}$ – підмножина правил логічного виводу для встановлення дійсних співвідношень між підмножиною класів (<Classes>) онтології ПрО та основними класами нотації BPMN;

$\{Ex^{(C)}\}$ – множина екземплярів (<Instances>) класів $\{C^{(E)}\}$, яка може використовуватися для встановлення співвідношення між підмножиною класів (<Classes>) онтології ПрО та основними класами нотації BPMN.

Взаємодія між конструкціями графічної нотації BPMN та OWL онтологією ПрО показана на рисунку 2.1. Напрямок відображення інтерпретації моделі сценарію АнД описує відображення (Mapping) від нотації BPMN до онтології ПрО, а відображення уявлення – зворотне відображення.

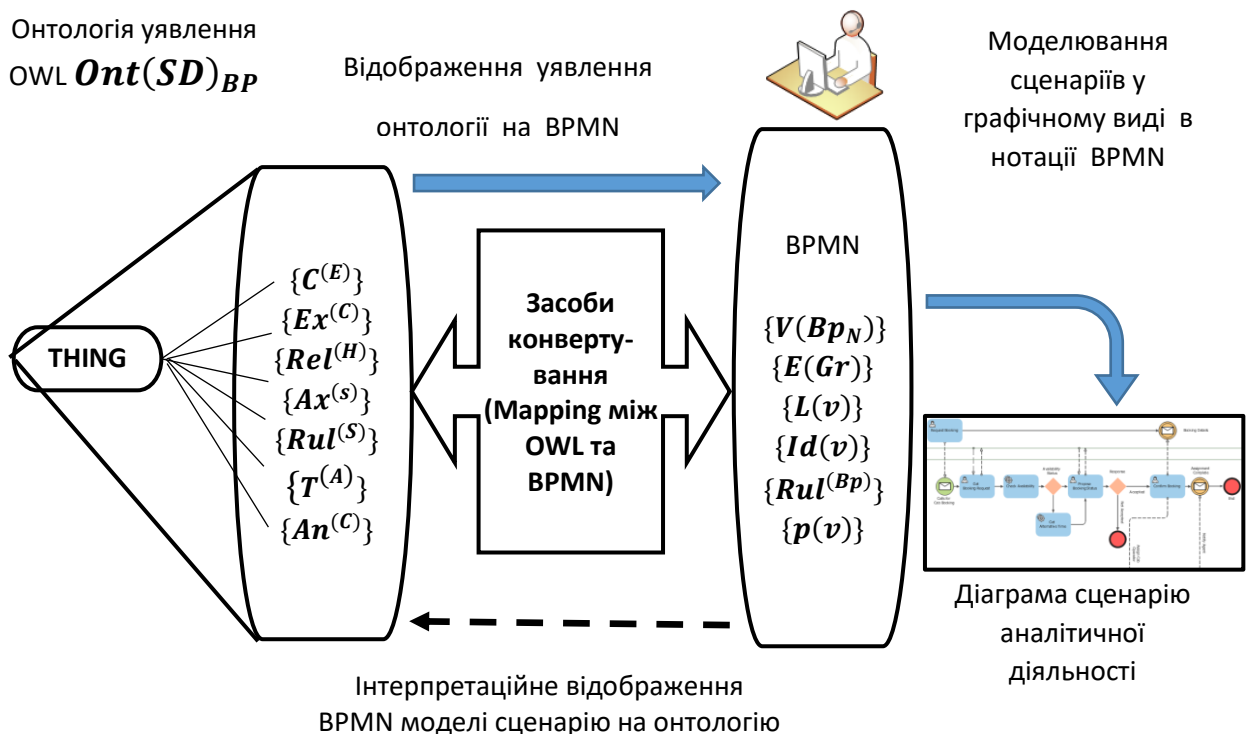


Рисунок 2.1 – Взаємодія між конструкціями графічної нотації BPMN та онтологією ПрО

Таке рішення має забезпечувати взаємне конвертування (Mapping) між конструкціями графічної нотації BPMN та OWL онтологією на мові XML. Умовою такої взаємодії є підтримка співвідношення 1:1 між концептами OWL-онтології та примітивами BPMN-нотації, які входять до складу спрямованого графу $Gr(Bp_N)$ моделі сценарію. Це вирішується на другому етапі методу за рахунок побудови тривірневої моделі онтології отриманих BPMN-моделей для кожного концепту (рисунок 2.1), що є розвитком положення теорії побудови моделі багаторівневої онтології (Bunge-Wand-Weber, BWW). Другий рівень (найбільш складний) складає мета-модель BPMN 2.0, яка і визначає

співвідношення 1:1 класів онтології до графічних примітивів (Core Element Set), та дозволяє відобразити графічні елементи BPMN в поняттях онтології.

2.3 Особливості формування онтології для моделювання сценарію в графічній нотації

Основним компонентом технології перетворення та встановлення співвідношення концептів онтологічного уявлення ПрО до графічних примітивів нотації BPMN є «Засоби конвертування між OWL та BPMN (Mapping)» (рисунок 2.1). Засоби конвертування являють собою спеціальне ПЗ, яке сприймає OWL-модель у вигляді XML-файлу та модель процесу в нотації BPMN теж у вигляді XML-файлу. В основі технології конвертування між OWL та BPMN лежать положення теорії побудови моделі багаторівневої онтології – Bunge-Wand-Weber (BWW) [146], яка дозволяє описувати та співвідносити OWL-модель ПрО з моделлю бізнес-процесу в нотації BPMN.

Онтологічний аналіз, спираючись на теоретичні положення BWW, дозволяє класифікувати п'ять станів співвідношень, які можуть виникнути при зіставленні концептів моделі OWL та графічних примітивів моделі процесу в нотації BPMN. У порядку важливості це такі співвідношення: еквівалентність, дефіцит, нерозрізненість, надмірність, неоднозначність [263]. У таблиці 2.1 наведено графічну інтерпретацію співвідношень та семантичне пояснення існування такого співвідношення.

Таблиця 2.1

Співвідношення	Коментарі	Відображення	Графічна інтерпретація
Еквівалентність	Однозначне співвідношення між концептами онтології і примітивами нотації	(1:1)	<p>Концепт онтології Примітив нотації</p>
Дефіцит	Окремі концепти онтології не знаходять відображення в примітивах нотації	(1:0)	<p>Концепт онтології Примітив нотації</p>

Нерозрізненість	Концепт онтології може бути відображений відразу в кількох примітивах нотації	(1:n)	<p>Концепт онтології</p> <p>Примітив нотації</p>
Надмірність	Жоден концепт онтології не може бути відображений в примітив нотації	(0:1)	<p>Концепт онтології</p> <p>Примітив нотації</p>
Неоднозначність	Більш ніж один концепт може бути відображений в примітив нотації	(n:1)	<p>Концепт онтології</p> <p>Примітив нотації</p>

Метою онтологічного аналізу, є виявлення глибокої структури – онтологічних невідповідностей між концептами інформаційної моделі ПрО (організації) та можливостями примітивів графічної нотації для опису конкретних сценаріїв Анд. Отже, онтологічний аналіз фокусується на вмісті (сенсі) концепту, а не на формі (синтаксисі), якщо два концепти онтології мають однакову семантику, але різний синтаксис, онтологічний аналіз не може розрізнити їх.

Важливий висновок, який можна зробити на підставі застосування онтології BWW, полягає в необхідності ретельного обґрунтування набору концептів моделі OWL-онтології та примітивів моделі процесів для існування співвідношення 1:1. Але на практиці, враховуючи складність системних вимог ПрО, розробник зіштовхується з серйозними проблемами, якщо множина концептів онтології та примітивів BPMN (Mapping) не є чітко визначеною. Тобто, швидше за все, онтологічний аналіз (для складної ПрО) призведе до невідповідностей між результатами часткових онтологічних аналізів, які проводяться різними виконавцями через розподіл робіт, та вимогами до

системи моделювання сценаріїв Анд в цілому. Виходом з цього становища є розробка ієрархії – багаторівневої онтології моделі BWW (рисунок 2.2). Відповідно до базових положень моделі BWW кожен рівень OWL-онтології зосереджується на опису сутностей певного домену онтології сценаріїв Анд та відображує певну модель подання знань.

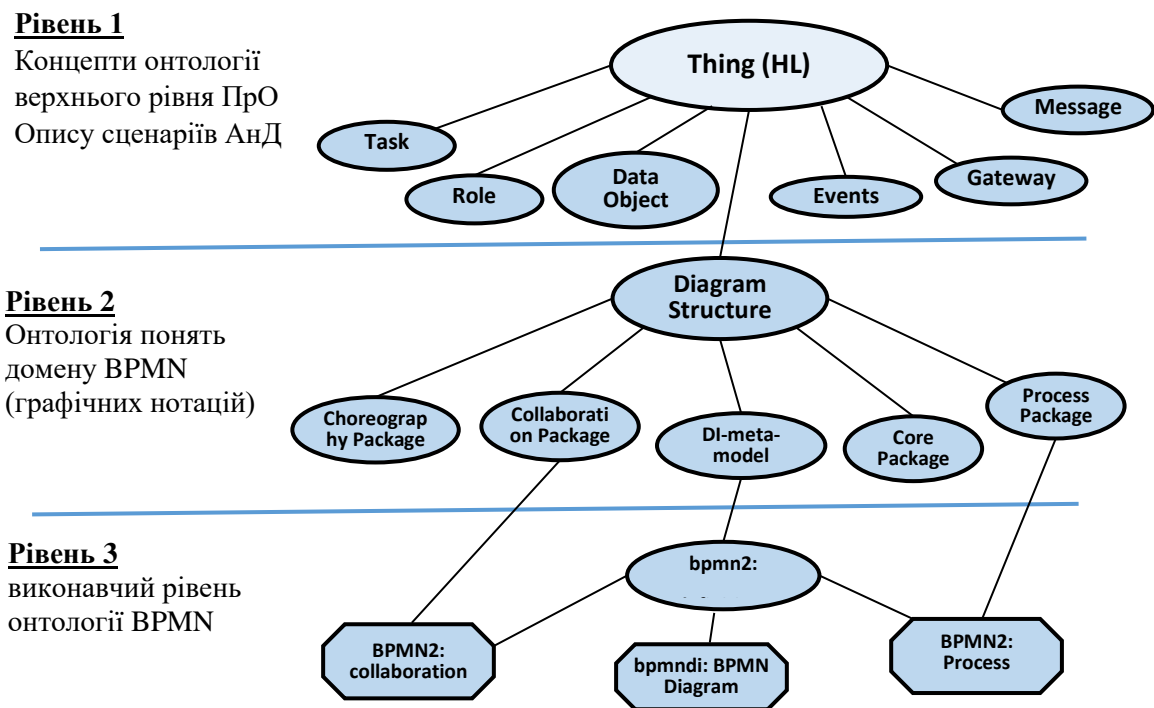


Рисунок 2.2 – Ієрархія онтології ПрО для опису співвідношення між концептами OWL-онтології і примітивами моделі сценарію в нотації BPMN

Так, рівень 1 (верхній) в ієрархії онтології відображує семантику та базові поняття певної ПрО, які на концептуальному рівні мають співвідносяться з базовими поняттями нотації BPMN. Базові поняття онтології описують галузь, мету та типові задачі аналітичного дослідження, кінцевий результат та типи його оформлення, початкові та проміжні умови виконання, потрібні документи та інформаційні ресурси, а також таксономію виконавців – акторів та їх опис. В термінах моделі подання знань – Open Knowledge Base Connectivity [47], це базові сутності, які представлені наступними класами:

<Task> – ієрархія задач аналітичного дослідження з поділом на загальні та елементарні задачі. Властивості класу визначають тип задачі, назву, мету,

приналежність, кінцевий результат та типи звітних матеріалів для кожної задачі, а також взаємозв'язки з іншими класами онтології, включаючи й онтологію понять домену BPMN;

<Events> – підмножина подій, які на семантичному рівні визначають умови виконання, завершення або логічного переходу на нову задачу.

<Role> – клас, який описує організаційну структуру, перелік ролей виконавців сценарію, їх повноваження, права доступу до інформаційних ресурсів та документів, крім того клас визначає взаємодію з відповідними класами домену BPMN;

<Data Object> – клас, який описує метадані інформаційних ресурсів, потрібних для виконання завдань аналітичного дослідження, включаючи документи та бази даних.

<Gateway> – підмножина правил, які описують формування логічних операторів для позначення переходів розгалуженого сценарію АнД при виникненні певних умов.

<Message> – клас для відображення взаємодії між учасниками процесу через набори потоків повідомлень, які представляється у вигляді символів обміну повідомленнями, пов'язаних посиланнями з учасниками інформаційної взаємодії в рамках сценарію.

Наступний нижчий рівень 2 – домен нотації BPMN – описує мета-модель BPMN 2.0 з прив'язкою до графічних примітивів нотації (Core Element Set). Мета-модель BPMN 2.0 через суперклас <DiagramStructure> зв'язана з онтологією верхнього рівня. Клас <ClassDiagramStructure> містить основні пакети (суперкласи), що описують специфікації всіх графічних примітивів нотації BPMN 2.0, властивості та умови їх використання і взаємодії з іншими сутностями домену нотації. Отже в мета-моделі онтології BPMN 2.0 виділені наступні класи:

<ChoreographyPackage> - клас опису діаграм та відображення приватних процесів у вигляді дій, які включають обмін повідомленнями між процесами;

<CollaborationPackage> - клас опису діаграм та відображення взаємодії між учасниками процесу через набори потоків повідомлень, які відображують інформаційну взаємодію.

<CompletePerformance> – клас, який охоплює основні елементи BPMN та описує інфраструктуру моделі: діаграми процесів, діаграми співпраці, діаграми взаємодії, елементи загального користування, сервіси.

<Core> - клас, який містить дочірні класи, які є загальними для шарів моделювання <Process> (Процес), <Choreography> (Хореографія), <Collaboration> (Співпраця) та необхідні для формування діаграми сценарію в цілому.

<DI-meta-model> - клас, який визначає елементи графічного відображення моделі процесу для переносу їх в конструкції мови XML.

<ProcessPackage> - клас, який визначає різні типи завдань, щоб розділити типи поведінки учасників сценарію, наприклад, GlobalManualTask, GlobalScriptTask і GlobalBusinessRuleTask.

Кожен з представлених вище суперкласів складається з дочірніх класів, які успадковують всі асоціації та властивості класу.

Рівень 3 – виконавчий рівень – більш детально описує семантичну відповідність кожного вузлу OWL-моделі вузлам спрямованого графу – $Gr(Bp_N)$ BPMN -діаграми. Цей рівень встановлює умови співвідношення вузлів реальної діаграми сценарію АНД з екземплярами відповідних класів OWL-моделі. Для цього застосовується механізм ідентифікації примірників BPMN-діаграми через суперклас <BPMNDiagram>, який являє собою зображення BPMN-діаграми, як набору екземплярів відповідних класів. В свою чергу <BPMNDiagram> має три дочірніх класи:

- <BPMNShape> - опису набору графічних елементів, які складають реальну діаграму;
- <BPMNEdge> - опису ребер спрямованого графу, що зв'язують графічні елементи реальної діаграми;

– $\langle \text{BPMNPlane} \rangle$ - опису контейнеру (pool), який поєднує та ідентифікує вузли спрямованого графу – $Gr(Bp_N)$ -.

Властивості ($\langle \text{Data Property} \rangle$ (сутності OWL моделі 2-го рівня), які містяться в системних вимогах, що характеризують семантичний опис сценарію та їх відповідність в системних вимогах організації для відображення процесів.

Ієрархічний зв'язок між рівнем 1 та рівнем 2 насправді є зв'язком між концептами домену ПрО, наприклад, описом типових завдань та процедур з онтологією мета-моделі BPMN 2.0 (рівень 2) та їх відображення на онтології верхнього рівня (таксономія базових класів ПрО). Цей зв'язок забезпечує узгодженість між двома рівнями та зменшує семантичне навантаження на рівень графічних нотацій, що, безпосередньо, описує процеси.

Щоб створити онтологію сценарію АнД, яка б співвідносилася з моделлю BPMN, необхідно зрозуміти відносини між компонентами моделей BPMN і онтологією OWL. Тобто необхідно встановити співвідношення між примітивами BPMN таким чином, щоби атрибути, екземпляри, спадкування та відносини між класами або властивостями збігалися з класами відповідних понять OWL моделі ПрО. Крім того, необхідно враховувати той факт, що в об'єктно-орієнтованій парадигмі процес розглядається як об'єкт, що піддається обробці, трансформації та зміні його стану, з урахуванням впливу подій, які ініціюють трансформацію.

Таким чином, сценарій АнД можна розглядати не тільки як сукупність робіт, але ж як послідовність зміни станів об'єкту, що підлягає аналітичній обробці в ході виконання сценарію. Це зауваження надзвичайно важливе для коректного сприйняття самого процесу моделювання поведінки об'єкту. Тому, сценарієм аналітичного процесу вважається не тільки опис послідовності виконання аналітичних кроків і функцій, але також й зміни станів об'єкту дослідження, які можуть виникнути в процесі виконання сценарію.

2.4 Створення мета-моделі онтології нотації BPMN 2.0

Сценарій в нотації BPMN 2.0, хоча і є дуже наочним для сприйняття людиною, але в нотації BPMN 2.0 він не є частиною знань організації. Для

цього BPMN-модель сценарію має бути конвертована в OWL-модель, яка за визначенням вважається складовою БЗ. Це означає, що до сценарію можуть бути застосовані механізми та переваги уніфікованої моделі системи подання знань – Open Knowledge Base Connectivity, включаючи й формування запитів на різних мовах DL Query, SPARQL, Semantic Web Rule Language та машин виводу – Reasoners. Умовою взаємодій моделей з різними платформами є забезпечення семантичної відповідності між конструкціями графічної нотації BPMN та онтологією ПрО.

Забезпечення семантичної відповідності між різними моделями – доволі складна наукова задача. Виходячи з теоретичних положень, запропонована трирівнева модель онтології отриманих BPMN-моделей. Згідно багаторівневої онтології моделі BWW другий рівень (найбільш складний) складає мета-модель BPMN 2.0, яка і визначає співвідношення класів онтології до графічних примітивів (Core Element Set). Згідно визначенню, запропонованому Gholizadeh [267], «модель – це абстрактне уявлення системи, а мета-модель – дозволяє відобразити графічні елементи BPMN в поняттях онтології». Конвертація BPMN діаграм в OWL-моделі призводить до створення БЗ організації [268]. Це розкриває можливість здійснення семантичного аналізу БП із застосуванням відомих алгоритмів аналізу за допомогою семантичних запитів Web-мовами: DL Query, SPARQL, Semantic Web Rule Language та іншими. Отже, онтологічна мета-модель BPMN 2.0 являє собою деталізований опис класів (графічних примітивів), процесів, учасників, інформаційних ресурсів, умов розгалуження потоків з можливістю співвідношення елементів діаграми з конкретними екземплярами діаграми для дослідження якості OWL-моделі на рівні знань.

Важливим фактором при конвертації BPMN-моделі до XML-формату з подальшим перетворенням її до OWL-моделі є вибір мови серіалізації. Найбільш значущим критерієм мови вважається її виразливість [270]. Це означає, що, по-перше, мова серіалізації повинна бути машино зчитуваною для забезпечення сумісності моделі на рівні виконання, а, по-друге, забезпечувати

коректну передачу семантики кожного мовного примітиву. Виходячи з цих вимог найбільш прийнятною вважається мова XPDЛ (XML Process Definition Language), яка має синтаксис XML, що не тільки сприймається засобами виконання моделей, але й основою синтаксису мови OWL.

Підсумовуючи аналіз стану теоретичних досліджень, а також реалізації мета-моделей BPMN 2.0 [261], слід констатувати, що на сьогодні відомі кілька повних реалізацій мета-моделі BPMN. Перша реалізація виконана в університеті Верони [267] у 2014 році, однак вона з невідомих причин не зазнала поширення. Друга онтологія була створена під керівництвом Natschläger у 2019 році [268]. Але ця онтологія не має вільного розповсюдження, тому з міркувань практичного застосування вважається недоступною для використання. І, нарешті, онтологічна мета-модель BPMN 2.0 від Institute for Public Policy Research (IPPR), який є не комерційною фундацією (Великобританія). Онтологічна мета-модель повної версії нотації BPMN 2.0 є вільно розповсюдженою та доступною на сайті для скачування <https://github.com/reiterma13/IPPR> у вигляді OWL-файлу з назвою IPPR.owl. Ще мета-модель BPMN 2.0 Ontology створена в Університеті прикладних наук в місті Грац, Австрія [269] з метою надання еталонного файлу онтології в нотації BPMN 2.0 для з'ясування, чи відповідають розроблені діаграми моделі процесів специфікації нотації BPMN 2.0. Результатом перевірки є три прототипи на основі мови Java для порівняння онтології BPMN зі стандартною або іншою онтологією та файл JAR, який описує створену онтологію Про.

Технологія та інтерфейс засобу IPPR.owl (сайт IPPR, рисунок 2.3) дозволяє розробникам BPMN-моделі сценарію перевірити в on-line режимі, чи відповідає отримана OWL-модель сценарію еталону специфікації BPMN 2.0. При цьому папки <BPMN2OWL>, <CompareOWL> та <OWLDescriber> містять файли JAR, які можна запустити. Папка <IPPRProjectFiles> містить вихідний код перевірки невідповідності кінцевих або проміжних подій, багатofакторних умов поділу потоку та інших факторів.

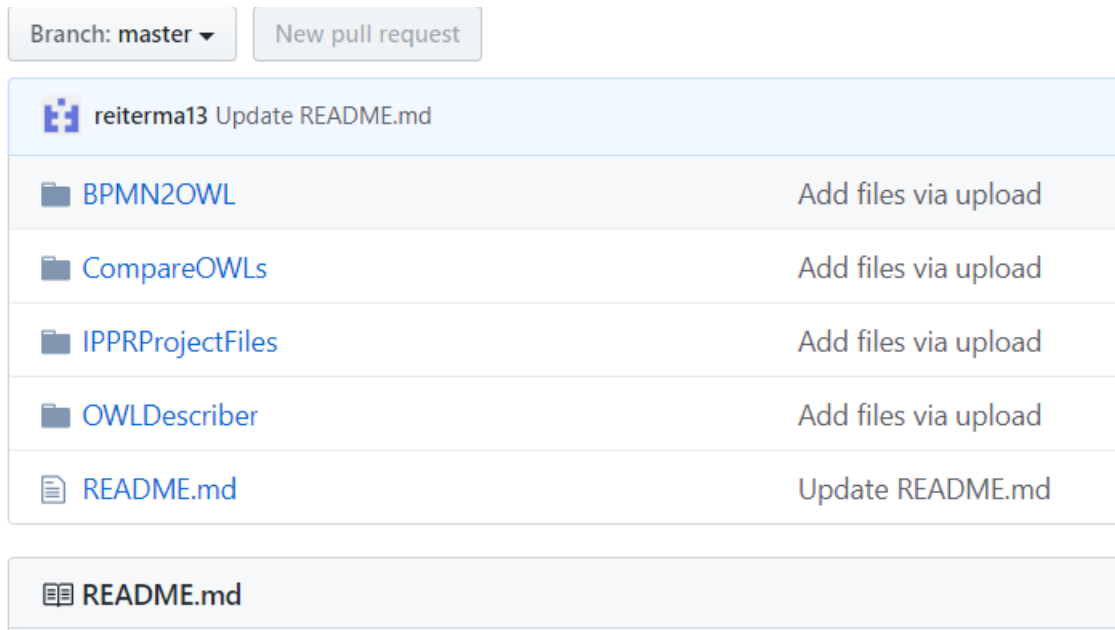


Рисунок 2.3 – Технологія та інтерфейс для перевірки відповідності
OWL-моделі еталону специфікації BPMN 2.0

Отже, в якості базової технології (шаблону) для моделювання та перевірки сценаріїв АнД використовується мета-модель BPMN 2.0 (IPPR) як рівень 2 багаторівневої онтології BWW. Мета-модель BPMN 2.0 (IPPR) містить всі базові графічні примітиви нотації BPMN 2.0 та використовується як проміжний рівень між концептуальним та виконавчим рівнями теоретичного положення BWW. Фізично модель являє собою таксономію класів (рисунок 2.4), які є відображенням графічних примітивів нотації BPMN 2.0 та співвідносяться з відповідними класами OWL-нотації. Як слід з рисунку 2.3 клас верхнього рівня `<owl:Thing>` містить два головних класи: `<ClassDiagramStructure>` та `<CMOF>`. Клас `<ClassDiagramStructure>` складається з пакетів (суперкласів), що описують сутності та специфікації основних процесів нотації: `<ChoreographyPackage>`, `<CollaborationPackage>`, `<CompletePerformance>`, `<Core>`, `<DI-meta-model>` та `<ProcessPackage>`. В свою чергу кожний суперклас має певне розмаїття дочірніх класів, які визначають графічні примітиви нотації BPMN 2.0 та особливості їх використання при побудові діаграм процесів.

Наприклад, клас `<DI-meta-model>` використовуються для опису графічних елементів нотації BPMN 2.0, з яких побудовано реальну діаграму процесу.

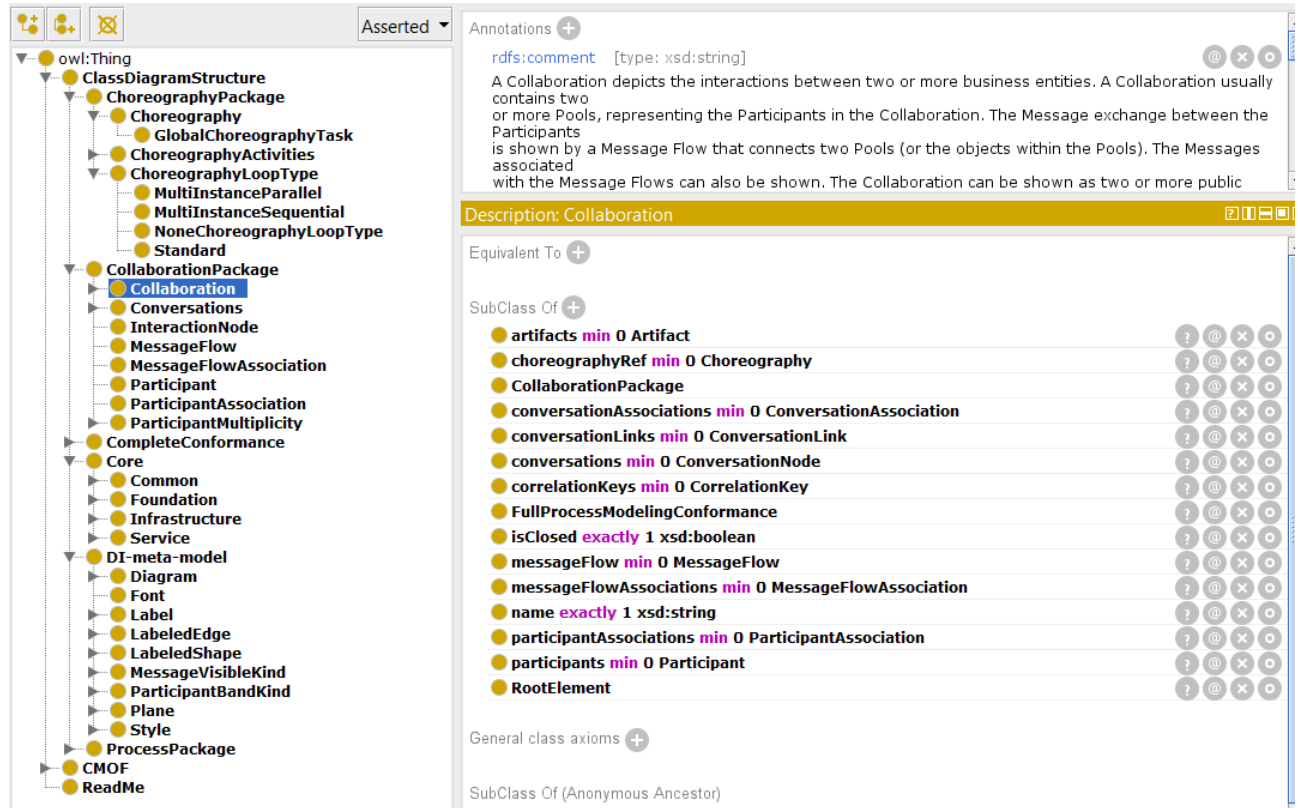


Рисунок 2.4 – Базова структури OWL мета-моделі нотації BPMN 2.0

На рисунку 2.5 наведено приклад відображення в мета-моделі BPMN 2.0 складу діаграми класу `<Task>`, який в свою чергу є дочірнім класом `<DescriptiveProcessModelingConformance>`. Як видно з діаграми клас `<Task>` також успадковує всі асоціації і властивості класу `<Activity>`. Клас `<Descriptive>` пов'язаний із графічними елементами та атрибутами, які використовуються аналітиками у моделюванні діаграми на високому рівні. В мета-моделі цей суперклас описує властивості всіх примітивів графічних елементів нотації BPMN 2.0, які позначені класами: `<DataObject>`, `<MessageEndEvent>`, `<FlowElement>`, `<EndEvent>`, `<Events>`, `<Group>`, `<ResourceRole>`, `<Resource>`,

<DataAssociation>, <SubProcess>, <Task> та інші. Сутності, які описують ці

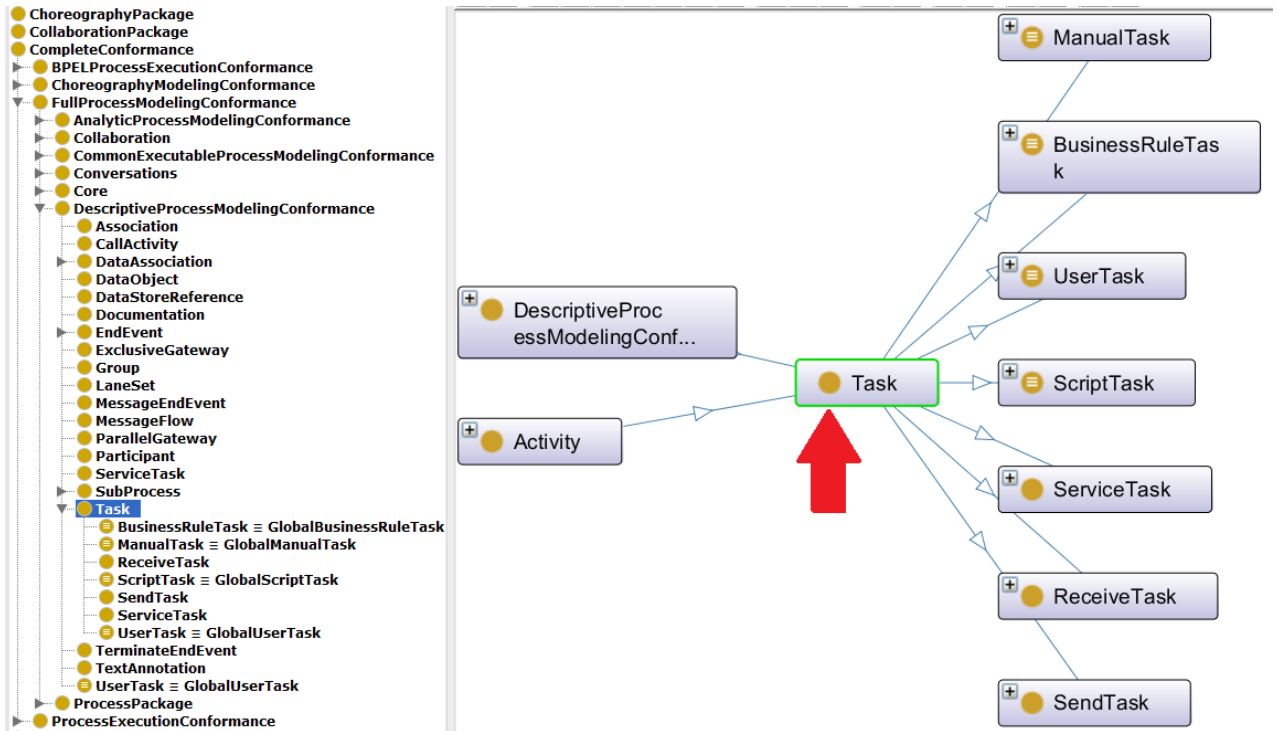


Рисунок 2.5 – Приклад відображення складу діаграми класу <Task>

класи, використовуються аналітиками в ході моделюванні діаграм процесів високого рівня. Кожний з класів описується певним набором допоміжних атрибутів – параметрів. Отже клас <Task> складається з семи дочірніх класів, які співвідносяться з відповідними графічними примітивами в нотатції BPMN 2.0. – типами задач:

1. <BusinessRuleTask> – ця сутність надає механізм, що забезпечує отримання результатів розрахунків з двигуна – engine. Задача <BusinessRuleTask> також успадковує атрибути та модельні асоціації класу <Activity>.

2. <ManualTask> – це завдання, яке буде виконуватися без допомоги будь-якого механізму виконання бізнес-процесів або будь-якої програми. Воно може розглядатися як некероване в розумінні того, що двигун бізнес-процесів (engine) не відстежує початок і завершення такого завдання.

3. <ReceiveTask> – призначена для очікування надходження повідомлення від зовнішнього учасника. Після отримання повідомлення завдання буде завершено. Задача часто використовується для запуску процесу.

Для того, щоб завдання отримало екземпляр процесу, в атрибуті `<instantiate>` **ОБОВ'ЯЗКОВО** встановлюється значення `true`. Під час виконання, дані автоматично переміщуються з Повідомлення (`<message>`) до вихідних даних.

4. `<ScriptTask` – виконується двигуном бізнес-процесів (`engine`). Аналітик визначає сценарій мовою, яку програма `engine` може інтерпретувати. Задача `<ScriptTask>` успадковує атрибути та модельні асоціації класу `<Activity>`.

5. `<SendTask>` – призначена для надсилання повідомлення зовнішньому учаснику. Після відправлення повідомлення – завдання виконується. Учасник, якому надсилається повідомлення, визначається потоком повідомлень у межах співпраці процесу. Задача `<SendTask>` успадковує атрибути та модельні асоціації класу `<Activity>`, який пов'язаний з класом `<Task>`.

6. `<ServiceTask>` – виконується за допомогою веб-служб або автоматизованих додатків та успадковує атрибути та модельні асоціації класу `<Activity>`. Задача має один вхідний набір і щонайбільше один вихідний набір.

7. `<UserTask>` – виконується за допомогою ПЗ користувача та визивається через менеджер списку завдань. Задача успадковує атрибути та модельні асоціації класу `<Activity>`. Може бути реалізовано за допомогою різних технологій, визначених атрибутом реалізації.

В онтології кожний клас може бути пов'язаний через родові або асоціативні відношення з іншими класами та властивостями. На рисунку 2.6-а показано опис властивостей класу `<Task>` через більш детальне визначення атрибутів та їх допустимих значень. Так, клас `<Task>` може мати ім'я із значеннями `name min 0 xsd:string` або `name max 1 xsd:string`, стартові умови через параметр `startQuantity exactly 1 xsd:integer`, визначати контрольоване завдання `auditing max 1 Auditing` або ні – `auditing min 0 Auditing`. Також клас `<Task>` визначається ідентифікаційним значенням через параметр `id exactly 1 xsd:string`, має можливість визначати ресурси `resources min 0 ResourceRole`, потрібні для виконання задачі та умови вводу `dataInputAssociations min 0 DataInputAssociations` або виводу даних `dataOutputAssociations min 0`

DataOutputAssociations. Важливим параметром є визначення умов виконання завдання у циклі через *loopCharacteristics max 1 loopCharacteristics*. Крім того, для визначення умов виконання класу <Task> використовують багато інших параметрів.

Слід відзначити, що в рамках класу <Task> підмножина параметрів опису для кожного типу задачі може незначно відрізнитися, як показано на рисунку 2.6-б для задачі <ServiceTask>.

Description: Task

- DescriptiveProcessModelingConformance

General class axioms +

SubClass Of (Anonymous Ancestor)

- auditing min 0 Auditing
- monitoring min 0 Monitoring
- name min 0 xsd:string
- auditing max 1 Auditing
- name max 1 xsd:string
- categoryValueRef min 0 CategoryValue
- monitoring max 1 Monitoring
- documentation min 0 Documentation
- extensionDefinitions min 0 ExtensionDefinition
- extensionValues min 0 ExtensionAttributeValu
- id exactly 1 xsd:string
- boundaryEventRefs min 0 BoundaryEvent
- ioSpecification min 0 InputOutputSpecification
- default min 0 SequenceFlow
- loopCharacteristics min 0 LoopCharacteristics
- isForCompensation exactly 1 xsd:boolean
- properties min 0 Property
- startQuantity exactly 1 xsd:integer
- dataOutputAssociations min 0 DataOutputAss
- resources min 0 ResourceRole
- completionQuantity exactly 1 xsd:integer
- default max 1 SequenceFlow
- loopCharacteristics max 1 LoopCharacteristic
- dataInputAssociations min 0 DataInputAssoci
- ioSpecification max 1 InputOutputSpecificatio

Рисунок 2.6-а – Приклад опису класу <Task>

Description: ServiceTask

SubClass Of (Anonymous Ancestor)

- auditing min 0 Auditing
- monitoring min 0 Monitoring
- name min 0 xsd:string
- auditing max 1 Auditing
- name max 1 xsd:string
- categoryValueRef min 0 CategoryValue
- monitoring max 1 Monitoring
- documentation min 0 Documentation
- extensionDefinitions min 0 ExtensionDefinition
- extensionValues min 0 ExtensionAttributeValue
- id exactly 1 xsd:string
- boundaryEventRefs min 0 BoundaryEvent
- ioSpecification min 0 InputOutputSpecification
- default min 0 SequenceFlow
- loopCharacteristics min 0 LoopCharacteristics
- isForCompensation exactly 1 xsd:boolean
- properties min 0 Property
- startQuantity exactly 1 xsd:integer
- dataOutputAssociations min 0 DataOutputAssociation
- resources min 0 ResourceRole
- completionQuantity exactly 1 xsd:integer
- default max 1 SequenceFlow
- loopCharacteristics max 1 LoopCharacteristics
- dataInputAssociations min 0 DataInputAssociation
- ioSpecification max 1 InputOutputSpecification

Instances +

Рисунок 2.6-б – Приклад опису властивостей класу <ServiceTask>

Онтологія BPMN 2.0, створена в рамках проекту IPPR, містить всі елементи BPMN 2.0 з їх атрибутами і модельними асоціаціями. Загальна мета-модель BPMN 2.0 Ontology містить 262 класи (Class), які пов'язані між собою 180 типами відносин (<Object Properties>) та 70 властивостей даних (<Data

Properties>). Всі посилання на інші елементи і всі успадковані атрибути включені в онтологію BPMN.

Представлена OWL-онтологія нотації BPMN 2.0 має можливість встановлювати певні межі достовірності даних, які визначають структури даних. Наприклад, дійсне ім'я людини складається з ім'я та прізвища, в той час, як це твердження невірно для назви компанії або задачі. Тому обмеження значень відношень та атрибутів залежать від семантичного контексту класу.

Щоб визначити ці обмеження для класів OWL-онтології, повинні бути визначені обмеження. Онтологія розрізняє обмеження на тип даних (обмеження типу даних) і обмеження на класи і властивості об'єкта (обмеження властивості об'єкта). Обмеження на тип даних впливають на вміст об'єкта даних і обмежують його (наприклад, числами). Обмеження на класи і властивості об'єктів впливають на їх передбачуване використання.

Для обмеження класів і властивостей об'єктів використовується концепт класу <Expression>. Використовуючи цей концепт, усі обмеження можуть бути визначені з використанням тексту природною мовою, якій зберігається за допомогою атрибута документації, успадкованого від класу <BaseElement> для обмеження даних конкретними об'єктами:

- *documentation min 0 Documentation*
- *extensionDefinitions min 0 ExtensionDefinition*
- *extensionValues min 0 ExtensionAttributeValue*

2.5 Критерії повноти опису для забезпечення співвідношення між різними моделями сценаріїв

Сучасні графічні технології моделювання BPMN (BizAgi Process Modeler, Bonita, ERWin Process Modeler та інші) дозволяють аналітику не тільки створити працюючий прототип сценарію, але й протестувати його роботу, та на ранніх етапах виявити ступінь відповідності моделі реальному процесу і, таким чином, зробити процедуру верифікації сценарію більш об'єктивною.

Основними категоріями графічних елементів нотації BPMN, з якими оперує аналітик при побудові сценаріїв Анд, є: діяльність (Activities),

підпроцеси (Subprocess), події (Events), шлюзи (Gateways), артефакти (Artifacts), об'єкт даних (Data objects), бази даних (Data stores), пули (Pools), доріжки (Lanes), потоки управління (Sequence flows) и потоки повідомлення (Message flows).

Для ефективної конвертації створеної BPMN-моделі в OWL-модель сценарію АнД, яка здатна відповідати на семантичні DL-запити, графічні елементи BPMN-моделі мають бути пов'язані з текстовою документацією відповідно до формату системи моделювання BPMN. Але слід нагадати, що не існує стандартизованого способу структурувати організацію текстової інформації елементів BPMN. Тому, для створення BPMN-моделі, яка ефективно конвертується в OWL-модель сценарію АнД, необхідно дотримуватися критеріїв повноти опису моделі сценаріїв. Отже, для створення коректного співвідношення пропонується 9 критеріїв, що дозволяють забезпечити співвідношення 1:1 між графічними елементами нотації та їх онтологічними еквівалентами.

1. Кожний процес <Pool> повинен мати хоча би одну функціональну задачу <Task>, а також супроводжуватися початковою подією та, принаймні, одною кінцевою з назвами <Start Events> (Початкові події), та <End Even> (Кінцеві події) відповідно.

2. Кожний процес <Pool> повинен асоціюватися щонайменше з одним учасником <Participant> - виконавцем процесу.

3. Текстова супроводження кожного графічного елемента нотації BPMN повинне містити принаймні одну з його описових властивостей (properties): <Name> (ім'я), <Description> (опис) або <Documentation> (документація).

4. Тип діяльності <Task> на діаграмі завжди повинен визначатися унікальною назвою <Name> та супроводжуватися відповідним значком на діаграмі для більшої наочності.

5. При визначенні графічного елемента <задача <Task> властивість ім'я <Name> бажано використовувати дієслово в інфінітиві, яке відображує

семантику задачі та може сприяти коректній ідентифікації в OWL-моделі та формування SPARQL-запитів.

6. Будь-яка задача <Task> повинна виконуватися хоча б одним учасником процесу <Participant>.

7. Назва виконуваної задачі <Task> не повинна призначатися іменами шлюзу (Gateways) (навіть якщо шлюзи представляють логіку маршрутизації після прийняття рішення).

8. Потоки (Sequence flows), які виходять із шлюзу ексклюзивного типу (exclusive gateway), обов'язково повинні мати унікальні назви <Name>, за винятком випадків, коли є лише два потоки, що представляють "Так" або "Ні" ("Yes" or "No"). У цьому випадку лише одне з імен може бути явним, а інше може бути виведеним.

9. Потоки (Sequence flows), які виходять із шлюзу inclusive gateway також обов'язково повинні бути іменовані.

Запропоновані критерії корелюються з передовими методами моделювання інформаційних систем та сприяють додаванню розширених атрибутів OWL-моделі. Розширені атрибути допомагають документувати модель сценарію АнД, наповнюючи її семантичними подробицями, що дає можливість ідентифікувати деякі програмні елементи при виконанні семантичного аналізу методами формування різноманітних SPARQL-запитів або DL-логіки.

2.6 Оцінка результату конвертації моделі BPMN в OWL-файл

Неможливо порівняти моделі, представлені у вигляді OWL-файлів, на основі візуального огляду, оскільки графічного представлення генеруючої онтології S-BPM досі немає. Підтвердження конвертації може бути зроблено тільки на основі перевірки створених файлів OWL у відповідному редакторі онтології або за допомогою SPARQL-запитів.

В якості тесту для перевірки теоретичних положень взято BPMN-модель сценарію <Аналіз запитів>. Графічне відображення сценарію в нотації BPMN

наведено на рисунку 2.7. Безпосередньо для конвертації BPMN-моделі в OWL-файл використовувалася файл «Аналіз запитів.xpdl».

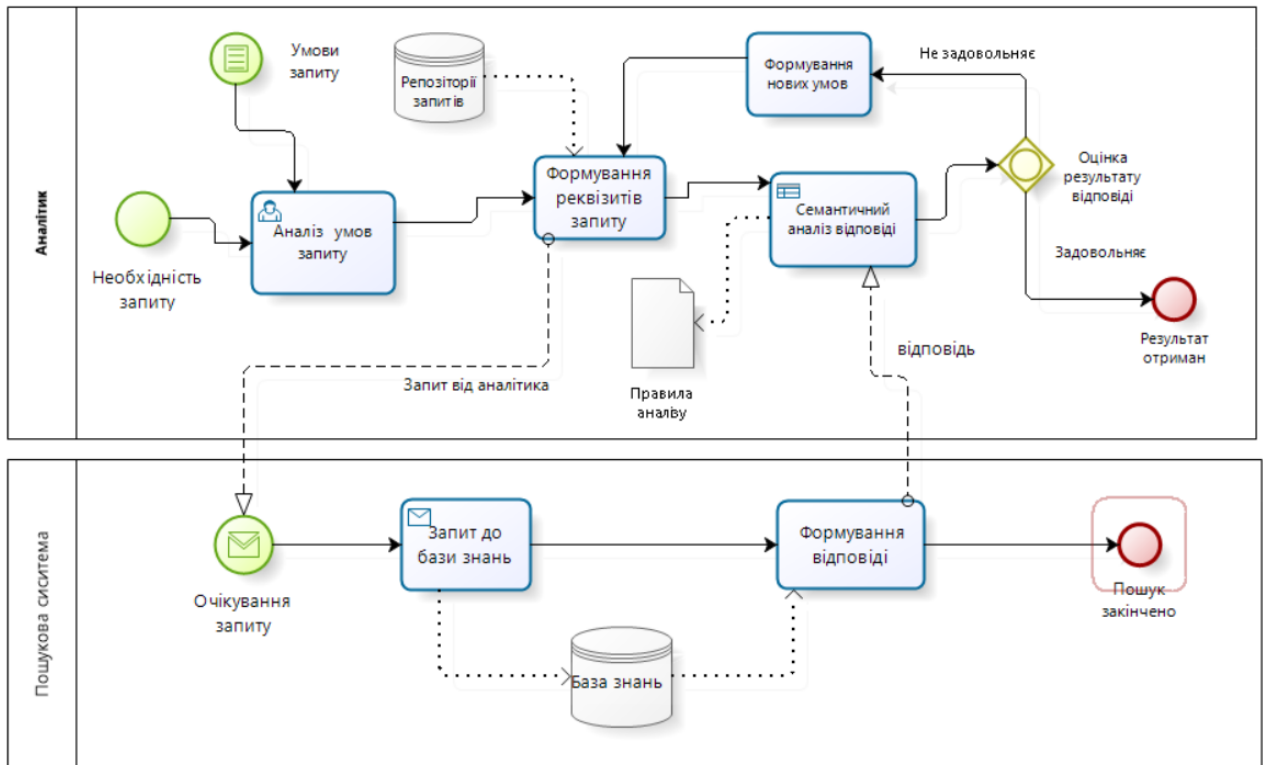


Рисунок 2.7 – Приклад сценарію аналітичної діяльності в нотації BPMN

Конвертація вхідного файлу являє собою ітераційний процес, який починається з аналізу XPDL-файлу для ідентифікації шаблонів елементів мета-моделі BPMN-онтології, з відповідними шаблонами XML-елементів у файлі XPDL для ідентифікації класів. Фрагмент тексту BPMN-моделі конвертований в OWL-файл наведено нижче:

```

<owl:Class rdf:about="http://www.reiter.at/ontology/generated#document">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.reiter.at/ontology/generated#bpmn2:definitions"/>
      <owl:qualifiedCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
      <owl:onClass rdf:resource="http://www.reiter.at/ontology/generated#bpmn2:definitions"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
  <owl:Class rdf:about="http://www.reiter.at/ontology/generated#bpmn2:definitions">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://www.reiter.at/ontology/generated#exporter"/>
        <owl:qualifiedCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>

```

```

    <owl:onDataRange rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class> <owl:Class rdf:about="http://www.reiter.at/ontology/generated#bpmn2:definitions">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.reiter.at/ontology/generated#exporterVersion"/>
      <owl:qualifiedCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
      <owl:onDataRange rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class> <owl:Class rdf:about="http://www.reiter.at/ontology/generated#bpmn2:definitions">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.reiter.at/ontology/generated#id"/>
      <owl:qualifiedCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
      <owl:onDataRange rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class> <owl:Class rdf:about="http://www.reiter.at/ontology/generated#bpmn2:definitions">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
rdf:resource="http://www.reiter.at/ontology/generated#targetNamespace"/>
      <owl:qualifiedCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
      <owl:onDataRange rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class> <owl:Class rdf:about="http://www.reiter.at/ontology/generated#bpmn2:definitions">
  <rdfs:subClassOf>
    <owl:Restriction>

```

Для відображення отриманої в процесі конвертації онтології використовується редактор Protégé 5 – (Стенфордський університет [119]), який відноситься до візуальних редакторів. Візуальні методи сприяють швидшому і повному розумінню структури знань предметної області.

На рисунку 2.8 наведена загальна схема онтології, отримана в результаті автоматичної конвертації вхідного XPDL-файлу за допомогою технології BPMN2SBPM. Схема онтології відображується редактором Protégé 5 та складається з наступних сутностей:

Class count (базових класів мета-моделі)

52

Object property (типів взаємодії між класами)	51
Data property (властивостей класів)	25
Individual (примірників, що описують діаграму)	690
Declaration axioms (декларованих аксіом)	818
Annotation (загальні анотації)	27

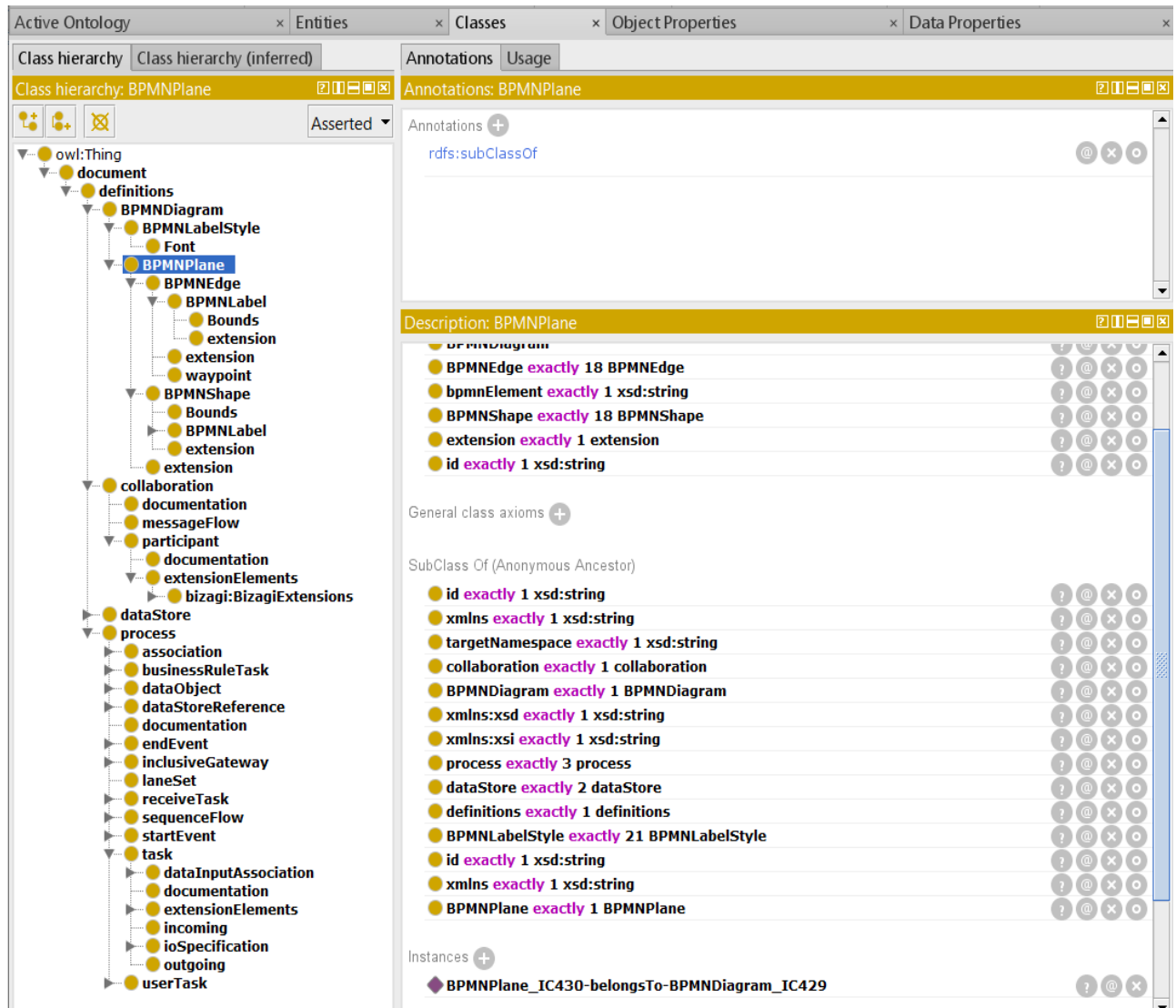


Рисунок 2.8 – Візуалізація конвертованого в OWL-файл сценарію аналітичної діяльності в нотації BPMN

BPMN-діаграма сценарію позначається через суперклас `<BPMNDiagram>`, який визначає зображення всієї або частини моделі BPMN-діаграми. Клас `<BPMNDiagram>` описується через специфікацію трьох дочірніх класів:

- `<BPMNPlane>` – визначає контейнер, який описує один процес на BPMN-діаграмі (еквівалент графічного елемента `<Pool>`) – кореневий елемент BPMN-діаграми.

- `<BPMNShape>` – визначає набір примірників графічних елементів нотації (Task, Event, Gateway, Sequence Flow, Message flow тощо), що входять до складу реального процесу BPMN-діаграму;

- `<BPMNEdge>` – опис ребер, які зв'язують примірники графічних елементів реальної BPMN-діаграми процесу;

Класи `<BPMNPlane>`, `<BPMNShape>` і `<BPMNEdge>` посилаються рівно на один елемент з моделі BPMN з використанням атрибута `<bpmnElement>`. Це візуальне скорочення, яке фактично нормалізує дві асоціації даних в моделі BPMN реального процесу.

Клас `<BPMNDiagram>` може також володіти колекцією елементів `<BPMNStyle>`, на які посилаються елементи `<BPMNLabel>` у діаграмі. Елемент `<BPMNLabel>` - це своєрідна мітка, яка зображує текстову інформацію про елемент BPMN. Межі визначення класу `<BPMNLabel>` завжди належить класу `<BPMNDiagram>` та відносяться до точки початку процесу з координатами X і Y – це положення лівого верхнього кута мітки – фактично (щодо лівого верхнього кута площини) графічний елемент `<Pool>` еквівалент класу `<BPMNPlane>`.

Усі елементи класу `<BPMNEdge>`, який описує підмножину ребер BPMN-діаграми, належать безпосередньо до `<BPMNPlane>` (тобто до кореневого елемента `<BPMNDiagram>`). Параметр `Waypoints` в BPMN-онтології визначає початок обходу графу (діаграми), завжди відноситься до точки початку обходу і обов'язково позначається позитивними координатами.

В конвертованому файлі OWL клас `<BPMNEdge>` фактично складається з підмножини окремих текстових екземплярів класу, які мають унікальні імена для визначення кожного окремого ребра діаграми.

На рисунку 2.9 наведено фрагмент онтографу конвертованого XPDL-файлу сценарію, який показує взаємодію класу `<BPMNEdge>` з його дочірніми

класами $\langle \text{BPMNPlane} \rangle$, $\langle \text{BPMNShape} \rangle$ і $\langle \text{BPMNEdge} \rangle$ при опису ребер спрямованого графа $Gr(Bp_N)$ для окремих пулів $\langle \text{Аналітик} \rangle$ і $\langle \text{Пошукова система} \rangle$.

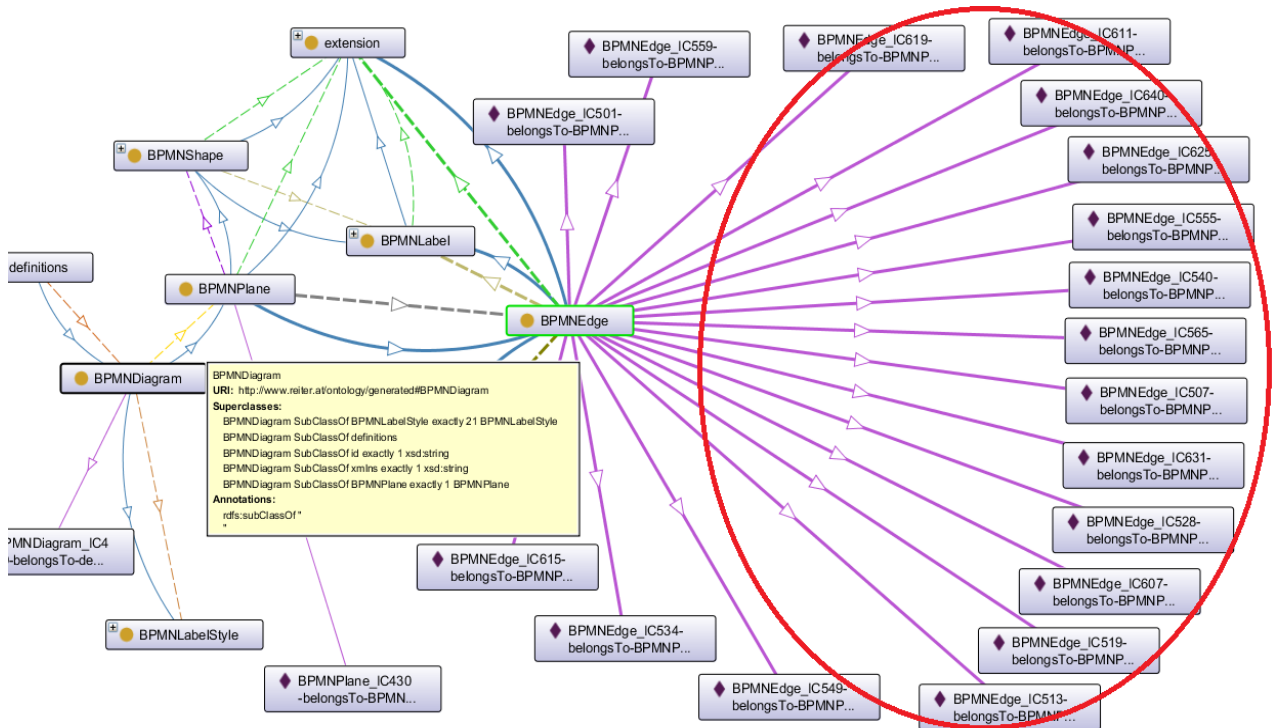


Рисунок 2.9 – Фрагмент онтографу конвертованого XPDL-файлу сценарію

З рисунку 2.9 видно як спрямований граф (BPMN-діаграма сценарію) в OWL-файлі описується через підмножину (примірників) класу $\langle \text{BPMNEdge} \rangle$, який описує ребра графу. Причому, кожний примірник позначається унікальним ім'ям, який визначає зв'язок з реальними вузлами графу (елементами BPMN-діаграми), наприклад,

BPMNEdge_IC501-belongsTo-BPMNPlane_IC430.

Подібним чином описуються всі ребра спрямованого графу для кожного окремого пулу.

Характеристика зв'язку між вузлами спрямованого графу визначається через клас $\langle \text{SequenceFlow} \rangle$ OWL-моделі, якій співвідноситься з графічним елементом BPMN «Потік управління» (Sequence flow) та позначається відповідним унікальним номером ідентифікатора.

На рисунку 2.10 на прикладі екземпляру *BPMNEdge_IC501-belongsTo-BPMNPlane_IC430* показано, яким чином для цього екземпляру класу `<BPMNEdge>` задаються параметри для визначення його ID-ідентифікатору.

В подальшому ID-ідентифікатори екземплярів класів використовується для оцінки якості перетворення BPMN-моделі в відповідний файл OWL

The screenshot displays the OWL editor interface. On the left, the 'Class hierarchy: BPMNShape' is shown, with 'BPMNShape' selected. The main area shows the 'Individual Annotations' for 'BPMNShape_IC483-belongsTo-BPMNPlane_IC430'. A red circle highlights the 'id' property value: 'DiagramElement_34d33c3a-3bfe-4500-9a74-f7bd6be02260'. Below, the 'Individuals by type' list shows 'BPMNShape (18)' with the selected instance highlighted. The right panel shows the 'Description' and 'Property assertions' for the selected instance.

Рисунок 2.10 – Приклад завдання параметрів для визначення ID-ідентифікатору екземпляра класу `<BPMNEdge>`

Аналогічним чином визначається підмножина екземплярів класу `<BPMNShape>`, завдяки яким описуються всі графічні елементи BPMN-діаграми сценарію, що складають пули `<Аналітик>` і `<Пошукова система>`.

На рисунку 2.11 наведено приклад застосування механізму змістовного опису параметрів екземпляра класу `<BPMNShape>` для завдання графічного елемента типу `<DataStore>` з ім'ям «База знань».

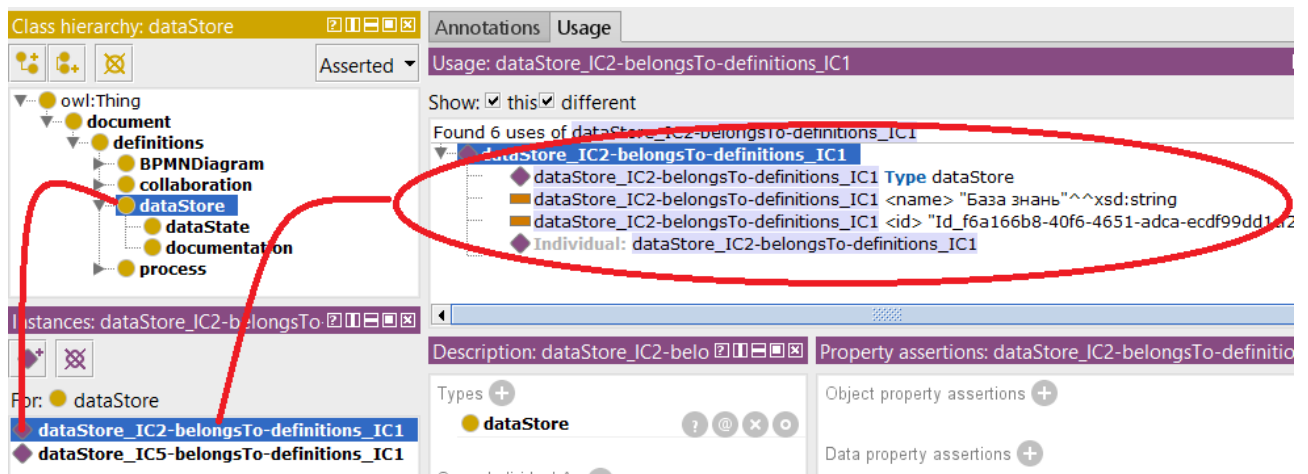


Рисунок 2.11 – Приклад застосування механізму змістовного опису параметрів екземпляра класу

Проведена оцінка якості перетворення BPMN-моделі в відповідний файл OWL на базі запропонованих критеріїв показала наступне:

1. Кожний графічний елемент BPMN-моделі з діаграми сценарію <Аналіз запитів> (XML-елемент у файлі XPDЛ) визначається як окремий клас OWL з унікальним ім'ям, що дає можливість його більш детального відображення та дослідження візуальними методами редакторів онтології.
2. Кожний процес сценарію <Pool> у BPMN-моделі співвідноситься з відповідним класом <BPMNPlane>, який визначає контейнер в OWL-моделі.
3. Кожний клас OWL-моделі описується певними властивостями (<Data properties> або <Object properties>), які асоційовані з відповідними XML-атрибутами у вхідному файлі XPDЛ
4. Кожен вузол спрямованого графу – $Gr(Bp_N)$ на діаграмі BPMN представлено відповідним екземпляром <BPMNEdge>
5. Всі обмеження властивостей певних класів онтології співвідносяться з обмеженнями на визначення відповідних графічних елементів BPMN-моделі сценарію та використовуються в аксіомах.

Таким чином, онтологічний аналіз результатів моделювання на прикладі сценарію <Аналітика> показав принципову можливість порівняння моделей сценаріїв АнД, створених в нотації BPMN-моделі, з конвертованою OWL-моделлю. З огляду на той факт, що в результаті конвертації ми, фактично,

отримуємо XML-файл BPMN-сценарію. Це дає принципову можливість виконання SPARQL-запитів до OWL-моделі сценарію з використанням синтаксису природної мови.

2.7 Висновки до другого розділу

1. Запропонований теоретичний підхід до моделювання сценаріїв Анд на основі графічних методів нотації BPMN і OWL-моделей показав принципову можливість порівняння моделей сценаріїв Анд, створених в нотації BPMN-моделі, з конвертованою OWL-моделлю сценарію. Завдяки конвертації BPMN-моделі, OWL-модель сценарію Анд фактично перетворюється в базу знань організації, яка може розширюватися за рахунок додавання нових сценаріїв Анд.

2. Отримана OWL-модель сценарію Анд може досліджуватися в інструментальному середовищі будь-якого зі стандартних редакторів онтології, зокрема Protégé 5. Залучаючи Web-технології, розробник сценарію Анд має можливість більш детально візуалізувати OWL-моделі, а також проводити семантичний аналіз коректності логічно-пов'язаних фрагментів – переходів сценарію (особливо багаторівневих) використовуючи інструментарій редакторів онтології.

3. Запропонований метод перетворення BPMN-моделі в OWL-модель сценарію дає можливість задіяти семантичне пояснення та інтерпретацію аналітичних процесів в процесі моделювання сценаріїв Анд. Суть метода в тому, що на першому етапі BPMN-модель має бути конвертована в OWL-файл (у форматі XML), якій включає такий набір сутностей онтології, що співвідноситься з усіма графічними елементами діаграми сценарію в нотації BPMN в співвідношенні 1:1. Це вирішується на другому етапі методу за рахунок побудови тривірневої моделі онтології отриманих BPMN-моделей для кожного концепту, що є розвитком положення теорії побудови моделі багаторівневої онтології. Завдяки технології конвертації BPMN-моделі в онтологію, OWL-модель сценарію Анд, фактично, перетворюється в базу знань, яка може розширюватися за рахунок додавання нових сценаріїв Анд.

РОЗДІЛ 3. ІНТЕЛЕКТУАЛІЗАЦІЯ ПРОЦЕСУ ФОРМУВАННЯ СЦЕНАРІЮ АНАЛІТИЧНОЇ ДІЯЛЬНОСТІ

3.1 Забезпечення семантичної сумісності процесів складного сценарію

Накопичування формалізованих знань про побудову сценаріїв АнД дає можливість полегшити процес моделювання складних сценаріїв АнД та зменшити його час за рахунок його інтелектуалізації. Одним із проявів інтелектуалізації є накопичення знань о діях користувача-аналітика. В цьому контексті сенс накопичення знань полягає в тому, щоб програмний засіб був в змозі самостійно класифікувати нові дані, одержувані при аналізі, і пропонувати користувачу найбільш доцільні кроки сценарію, виходячи з накопичених знань та ситуації що склалася на попередніх кроках. Але при цьому у контексті виконання багаторівневих сценаріїв АнД, одним із ключових питань залишається вирішення задачі забезпечення семантичної сумісності процесів, з яких складається будь-який складний сценарій АнД.

Сумісність є однією з основних проблем галузі програмної інженерії, що зумовлена зростаючою потребою складання динамічно змінюваних процесів у різних сферах діяльності людини. Ця динаміка часто ускладнюється відмінностями семантики взаємодії програмних компонентів. Щоб усунути ці відмінності без зміни програмних компонентів, потрібні посередники, які систематично забезпечують сумісність між функціонально сумісними компонентами шляхом відображення їх інтерфейсів та координації їх поведінки [9].

У контексті виконання багаторівневих сценаріїв АнД, *семантична сумісність* процесів, з яких складається будь-який складний сценарій АнД, теж залишається одним із ключових питань, яке потребує ретельного вивчення. Для цього пропонується підхід для вирішення сумісності процесів (особливо при здійсненні переходів в багаторівневому сценарії) за рахунок застосування

онтології ПрО для вирішення конфліктів, пов'язаних з невідповідностями, які виникають при взаємодії кроків сценарію.

Для визначення того, чи сумісні процеси при визначенні умов переходів, викладено та доведено набір аксіом та теорема. У разі сумісності процесів запропоновано посередницький підхід – Semantic Process Mediation (SPM) для розв'язання розбіжностей, які виникають у переходах між кроками виконання складного сценарію АнД. Цей підхід засновано на базі онтологічної моделі комунікаційного посередника – Semantic Communication Mediator (SCM), який застосовується для вирішення аналогічних проблем у сфері технологій Semantic Web Service [132, 133].

З теоретичної точки зору виконання кроків складного сценарію АнД описується двома принципово різними компонентами: даними та процесами. Семантична модель посередництва (SCM) складається з механізму вирівнювання онтології у взаємодіючих процесах та базується на кількох функціях забезпечення семантичної сумісності, які включають головним чином:

- семантичну анотацію даних для перевірки коректності логічних переходів між різними кроками сценарію. Семантична анотація даних сценарію досягається шляхом покращення вмісту, умов визначення переходів, що забезпечує взаємозв'язок між кроками сценарію, а також робить їх придатними для машинного читання;

- семантичне вирівнювання онтології (встановлення співвідношень між сутностями онтології, які виконані в різних моделях та нотаціях) з урахуванням гетерогенних виконавчих комп'ютерних моделей при взаємодії різних рівнів та умов переходів складного сценарію;

- загальну модель даних для єдиного представлення ПрО сценаріїв АнД, яка містить базові сутності онтології, включаючи опис логіки умов переходів;

– онтологічні моделі ПрО сценаріїв для підтримки процесів семантичного посередництва при вирішенні розбіжностей SCM, які можуть виникнути у переходах між кроками виконання складного сценарію АнД.

Дослідження взаємодії між кроками складного сценарію АнД, проведене в роботі, базується на таких новітніх технологіях: опису загальної онтології ПрО OWL2 [3]; онтології моделювання веб-сервісів (Web Service Modeling Ontology – WSMO) [4]; середовищі виконання веб-служб (Web Service Execution Environment – SEE) [5]; моделювання бізнес-процесів (Business Process Model and Notation – BPMN) [6]; технології перетворення BPMN-моделі в OWL-модель разом із підтримкою інструменту моделювання [7, 8]. Ці технології використовуються для вирішення розбіжностей при взаємодії кроків сценарію.

Під час виконання сценарію АнД семантичний посередник SCM здійснює автоматичний аналіз умов переходів з $S_n(Pr_i)$ кроку сценарію на наступний $S_{n+1}(Pr_j)$ крок. При цьому внутрішні рішення, прийняті всередині $S_n(Pr_i)$ кроку, у цьому випадку не мають значення, оскільки SCM під час роботи діє на рівні обміну повідомленнями (умови переходів), надісланих та отриманих під час взаємодії кроків сценарію. На попередньому кроці можуть виникнути процеси, які формують умови переходів $Cond_{S_i}$ на наступний крок. На рисунку 3.1 показано місце та функції SCM на прикладі розв'язання розбіжностей у варіантах переходу між двома кроками сценарію. Стандартною є ситуація, коли в залежності від умов попереднього кроку, SCM повинен вирішувати завдання вибору переходу до певного процесу $S_{n+1}(Pr_j)$ – стану сценарію.

Для відображення взаємозв'язку між двома процесами, тобто переходу між кроками сценарію $S_n(Pr_1)$ та $S_{n+1}(Pr_2)$, використовуються таке позначення:

$$\frac{Cond_{S_i} \text{ sequence of } S_n(Pr_i)}{Cond_{S_j} \text{ sequence of } S_{n+1}(Pr_j)}$$

Якщо перехід між двома кроками має більше двох процесів (станів), як це показано на рисунку 3.1, для відображення зв'язку між кроками сценарію можна використовувати кілька пар позначень за кількістю можливих станів та умов переходів:

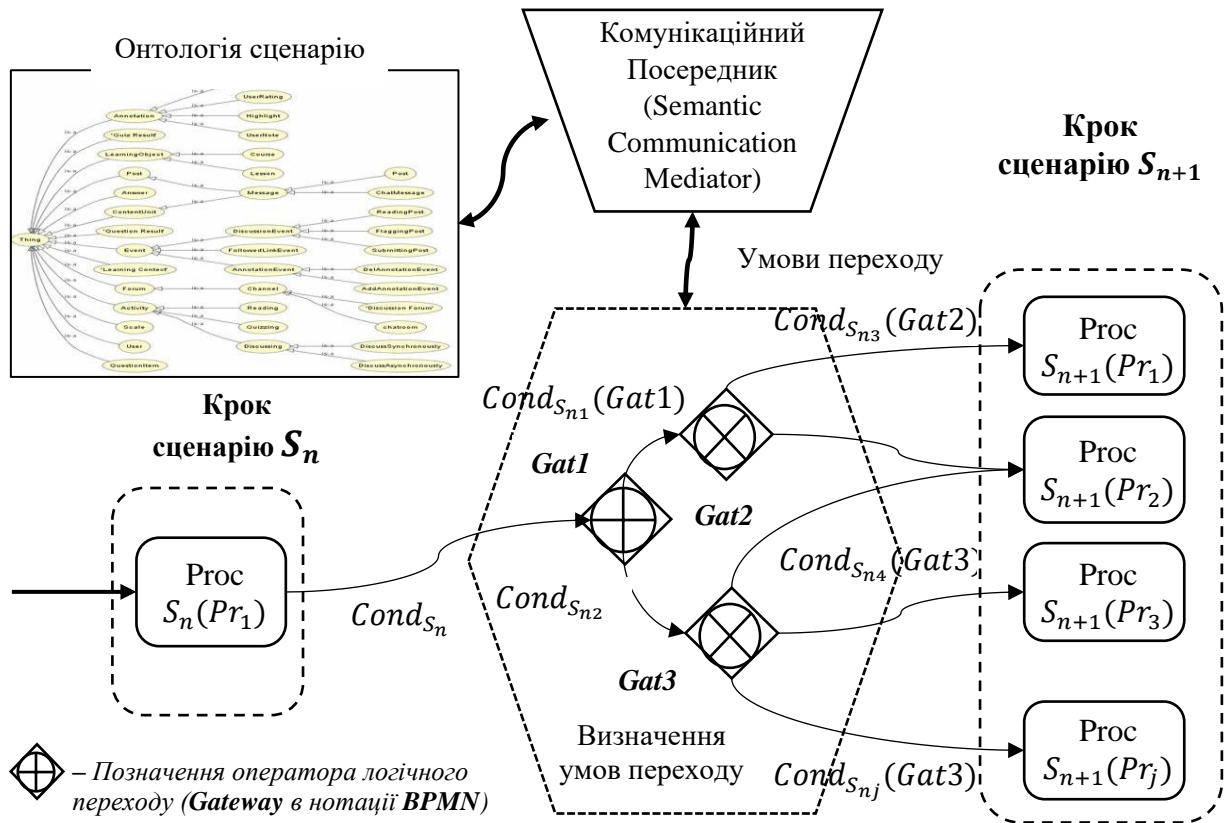


Рисунок 3.1 – Місце семантичного посередника SCM при розв’язанні розбіжностей в складному переході

$$\frac{Cond_{S_i} \text{ sequence of } S_n(Pr_1) \text{ in interaction with } S_{n+1}(Pr_1)}{Cond_{S_i} \text{ sequence of } S_{n+1}(Pr_1) \text{ in interaction with } S_n(Pr_1)}$$

$$\frac{Cond_{S_i} \text{ sequence of } S_n(Pr_1) \text{ in interaction with } S_{n+1}(Pr_2)}{Cond_{S_i} \text{ sequence of } S_{n+1}(Pr_2) \text{ in interaction with } S_n(Pr_1)}$$

$$\frac{Cond_{S_i} \text{ sequence of } S_n(Pr_1) \text{ in interaction with } S_{n+1}(Pr_j)}{Cond_{S_i} \text{ sequence of } S_{n+1}(Pr_j) \text{ in interaction with } S_n(Pr_1)}$$

$$\frac{Cond_{S_i} \text{ sequence of } S_{n+1}(Pr_1) \text{ in interaction with } S_n(Pr_1)}{Cond_{S_i} \text{ sequence of } S_{n+1}(Pr_2) \text{ in interaction with } S_n(Pr_1)}$$

• • •

$$\frac{Cond_{S_i} \text{ sequence of } S_n(Pr_1) \text{ in interaction with } S_{n+1}(Pr_j)}{Cond_{S_i} \text{ sequence of } S_{n+1}(Pr_j) \text{ in interaction with } S_n(Pr_1)}$$

$$\frac{Cond_{S_i} \text{ sequence of } S_{n+1}(Pr_1) \text{ in interaction with } S_n(Pr_1)}{Cond_{S_i} \text{ sequence of } S_{n+1}(Pr_j) \text{ in interaction with } S_n(Pr_1)}$$

Порядок послідовності умов переходів зображується символом \rightarrow . Наприклад, процес кроку сценарію $S_n(Pr_1)$ взаємодіє з процесом $S_{n+1}(Pr_2)$, якщо виконуються наступні умови:

If $S_n(Pr_1) = S_{n+1}(Pr_2)$

де

$$S_{n+1}(Pr_1) = Cond_{S_i} \rightarrow Cond_{S_{n1}}(Gat1)^1 \rightarrow Cond_{S_{n1}}(Gat2)^2$$

та

$$S_{n+1}(Pr_1) = Cond_{S_i} \rightarrow Cond_{S_{n1}}(Gat1)^3 \rightarrow Cond_{S_{n1}}(Gat3)^1$$

Перехід між двома кроками складного сценарію зазвичай складається з багатьох умов, тому пошук еквівалентності у шаблонах умов переходів до наступних кроків – це зовсім не тривіальне завдання. Інтуїтивно найпростіший спосіб це спочатку визначити невідповідності, а потім шукати спосіб їх усунення.

Дослідження показують, що в процесі обробки умов переходів класифікуються три можливі випадки:

1. **Повна відповідність** – коли обидва процеси мають абсолютно однаковий шаблон переходу за визначеними умовами. В цьому випадку взаємодія між процесами сценарію може відбуватися автоматично з мінімальним втручанням SCM в роботу сценарію.

2. **Вирішені невідповідності** – коли взаємодіючі процеси мають різні шаблони переходу за умовами, і для їх подолання можна виконати кілька перетворень, які посередник SCM може подолати самостійно, за рахунок онтологічної бази знань прогнозованих дій.

3. **Невирішені невідповідності** – які складаються або з невідповідності моделей процесів між двома кроками сценарію, або невідповідності процесу комунікації (взаємозв'язку), яку неможливо вирішити будь-якими перетвореннями в послідовностях обміну повідомленнями $Cond_{S_i}$ під час виконання переходу між кроками сценарію.

У випадку «*Вирішені невідповідності*» можна класифікувати типові ситуації, які можуть бути автоматично вирішені семантичним посередником SCM з використанням онтологічного опису умов переходів сценарію.

1. Процес $S_n(Pr_1)$ надсилає повідомлення $Cond_{S_i}$, тоді як процес $S_{n+1}(Pr_1)$ не готовий його отримувати.

2. Процес $S_n(Pr_1)$ надсилає одразу кілька екземплярів повідомлень $Cond_{S_i}^{e1}$, $Cond_{S_i}^{e2}$, $Cond_{S_i}^{en}$, тоді як процес $S_{n+1}(Pr_1)$ хоче отримати цю інформацію як частину кількох повідомлень.

3. Процес $S_n(Pr_1)$ надсилає повідомлення в порядку $Cond_{S_i}^{e1}$, $Cond_{S_i}^{e2}$, $Cond_{S_i}^{en}$ іншому порядку, ніж їх очікує процес $S_{n+1}(Pr_1)$. Порядок надісланих повідомлень може бути автоматично інвертований до проекції, визначеної у відповідних аксіомах та лемах, які виведено у дисертаційній роботі.

4. Процес $S_n(Pr_1)$ надсилає повідомлення $Cond_{S_i}$, яке отримує процес $S_{n+1}(Pr_1)$, і очікує підтвердження того, що повідомлення було отримано цільовим процесом.

Існує набір невідповідностей, які посередник процесу SCM не може автоматично вирішити. Ці розбіжності стосуються неіснуючих повідомлень, які не відповідають семантичним умовам переходу. Наприклад, процес $S_{n+1}(Pr_1)$ очікує повідомлення $Cond_{S_i}$, яке не було надіслано процесом $S_n(Pr_1)$ до певного часу. Інший випадок розглядає ситуацію, коли процес $S_{n+1}(Pr_1)$ отримав повідомлення $Cond_{S_i}$, але процес $S_n(Pr_1)$ очікує підтвердження його отримання.

3.1.1 Теоретичне обґрунтування вирішення невідповідностей при взаємодії кроків сценарію

Складні невідповідності. В реальності взаємодія між кроками сценарію складається не лише з елементарних невідповідностей, представлених раніше, але може включати й проекції процесів, які за певних умов розглядаються як семантичні еквіваленти процесів – проекції перетворення. Метою отримання проекцій є формалізація умов переходів для більш коректного застосування

онтології сценарію при вирішенні невідповідностей з залученням семантичного посередника SCM. Для цього досліджується, як семантична сумісність між двома прогнозами впливає на сумісність прогнозованих процесів.

Для застосування онтологічних моделей при вирішенні невідповідностей в умовах переходів між кроками сценарію дамо кілька принципових визначень.

Визначення 1. Проекція процесу $S_n(Pr_i)$, що позначається через $\pi(S_n(Pr_i))$, - це похідний процес, отриманий із $S_n(Pr_i)$ шляхом застосування набору операцій до послідовності повідомлень, у такий спосіб, щоб підтримувалися мета та семантика процесу.

Це означає, що $S_n(Pr_i)$ та $\pi(S_n(Pr_i))$ відрізняються лише послідовністю розгляду умов переходів на певний крок сценарію, яка не порушує семантичну складову цих умов.

Визначення 2. Зв'язок між двома процесами $S_n(Pr_i)$ та $S_{n+1}(Pr_j)$ сценарію вважається семантично еквівалентним проекції $\pi(S_n(Pr_i))$ та процесу $S_{n+1}(Pr_j)$, якщо ця проекція жодним чином не впливає на комунікацію з точки зору початку функціонування кроку сценарію $S_{n+1}(Pr_j)$.

Визначення 3. Існує семантична відповідність між двома процесами, якщо зв'язок між ними може бути представлено у вигляді послідовності значень умов переходів цих процесів. Семантичну відповідність процесів, на рівні визначення умови переходу між кроками сценарію, позначимо як

$$Match(S_n(Pr_i), S_{n+1}(Pr_j))$$

Визначення 4. Два процеси вважаються сумісними (семантично зв'язаними), якщо між ними є семантична сумісність, або якщо кожна невідповідність може бути вирішена на рівні зміни розгляду умов переходів. Безумовно, це положення вірне, якщо зміни розгляду умов переходів не порушують семантику процесу в цілому. Позначення сумісності двох процесів $S_n(Pr_i)$ та $S_{n+1}(Pr_j)$ таке:

$$Compatible(S_n(Pr_i), S_{n+1}(Pr_j))$$

Виходячи з наданих визначень, умови існування сумісності - *Compatible* та семантичної відповідності - *Match* між процесами, які визначають переходи між кроками сценарію, можна записати у вигляді:

$$\forall S_n(Pr_i), S_{n+1}(Pr_j)$$

$$Match(S_n(Pr_i), S_{n+1}(Pr_j)) \Leftrightarrow Match(S_{n+1}(Pr_j), S_n(Pr_i))$$

та

$$\forall S_n(Pr_i), S_{n+1}(Pr_j)$$

$$Compatible(S_n(Pr_i), S_{n+1}(Pr_j)) \Leftrightarrow Compatible(S_{n+1}(Pr_j), S_n(Pr_i))$$

Ці вирази відображують симетричність (\Leftrightarrow) написання умов існування сумісності та семантичної відповідності між процесами.

Теорема. *Будь-які два процеси сценарію $S_n(Pr_i)$ та $S_{n+1}(Pr_j)$ вважаються сумісними (семантично зв'язаними) тоді і тільки тоді, коли їх існуючі проєкції $(\pi(S_n(Pr_i)))^k$ та $(\pi(S_{n+1}(Pr_j)))^f$ відповідних процесів сценарію, теж повністю відповідні, де $1 \leq i \leq k$ та $1 \leq j \leq f$:*

$$\forall S_n(Pr_i), S_{n+1}(Pr_j), Compatible(S_n(Pr_i), S_{n+1}(Pr_j)) \Leftrightarrow$$

$$\exists k, f, Match(\pi(S_n(Pr_1))^k, \pi(S_{n+1}(Pr_1))^f)$$

Де k і f - кількість можливих проєкцій перетворення $(\pi(S_n(Pr_i)))^k$ та $(\pi(S_{n+1}(Pr_j)))^f$ для процесів $S_n(Pr_i)$ та $S_{n+1}(Pr_j)$, відповідно;

$(\pi(S_n(Pr_1)))^k$ та $(\pi(S_{n+1}(Pr_1)))^f$ - позначення проєкцій, які вважається семантичними еквівалентами відповідних процесів та можуть використовуватися для аналізу умов переходів між кроками сценарію із застосуванням онтології ПрО.

З теореми слідує, що процеси сценарію $S_n(Pr_i)$ та $S_{n+1}(Pr_j)$ вважаються сумісними, якщо хоча би одна пара проєкцій $(\pi(S_n(Pr_i)))^k$ та $(\pi(S_{n+1}(Pr_j)))^f$ є відповідною – *Match*.

Демонстрація застосування теореми при вирішенні невідповідностей при взаємодії кроків сценарію передбачає доведення положення теореми в обох

напрямах, тобто вирішення невідповідностей семантичним посередником SCM при переході можливо тільки за умов:

$$\forall S_n(Pr_i), S_{n+1}(Pr_j), Compatible(S_n(Pr_i), S_{n+1}(Pr_j)) \\ \Rightarrow \exists k, f Match(\pi(S_n(Pr_i))^k, \pi(S_{n+1}(Pr_j))^f)$$

та

$$\exists k, f Match(\pi(S_n(Pr_i))^k, \pi(S_{n+1}(Pr_j))^f) \Rightarrow \\ \forall S_n(Pr_i), S_{n+1}(Pr_j), Compatible(S_n(Pr_i), S_{n+1}(Pr_j))$$

Одним з методів доведення цього твердження є рівноважність процесів та їх еквівалентних проєкцій, яка доводиться шляхом застосування принципу суперечності та індукції, з припущення, що з твердження $\exists k, f, match(\pi(S_n(Pr_i))^k, \pi(S_{n+1}(Pr_j))^f)$ випливає наступне:

$$\exists k, f, Match(\pi(S_n(Pr_i))^k, \pi(S_{n+1}(Pr_j))^f) \Leftrightarrow \\ \forall S_n(Pr_i), S_{n+1}(Pr_j), \neg Match(S_n(Pr_i), S_{n+1}(Pr_j)).$$

та

$$\forall S_n(Pr_i), S_{n+1}(Pr_j), \neg Match(S_n(Pr_i), S_{n+1}(Pr_j)) \\ \Rightarrow \neg Compatible(\pi(S_n(Pr_i))^k, \pi(S_{n+1}(Pr_j))^f)$$

Доведення цих положень виконується методом вичерпання, тобто розглядаються всі можливі випадки, кожний з яких аналізується на предмет відповідності положень теореми. Враховуючи специфіку ПрО було розглянуто 7 можливих випадків, які випадають на k, f – кількість варіантів створення проєкції процесів переходів, які мають семантичний сенс в умовах переходу.

Теорема представляє більш офіційне визначення сумісності процесів та семантичної відповідності, які використовуються посередником SCM для вирішення невідповідностей при взаємодії кроків сценарію Анд.

Щоб дозволити функціонально сумісним процесам сценарію взаємодіяти, семантичний посередник SCM повинен не тільки вирішувати відмінності між їхніми інтерфейсами, але й координувати їх поведінку, щоб забезпечити їх правильну взаємодію. Це положення можна позначити таким чином

Визначення 5. Якщо взаємодія між процесами $S_n(Pr_i)$ та $S_{n+1}(Pr_j)$ вкинується за допомогою посередника M^{SCM} процеси вважаються семантично сумісними через посередника M^{SCM} .

$$Compatible(S_n(Pr_i) \Leftrightarrow M^{SCM} [S_{n+1}(Pr_j)])$$

Це означає, що сумісність між процесами сценарію $S_n(Pr_i)$ та $S_{n+1}(Pr_j)$ через посередника M^{SCM} дає можливість застосування всього апарату логіки виводу при вирішенні невідповідностей за рахунок підключення онтології з боку посередника.

На виконавчому рівні має бути забезпечена синхронізація процесів та даних між посередником та обома кроками сценарію. Така синхронізація дозволяє забезпечити успішне завершення взаємодії кожним процесом, досягнувши стану END.

3.1.2 Типовий алгоритм встановлення взаємодії між двома процесами сценарію

Алгоритм встановлення взаємодії між процесами починається з перевірки базової конфігурації, яка описує кожний з цих процесів з позиції послідовності та семантики обміну повідомленнями. Взаємодія вважається повністю встановленою, якщо обидва процеси досягають своїх кінцевих станів. В алгоритмі це означає досягнення посередником M^{SCM} стану END (Рядки 1-3). Далі алгоритм створює екземпляри посередника M^{SCM1} та M^{SCM2} для кожного з процесів для завантаження онтології з умовами виконання взаємодії між процесами. В результаті успішного виконання алгоритм дозволяє забезпечити завершення взаємодії кожним процесом, досягнувши стану END.

Алгоритм встановлення взаємодії між двома процесами сценарію представлено нижче:

Require: $S_n(Pr_i)$, $S_{n+1}(Pr_j)$

Ensure: A Semantic mediator M^{SCM}

if $S_n(Pr_i) = \text{END}$ and $S_{n+1}(Pr_j) = \text{END}$ **then**

return END

end if

$M^{SCM} \leftarrow \text{END}$

for all $S_n(Pr_i) \rightarrow \pi(S_n(Pr_i))^k, k = 1, 2$ **do**

$mappingList \leftarrow FindEligibleMapping(Cond_{S_i}, S_n(Pr_i), \pi(S_n(Pr_i))^k)$

while $\neg found$ and $mappingList \neq 0$ **do**

$Map(Cond_{S_n}, Cond_{S_{n+1}}) \leftarrow selectMapping(mappingList)$

such that $S_n(Pr_1) \xrightarrow{Cond_{S_n}} \pi(S_n(Pr_1))^k$ and $S_{n+1}(Pr_1) \xrightarrow{Cond_{S_{n+1}}} \pi(S_{n+1}(Pr_1))^f$

$M^{SCM1} \leftarrow SynthesiseMediator(\text{for proc } S_n(Pr_i))$

if $M^{SCM1} \neq fail$ **then**

$found \leftarrow true$

$M^{SCM2} \leftarrow SynthesiseMediator(\text{for proc } S_{n+1}(Pr_i))$

if $M^{SCM2} \neq fail$ **then**

$found \leftarrow true$

end if

$M^{SCM} \leftarrow M^{SCM1} | M^{SCM2}$

end for

return M^{SCM}

Застосовуючи наведені визначення та теорему, яка формалізує умови взаємодії між процесами, можна побудувати семантичний посередник SCM, призначений для усунення певних типів невідповідностей, а саме:

– *припинення несподіваного повідомлення*. Семантичний посередник SCM аналізує процеси переходу та визначає, чи справді надіслана інформація очікується наступним кроком. Якщо ця інформація не очікується, вона не зберігається у внутрішньому сховищі та не потрапляє для подальшого розгляду.

– *розділення повідомлення*. Семантичний посередник SCM розділяє повідомлення, аналізує семантичний опис цільового процесу, визначає у якому форматі має бути передана ця інформація, готує її та надсилає отримувачу.

– *профілювання послідовності команд*. Семантичний посередник SCM аналізує умови переходів, визначає порядок їх надходження та змінює його, якщо це не впливає на загальну семантику процесу.

3.2 Парадигма інтелектуалізації кроків сценаріїв в процесі аналітичної діяльності

Вирішення задачі забезпечення семантичної сумісності процесів моделювання кроків сценарію АнД дає можливість перейти до вирішення завдання інтелектуалізації процесу побудови сценарію АнД.

Одною з головних вимог до сучасних АНС є комфортність самого процесу спілкування з ПС за рахунок інтелектуалізації засобів моніторингу дій аналітика та реалізації ПЗ аналітики. Тобто, аналітик сподівається на дружній інтерфейс системи, який може пропонувати йому найбільш ймовірній крок сценарію, виходячи з аналізу попередніх дії (кроків сценарію) та накопичених знань. При цьому для зменшення часу та підвищення рівня її ефективності з урахуванням зростаючої складності самих процесів аналізу даних постає задача автоматизації процесу формування сценаріїв та його адаптації, виходячи з парадигми інтелектуалізації ПЗ. Одним з підходів вирішення цієї задачі може бути методи машинного навчання.

Сценарій АнД, виходячи з парадигми інтелектуалізації ПЗ, можна розглядати як певну структуру представлення знань, яка використовується для опису послідовності пов'язаних подій – у вигляді орграфа. В цьому підході орграф фактично визначає сукупність шляхів досягнення мети в конкретній стереотипній ситуації (pattern) для заданої ПрО, представленої у вигляді семантичної мережі, наприклад, онтології [4]. Тобто, для генерування сценарію АнД з елементами інтелектуалізації має формуватися орієнтований ациклічний граф можливих операцій сценарію (Directed Acyclic Graph – DAG). Слід нагадати, що DAG відображає припущення про взаємозв'язок між змінними – вузлами в контексті побудованого графу, в якому відсутні орієнтовані цикли [134]. При такому підході задача побудови сценарію АнД фактично зосереджується на вирішенні відомої задачі пошуку найкоротшого шляху (Single Source Shortest Path) [135]. Отже, пошук найкоротшого шляху в орграфі від витoku до його стоку з урахуванням накопичених знань про стереотипні дії користувача можна розглядати як основу для інтелектуалізації процесу

формування сценарію Анд. Враховуючи тенденції щодо інтелектуалізації програмних засобів, така постановка задачі й досі є дуже актуальною не тільки для інтелектуалізації процесу формування сценарію Анд, але й для вирішення певного кола завдань.

Роль та місце інтелектуальних програмних засобів при виконанні сценаріїв Анд показано на рисунку 3.2. Отже, це спеціалізовані програмні рішення, які містять в собі необхідні інструменти для здійснення моніторингу за діями користувачів, засоби консолідації інформації в сховище даних, витягу, перетворення, трансформації даних з метою їх адаптації до алгоритмів data

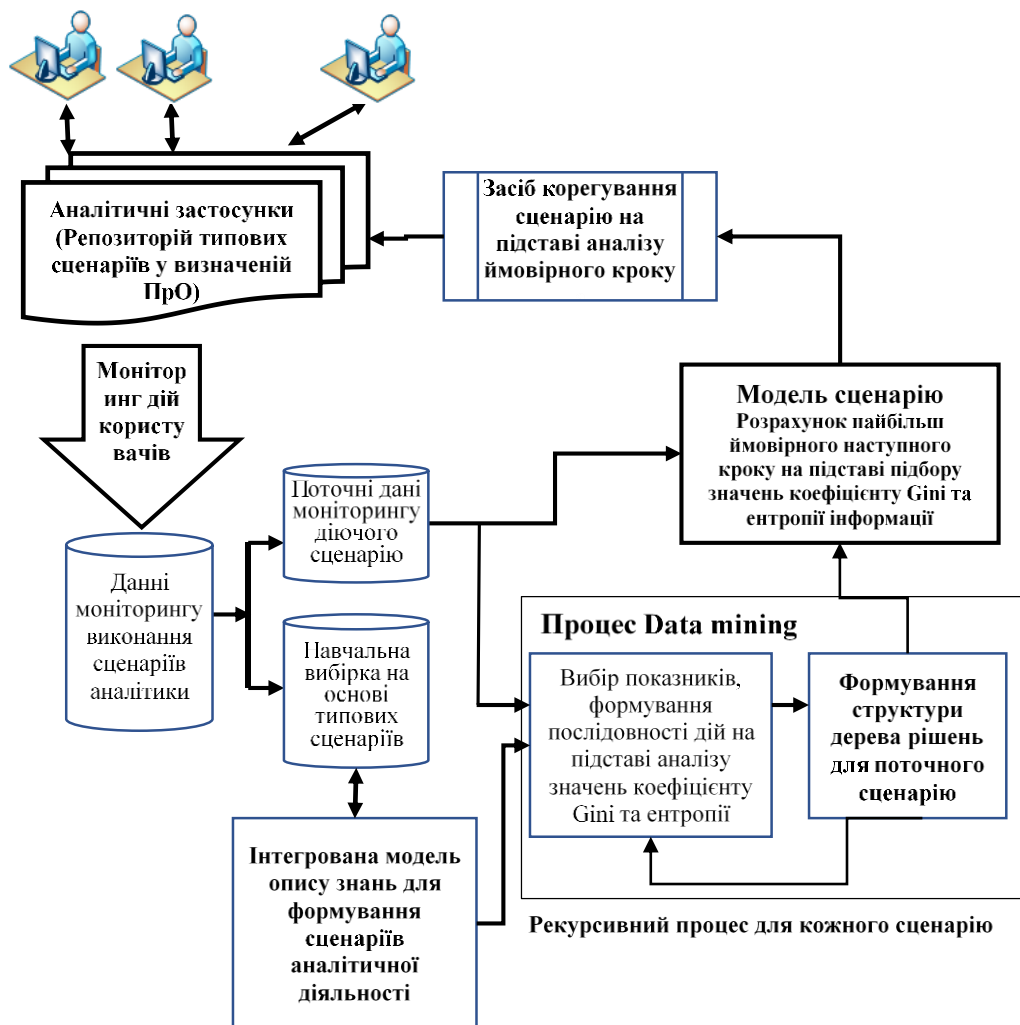


Рисунок 3.2 – Структурна схема інтелектуальних програмних засобів при виконанні сценаріїв аналітичної діяльності.

mining (зокрема, методів машинного навчання деревами класифікації і регресії – Classification and Regression Trees), а також засоби візуалізації результатів та можливості корегування сценаріїв на підставі аналізу попередніх кроків.

3.3 Інтегрована модель опису знань для побудови сценаріїв аналітичної діяльності

При створенні АНС, заснованої на знаннях, саме побудова її моделі БЗ є найскладнішим та відповідальним процесом. Перш за все, необхідно чітко розмежовувати природу знань про ПрО (особливо якщо вона відноситься до багатофакторних систем), тобто якою моделлю найліпшим образом описуються знання ПрО: реляційна, фреймова, об'єктно-орієнтована, семантична-мережа, гіпертекстова, онтологічна або інші моделі.

Так, семантичні мережі [79, 122, 123] являють собою інформаційну модель предметної області, що має вигляд орієнтованого графа, вершини якого відповідають об'єктам ПрО, а ребра задають відносини між ними. Об'єктами можуть бути поняття, події, властивості, процеси використовуються для зберігання інформації, в термінах об'єктів і відносин між ними.

Продукційні системи [66] розглядаються як засіб подання знань експертів у вигляді правил виду «ЯКЩО-ТО» і організації логічного висновку.

Фреймові моделі [40, 151] представляють об'єктно-орієнтований підхід та служать як для підвищення рівня представлення знань, так і для забезпечення можливості спільного використання декларативних і процедурних знань. До першого типу відносяться знання, що описують простір розв'язуваних в системі задач, включаючи розбиття задач на підзадачі і зв'язок підзадач з методами їх вирішення, до другого - знання, що представляють як методи вирішення завдань, так і конкретні алгоритми виведення.

Для опису об'єктно-орієнтованої моделі знань мають бути визначені основні сутності ПрО, а також знання про те, яким чином ці сутності пов'язані між собою [51 – 53, 95 - 97]. Тобто, ця модель також визначає знання про відносини, які безпосередньо пов'язують поняття, різного роду залежності між

поняттями ПрО, що описуються як на логічному, так і на функціональному рівнях.

Крім того, для багатьох ПрО досить часто виникає необхідність використовувати знання, що мають неточні значення атрибутів - "fuzzy logic" [206]. Отже, для таких знань обов'язково мають використовуватися методи та процедури опису fuzzy-значень. Тому адекватним засобом формалізації онтології можуть служити моделі на базі лінгвістичних змінних, нечіткі множини, нечіткі відносини, нечіткі графи і нечіткі дерева, нечіткі обмеження. Тобто для подання fuzzy-знань в моделі ПрО необхідно передбачити механізми встановлення в атрибутах невизначені значення, а також завдання обмежень на значення атрибутів, що зв'язують такі атрибути.

Також доволі часто виникає потреба в поданні знань у вигляді обмежень на значення параметрів, що описують деякий об'єкт або систему об'єктів. Бази знань АНС також повинні містити конкретні знання, які подаються у вигляді примірників (екземплярів конкретних об'єктів) і зв'язків між ними або обмежень, заданих на значеннях атрибутів екземплярів понять.

Крім знань, які постійно присутні в БЗ, необхідно представляти знання, що описують фрагмент дійсності (ситуацію), який задає контекст і вхідні дані для задач. Такі знання, як і предметні знання, зазвичай представляються в обмеженнях, заданих на значеннях атрибутів екземплярів понять.

Аналіз сучасних тенденцій [26, 29, 84, 99, 167, 200, 238, 251, 263, 259, 265] показує, що при створенні прикладних АНС все більш актуальним є питання об'єднання різних методів опису знань, оскільки жодний окремий метод не може забезпечити в повному обсязі опис моделі знань реальної аналітичної системи. Це пояснюється тим, що моделі подання знань непрямим чином пов'язані також із типом вхідних даних (структуровані, неструктуровані, читки нечіткі, текстові, гіпертекстові та інші типи даних), які саме формують контекст для задач, що вирішуються в АНС. Тому бажано щоби інтегрована модель подання знань враховувала не тільки методи подання знань різних моделей

(онтологічної, мережевий, продукційної та інших), але й засоби формування логічних правил та процесів виведення нових знань.

У зв'язку з цим виникає об'єктивна необхідність в розробці підходу для опису інтегрованої моделі подання знань, яка б поєднувала об'єктно-орієнтовані методи опису моделей даних, з семантичним їх уявленням, достатнім для взаємодії різних моделей знань в межах створюваної аналітичної системи. Підхід засновано на парадигмі Semantic Web [122 - 125], яка передбачає публікацію знань (метаданих) на мовах, спеціально розроблених для роботи з різними форматами даних: Framework Description (RDF), Web Ontology Language (OWL) та Extensible Markup Language (XML) [106, 113, 38]. HTML описує документи та зв'язки між ними. Тобто, парадигма Semantic Web надає необхідні механізми інтеграції різних моделей опису знань за рахунок використання метаданих, що описують різні моделі даних та їх семантичної зв'язаності на основі опису складних залежностей у вигляді графу – RDF-графів та мови універсальної мови – XML.

Отже, інтегрована модель подання знань - $InM(KB)$, заснована на парадигмі Semantic Web, яка поєднує різні моделі, може бути описана математичним виразом:

$$InM(KB) = \langle Ont(SD), SNet, FScen, PRul, PCase, PRes \rangle$$

де $Ont(SD)$ – прикладна онтологія, що задає опис основних сутностей (понять і відносин) ПрО у вигляді класів об'єктів, екземплярів класів, їх властивостей та відносин між класами та властивостями, включаючи опис IP необхідних для здійснення Анд;

$SNet$ – об'єктно-орієнтована семантична мережа, яка описує властивості об'єктів та IP ПрО, а також відносини між компонентами інтегрованої моделі, використовуючи мережеві методи опису компонентів у вигляді графової моделі;

$FScen$ – модель сценаріїв Анд або функціональна модель, яка описує аналітичні функції та процедури. На базі компонентів семантичної мережи формуються виконавчі сценаріїв Анд для здійснення операцій над екземплярами

класів прикладної онтології. Модель сценаріїв АнД враховує реальні обмеження на значення атрибутів об'єктів семантичної мережі *SNet*;

PRul – модель продукційних правил, яка описує правила логічного виводу в термінах класів та відношень для здійснення операцій над екземплярами класів прикладної онтології *Ont(SD)*. Модель продукційних правил також спирається на аксіоми, які визначають несуперечні твердження ПрО для виведення на основі дескриптивної логіки, або множину прецедентів для нечіткого виведення.

Pres – модель механізмів виведення нових знань (ПЗ логічного виведення) та обробки даних. Для сучасних АнС ця модель повинна містити відповідні механізми як чіткого виведення (на основі правил та аксіом), так і механізми виведення на основі нечітких правил, які маніпулюють з лінгвістичними змінними;

PCase – виконавча модель ПЗ аналітичної діяльності, яка складається з окремих програмних блоків та компонентів, призначених для реалізації сценаріїв, множині відповідних аналітичних функцій, процедур обробки даних та виводу нових знань.

За базис інтегрованої моделі подання знань - *InM(KB)* запропоновано використовувати модель прикладної онтології *Ont(SD)*, яка є основним засобом опису як предметної області, так і програмного забезпечення, яке її реалізує. Така онтологія визначає головні сутності (класи об'єктів, властивості і відносини між ними), які потім використовуються іншими засобами моделі. Формально прикладна онтологія предметної області може бути представлена у вигляді математичного виразу:

$$\mathbf{Ont(SD)} = \langle \mathbf{C}^{(SD)}, \mathbf{R}^{(In)}, \mathbf{Rb}^{(C)}, \mathbf{T}^{(SD)}, \mathbf{Ac}^{(C)}, \mathbf{Cntr}, \mathbf{Rul}^{(S)} \rangle$$

де $\mathbf{C}^{(SD)} = \{ \mathbf{C}^{(SD1)}, \dots, \mathbf{C}^{(SDm)} \}$ – множина класів, яка описує основні поняття об'єктів предметної області та програмного забезпечення, що реалізує створювану аналітичну систему;

$R^{(In)}$ де $R^{(In)} \subseteq C^{(SD)} \times C^{(SD1)}$ – строгий частковий порядок на множині класів $C^{(SD)}$, що задає відношення успадкування;

$Rb^{(C)}$ де $Rb^{(C)} = Rb^{(C1)}, \dots, Rb^{(Cn)}$, а в свою чергу $R^{(Ci)} \subseteq C^{(SD)} \times C^{(SD1)}$ – кінцева множина бінарних відносин, які встановлюється між двома елементами множини заданих на $C^{(SD)}$ класах онтології;

$T^{(SD)} = \{ T^{(SD1)}, \dots, T^{(SDt)} \}$ - множина типів даних, що притаманні ПрО;

$Ac^{(C)} = \{ a_1^{(Ci)}, \dots, a_r^{(Ci)} \}$ – кінцева множина атрибутів, що описують властивості класів $C^{(SD)}$ предметної області;

$Cntr$ – обмеження, що задаються на атрибутах класів $Ac^{(C)}$, які записуються логічними виразами вигляду:

$Cntr(Cn_{i1}, \dots, Cn_{iw})$ де $Cn_{ik} \in Ac^{(Ci)}$ або $Cn_{ik} \in T^{(SDi)}$, тобто Cn_{ik} може бути або ім'ям атрибута, або константою, яка використовується для завдання детермінованих умов виводу знань за допомогою відповідних програм.

$Rul^{(S)}$ підмножина правил логічного виводу для отримання нових знань в ПрО завдяки машині логічного виводу - $PRes$.

Таким чином, прикладна онтологія $Ont(SD)$ задає комплексний опис понять і відносин як ПрО, так і ПЗ, що його реалізує. Цей опис здійснюється у вигляді класів об'єктів і відносин, інкапсулюючих в собі семантичні властивості та обмеження на свої атрибути. Такий опис фактично визначає структуру семантичної і функціональної мережі, за допомогою якої представляються і знання про предметну область, а також знання про функції, які доцільно виконувати при вирішенні конкретних аналітичних завдань. Слід також підкреслити, що за допомогою семантичної мережі задаються сутності ПрО, в термінах яких описуються продукційні правила логічного виводу нових знань.

Головною перевагою онтологічної моделі перед іншими моделями знань є можливість опису ієрархії відношень типу «загальне – приватне» для класів онтології – $C^{(SD)}$. Така перевага суттєво спрощує процес опису ПрО за рахунок

використання механізмів успадкування властивостей (включаючи атрибути, відносини і обмеження) вищих класів нижчими.

Завдяки включенню до інтегрованої моделі апарату лінгвістичних типів даних, все прості типи даних, що входять до множини типів даних – $T^{(SD)}$, мають лінгвістичні розширення. Це дозволяє визначати лінгвістичні значення даних у вигляді інтервалу допустимих значень та в подальшому оперувати саме з ними. Зокрема, задавати атрибутам об'єктів онтологічної моделі певні значення, які сприяють більш точному виконанню логічних виводів.

Функціональні залежності a_j між атрибутами сутностей можуть задаватися в описах класів об'єктів і відносин у вигляді обмежень $Cntr_i \in Cntr$, що зв'язують значення атрибутів об'єктів. (В описах бінарних відносин обмеження задаються на атрибутах їх аргументів.) Такі обмеження є кінцевою множиною логічних виразів, що зв'язують логічними операціями значення атрибутів об'єктів предметної області. Для опису правил логічного виведення $PRul$ використовується як модель тверджень про факти, що містить онтологія предметної області, представлені у вигляді аксіом, так і модель тверджень у вигляді «ЯКЩО-ТО». В обох випадках використовується семантична мережа понять, які визначені у термінах предметної області.

Функціональна або сценарна модель - $FScen$ займає особливе місце у складі інтегрованої моделі. Вона складається з типових сценаріїв обробки даних та інформації, представленої різними типами (структурованої, неструктурованої, слабо структурованої, текстової, Web) з метою отримання нових знань. Сценарна модель пов'язує алгоритми програмної реалізації АнС – $PCase$ з моделлю продукційних правил. У разі якщо реалізація запропонованими засобами ускладнена значними витратами або неефективна, передбачається доповнення моделі більш ефективними методами обробки даних. Взаємозв'язки інтегрованої моделі опису знань для здійснення аналітичної діяльності показані на рисунку 3.3.

Інтегрована модель дозволяє представляти завдання аналітичній системі за допомогою онтологічної моделі $Ont(SD)$, де описуються типи завдань і

методів, призначених для їх вирішення $FScen$, а за допомогою семантичної мережі $SNet$ задається розбиття задач на підзадачі і зв'язок завдань з методами їх вирішення. При цьому самі методи вирішення можуть бути представлені продукційними правилами, об'єднаними в сценарії $FScen$. Для цього в модель включаються засоби завдання продукційних правил і управління ними - $PRul$, а також оператори активації правил, які можуть поєднуватися з виконавчими засобами управління та виконуються за допомогою машин логічного виводу - $PRes$.

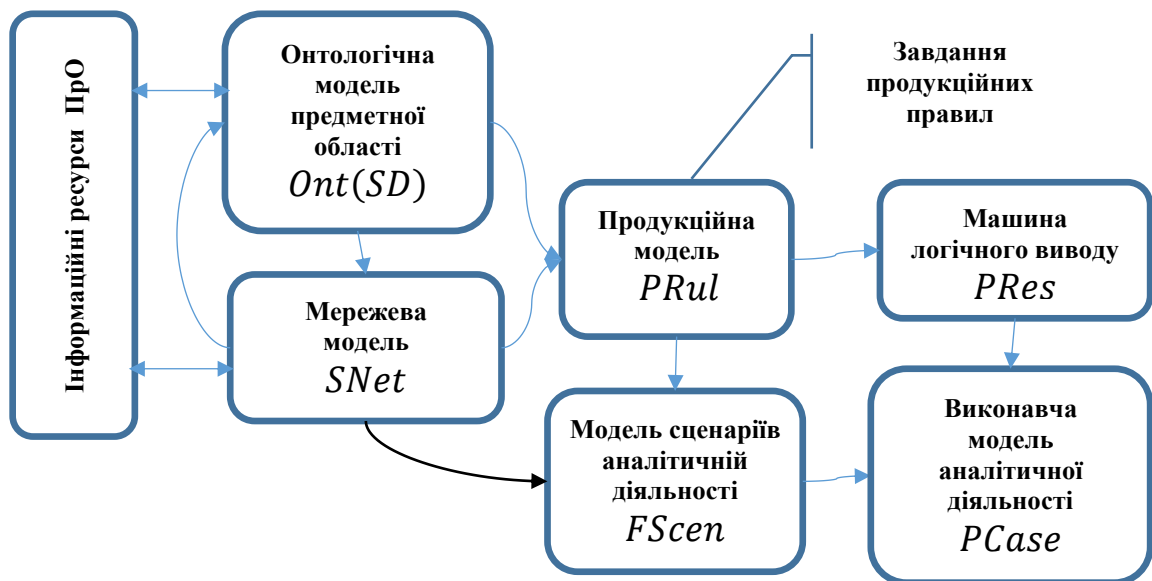


Рисунок 3.3 – Взаємозв'язки в інтегрованій моделі опису знань для здійснення аналітичної діяльності

Слід підкреслити, що функціональна (сценарна) модель - $FScen$ призначена для підтримки методів та алгоритмів програмної реалізації аналітичної системи $PCase$, у разі якщо реалізація іншими засобами ускладнена значними витратами або неефективна.

Запропонована інтегрована модель опису знань для здійснення аналітичної діяльності $InM(KB)$ поєднує об'єктно-орієнтовані підходи з семантичним уявленням про знання предметної області, що дозволяє використовувати різні моделі знань в одній аналітичній системі (онтологічні, мережеві, реляційні, продукційні) та дані, що можуть бути розташовані в

розподіленому середовищі, включаючи WWW-простір. При цьому, для універсализації опису знань:

- поняття та відносини ПрО і програмних засобів представляються класами об'єктів онтології - $Ont(SD)$, які інкапсують в собі семантичні властивості та обмеження – $Cntr_i \in Cntr$, що задаються на атрибути і аргументи класів та зв'язують значення атрибутів об'єктів;

- об'єкти (екземпляри класів) і екземпляри відносин представляються в об'єктно-орієнтованій семантичній мережі – $SNet$, структура і властивості якої визначені в моделі онтології - $Ont(SD)$;

- обмеження на значення атрибутів об'єктів представляються у функціональній моделі – $FScen$ у вигляді обмежень, які накладаються на сценарії виконання функції та процедур, що реалізують завдання системи;

- усі необхідні процеси виведення і обробки інформації здійснюються системою продукційних правил – $PRul$, яка пов'язана з семантичною мережею $SNet$ та сценаріями виконання аналітичних функцій і процедур $FScen$, що реалізують завдання аналітичної системи;

- реалізація сценарію виконання функції і процедур здійснюється через засоби програмування і механізм виводу/обробки знань та даних – $PCase$ та спеціалізовані програми (машини логічного виводу).

3.3.1 Задання ваг важливості понять онтології для задач семантичного типу

Відомо, онтологія заснована на об'єктно-орієнтованій парадигмі. Це означає що кожне поняття онтології задається як окремий клас. Для задач семантичного типу запропонуємо такий підхід до обчислення ваг важливості понять:

1 Повна вага важливості W_j^i поняття (класу) онтології дорівнює сумі власної ваги важливості $W_o_j^i$, ваг важливості підкласів $W_s_j^i$ та ваг важливості суміжних класів $W_n_j^i$ (класів, зв'язаних з цим класом не IS-A зв'язком):

$$W_j^i = W_o_j^i + W_s_j^i + W_n_j^i,$$

де $Ws_j^i = \sum_k Wc_k^{i+1} \cdot L_{j,k}$ – вага важливості k підкласів j -го класу i -го

рівня, причому для кореневого класу рівень $i = 0$;

$Wc_k^{i+1} = Wo_k^{i+1} + Ws_k^{i+1}$ – вага важливості класу C_k^{i+1} ;

$L_{j,k}$ – вага важливості зв'язку між класами C_j^i та C_k^{i+1} онтології.

Перерахунок окремих компонент повної ваги важливості класу відображено на схемі (Рисунок 3.4).

2 У момент внесення на $(i + 1)$ -й рівень нового підкласу йому присвоюється власна вага важливості Wo_j^{i+1} , що дорівнює половині власної

ваги важливості класу, вищого i -го рівня: $Wo_j^{i+1} = \frac{1}{2}Wo_j^i$.

Важливість ваг класу Wc_j^i та усіх батьківських класів аж до кореневого збільшується на величину ваги важливості новоствореного підкласу:

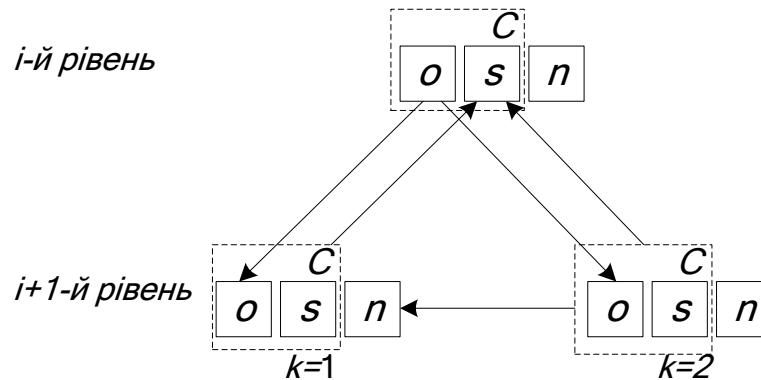


Рисунок 3.4 – Схема перерахунку окремих компонент повної ваги важливості класу

$$Wc_j^m = Wc_j^m + Wo_j^{i+1}, \forall m \leq i.$$

3 Під час встановлення зв'язку між поняттями k_1 та k_2 між відповідними вершинами графу онтології з'являється ребро, а до ваги важливості суміжних класів Wn_1 додається вага важливості Wc_2 і навпаки – до Wn_2 додається вага важливості нового суміжного до нього класу Wc_1 так, що:

$$Wn_j = \sum_k Wc_k \cdot L_{j,k}.$$

Повторне встановлення зв'язків призводить до появи кратних ребер у графі.

4 Кратність ребер відображає частоту зустрічі V пари семантично пов'язаних понять $L_{t+1} = V \cdot L_t$. Кратні ребра після перерахунку не збільшують валентність вершини.

5 Вага важливості екземпляра БЗ дорівнює повній важливості його класу.

Отже, визначена модель онтології БЗ дає змогу розраховувати ваги важливості своїх компонентів під час їх додавання, вилучення і використання у разі експлуатації системи, завдяки чому реалізує механізм адаптації до заданої користувачем ПрО.

Зауважимо, що насправді ваги важливості понять та зв'язків між ними є векторними величинами розмірності кількості класів (наприклад, різної тематики), або задач, визначених у конкретній ПрО. Так, якщо розглядаємо задачу класифікації текстових документів, то в межах онтології має описуватись кілька різних тем. Тому ваги важливості понять та зв'язків залежать від тематики. Нехай адаптивна онтологія \hat{O} описує m тем ПрО – Th_1, Th_2, \dots, Th_m , тоді ваги важливості понять та зв'язків насправді є векторами, компонентами яких є відповідні значення залежно від теми, тобто $W = (W_1, W_2, \dots, W_m)$, $L = (L_1, L_2, \dots, L_m)$. Для автоматизованого реферування наперед потрібно вибрати тему, до якої належатиме текстовий документ, що опрацьовується, щоб система використовувала правильні ваги важливості.

3.3.2 Методи задання ваг важливості понять онтології на основі інтелектуального аналізу даних

Розглянемо визначення ваг важливостей термінів онтології на основі інтелектуального аналізу даних, а саме на основі побудови дерева рішень (ДР). Як відомо ознакові задачі дозволяють для класифікації будувати ДР. Однак ДР

не є панацеєю, оскільки згадувані ознаки, що лежать на відповідній гілці, яку задає клас, не гарантують врахування повної множини ознак, які необхідно врахувати під час класифікації. Тому комбінований метод (побудова ДР – розвинення ДР на основі онтології) виглядає перспективним. Пропонуємо використовувати ДР для визначення ваг важливостей базових термінів, які задають деякий клас, а потім на основі онтології ПО розвинути одержані ваги важливості на всю онтологію для відповідного класу. Тоді для задачі класифікації використовуються значення тих n концептів, які для відповідного класу мають найбільші ваги важливості. На основі побудованого дерева рішень будуються відповідні класифікаційні функції.

Розглянемо гілку дерева. Вершини (ознаки) цієї гілки розташовані на k рівнях. Очевидно, що чим вищий рівень, тим значуща ознака, яка на цьому рівні розташована. Ця евристична думка має бути відображена в значеннях ваг важливості цих ознак. Крім того, пропонується ці ваги важливості пронормувати, тобто, щоб їхня сума для кожного класу (гілки) дорівнювала 1.

Розглянемо два способи визначення ваг важливості базових ознак, які задовольняють вищеописані два припущення.

1 спосіб. Арифметичні ваги важливості. Визначаються як відношення різниці $(k+1)$ рівня дерева та рівня, на якому розташована ознака до суми всіх рівнів гілки, тобто, ґрунтуються на сумі арифметичної прогресії:

$$w_i = \frac{k+1-i}{\sum_{j=1}^k j} = \frac{k+1-i}{\frac{(1+k)k}{2}}$$

2 спосіб. Геометричні ваги важливості. Ґрунтуються на сумі геометричної прогресії:

$$w_i = \frac{2^{k-i}}{2^k - 1}$$

Так для гілки рівня $k=4$ одержимо:

Рівень знаходження ознаки в ДР	Арифметична вага важливості	Геометрична вага важливості
1-й	$w_1 = \frac{2}{5}$	$w_1 = \frac{8}{15}$
2-й	$w_2 = \frac{3}{10}$	$w_2 = \frac{4}{15}$
3-й	$w_3 = \frac{1}{5}$	$w_3 = \frac{2}{15}$
4-й	$w_4 = \frac{1}{10}$	$w_4 = \frac{1}{15}$

Легко бачити, що $\sum_{i=1}^4 w_i = 1$ для обох способів визначення ваг важливості.

Отримані на основі ДР ваги важливості назвемо вагами важливості базових ознак класу і позначимо таку множину ваг важливості W_B . Тепер необхідно їх розвинути на всю онтологію ПО, використовуючи таксономію понять онтології, відношення між поняттями та їхню інтерпретацію. Математично (формально) цей процес може бути записаний у вигляді

$$W_B \xrightarrow{\hat{o}} W.$$

Детальніше це відображення розглянемо наприкінці наступного розділу, наперед детально розглянувши у ньому структуру онтології та типи відношень.

3.4 Інтелектуалізація програмних засобів виконання сценарію аналітичної діяльності на основі методу машинного навчання

Останнім часом для певного кола аналітичних процесів спостерігаються стійка тенденція інтелектуалізації програмних компонентів, які реалізують ці процеси [197]. Це можна пояснити наявністю як об'єктивних, так і суб'єктивних факторів. До об'єктивних факторів відноситься зростаюча складність самих процесів аналізу даних, і, як наслідок, сценаріїв їх обробки. В залежності від цілей та мети аналізу, а також природи даних (структуровані, неструктуровані, слабкоструктуровані, або навіть Web-дані), для аналізу можуть залучатися

різноманітні методи обробки, наприклад, очищення, нормалізація, перетворення форматів, кластерний аналіз, навчання, моделювання, прогнозування, оцінка якості тощо, які самі по собі потребують від аналітика як навичок їх застосування, так і спеціалізованих знань. Крім того, для аналізу різних типів даних мають використовуватися саме ті сценарії обробки, які відповідають як типу даних так й цілям та мети аналізу. Суб'єктивні фактори асоціюються як з діями людини – кроками виконання процесів аналізу складної предметної області, так і можливістю сприйняття та інтерпретації результатів аналітичного дослідження. Безумовно, як об'єктивні, так і суб'єктивні фактори впливають на ефективність методів аналізу. Тому для певного кола аналітичних систем дуже важливо полегшити процеси виконання складних сценаріїв за рахунок їх інтелектуалізації.

Одним із проявів інтелектуалізації є накопичення знань щодо особливостей функціонування аналітичної системи, включаючи й знання про дії аналітика. В цьому контексті сенс накопичення знань полягає в тому, щоб програмний засіб був в змозі самостійно класифікувати нові дані, одержувані при аналізі, і пропонувати користувачу найбільш доцільні кроки сценарію, виходячи з накопичених знань та ситуації що склалася на попередніх кроках [2, 3]. Тобто бажано щоб сучасний аналітичний засіб мав можливість прогнозувати найбільш ймовірний наступний крок складного сценарію та пропонувати його користувачеві, виходячи з наявних в системі знань.

Сценарій АнД, виходячи з парадигми інтелектуалізації програмних засобів, можна розглядати як певну структуру представлення знань, яка використовується для опису послідовності пов'язаних подій – у вигляді орграфа. В цьому підході орграф фактично визначає сукупність шляхів досягнення мети в конкретній стереотипній ситуації (pattern) для заданої ПрО, представленої у вигляді семантичної мережі, наприклад, онтології [4]. Тобто, для генерування сценарію АнД з елементами інтелектуалізації має формуватися орієнтований ациклічний граф можливих операцій сценарію (Directed Acyclic Graph – DAG). Слід нагадати, що DAG відображає припущення про

взаємозв'язок між змінними – вузлами в контексті побудованого графу, в якому відсутні орієнтовані цикли [247]. При такому підході задача побудови сценарію АнД фактично зосереджується на вирішенні відомої задачі пошуку найкоротшого шляху (Single Source Shortest Path) [6]. Отже, пошук найкоротшого шляху в орграфі від витoku до його стоку з урахуванням накопичених знань про стереотипні дії користувача можна розглядати як основу для інтелектуалізації процесу формування сценарію аналітичної діяльності. Враховуючи тенденції щодо інтелектуалізації програмних засобів, така постановка задачі й досі є дуже актуальною не тільки для інтелектуалізації процесу формування сценарію АнД, але й для вирішення певного кола завдань.

У роботі пропонується підхід до вирішення задачі інтелектуалізації процесу побудови сценарію АнД, якій базується на запропонованому інтелектуальному методі визначення наступних подій сценарію АнД, що заснований на розвитку методу машинного навчання, а саме розвитку методу навчання деревами класифікації і регресії (Classification and Regression Trees – CART) з використанням комбінації метрик оцінок ефективності запропонованого варіанту.

Слід зазначити, що в Web-просторі зараз існують певні реалізації вирішення відомої задачі пошуку найкоротшого шляху (Single Source Shortest Path). Найбільш відомим є алгоритм Graph Algorithms in Neo4j, якій входить до складу Javascript library [7] та позиціонується як free and open-source javascript. Алгоритм Graph Algorithms in Neo4j обчислює найкоротший (зважений) шлях від вузла до всіх інших вузлів на графі. Цей алгоритм застосовується одночасно з алгоритмом Shortest Path і алгоритмом Dijkstra та спрямований на моделювання та прогнозування складних динамік, таких як потік ресурсів або інформації.

Але враховуючи особливості функціонування розвинених сценаріїв АнД [1, 3], запропонована власна удосконалена версія методу Classification and Regression Trees, яка відрізняється від відомого методу [7,8] наявністю і можливістю використання інтегрованої моделі опису знань і можливістю

застосування різних метрик при аналізі якості розбиття та через нього вибору наступного кроку ймовірної дії аналітика, більш пристосованого для побудови розвинених сценаріїв. На відміну від існуючих підходів при виконанні машинного навчання, пропонується можливість вибору найбільш оптимальної метрики оцінки якості наближення до бажаного результату навчання – коефіцієнту Gini [11] або методу розрахунку ентропії корисності інформації [12].

Роль та місце інтелектуальних програмних засобів при виконанні сценаріїв аналітичної діяльності показано на рисунку 3.3. Отже, це спеціалізовані ПЗ, які містять в собі необхідні інструменти для здійснення моніторингу за діями користувачів, засоби консолідації інформації в СД, витягу, перетворення, трансформації даних з метою їх адаптації до алгоритмів data mining (зокрема, методів машинного навчання – Classification and Regression Trees), а також засоби візуалізації результатів та можливості корегування сценаріїв на підставі аналізу попередніх кроків.

Формально аналітичний процес можна представити у вигляді вершин орієнтованого графа – $Gr(Sc)$ з'єднаних ребрами (стрілками), який відображує послідовності дій користувача по досягненню мети дослідження. Для більшої реалістичності будемо вважати, що деякі дані можуть бути або втрачені, або спотворені. Спотворені дані зображені у вигляді порожнього кола. Крім того, одна з дій на графі позначена не символами, а цілим числом, яке розглядається як граничне значення (Рисунок 3.5). Завдання полягає в тому, щоб з накопичених даних (послідовності кроків при вирішенні певного кола завдань) отримати знання, на основі яких інтелектуальний засіб (удосконалений алгоритм CART) в змозі запропонувати найбільш вірогідну дію користувача (процедуру обробки даних), виходячи з аналізу попередніх кроків та спираючись на знання про семантику ПрО.

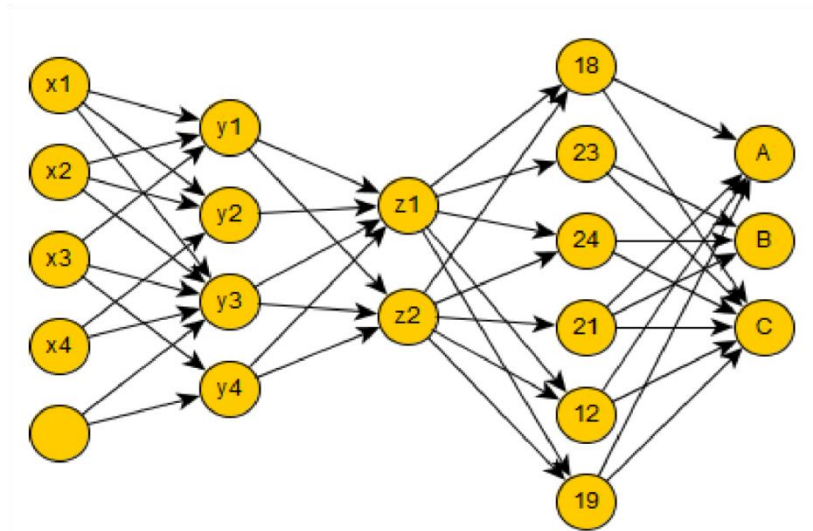


Рисунок 3.5 - Граф всіх варіантів можливих дій користувача

Першим кроком запропонованого методу є опис матриці всіх можливих станів орієнтованого графа – $Gr(Sc)$ (включаючи й втрачені або спотворені дані) – загальної моделі можливих дій користувача (Рисунок 3.6). Для опису матриці можливих станів скористаємося об'єктно-орієнтованою мовою високого рівня Python v 3.6.4 [9]. Вибір цей мови обумовлено необхідністю застосування різних парадигм програмування, зокрема: об'єктно-орієнтованої, процедурної, функціональної та інших, які ця мова підтримує [10].

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import cart
>>> from pprint import pprint
>>> pprint (cart.divideset(in_data, 2, 'z1'))
([['x1', 'y1', 'z1', 18, 'A'],
 ['x2', 'y2', 'z1', 23, 'B'],
 ['x3', 'y1', 'z1', 24, 'C'],
 ['x4', 'y2', 'z1', 23, 'C'],
 ['x1', 'y2', 'z1', 19, 'A'],
 ['x3', 'y4', 'z1', 12, 'C'],
 ['x2', 'y3', 'z1', 18, 'C'],
 ['x4', 'y2', 'z1', 19, 'C']],
 [['x2', 'y3', 'z2', 21, 'B'],
 ['(empty)', 'y4', 'z2', 12, 'A'],
 ['(empty)', 'y3', 'z2', 21, 'C'],
 ['x2', 'y1', 'z2', 24, 'B'],
 ['x3', 'y1', 'z2', 18, 'A'],
 ['x2', 'y3', 'z2', 18, 'A'],
 ['x4', 'y3', 'z2', 19, 'A'],
 ['x1', 'y3', 'z2', 21, 'A']])
>>>
```

Рисунок 3.6 – Матриця всіх можливих станів орієнтованого графа – $Gr(Sc)$

У відповідності до гіпотетичного графу $Gr(Sc)$ в останньому елементі кожного рядку матриці вказується кінцева дія користувача ("A", або "B", або

"С"), виконання якої потрібно спрогнозувати – передбачити. Передбачення ймовірних дій користувача потребує, по-перше, навчання інтелектуальної системи на підставі контрольної вибірки, а, по-друге, формування обґрунтованої гіпотези щодо можливості досягнення останнього елемента кожного рядку матриці станів, тобто забезпечення працездатності усіх вузлів графу – передачі даних для аналізу від одної дії користувача до другої.

Умова забезпечення працездатності усіх вузлів графу:

$$\forall v_i \in V \Leftrightarrow \omega_i(\tau) = 1, \quad \tau \in [0, t),$$

де $V = \{v_i\}$ – множина вузлів графу; $\omega(\tau)$ – булева функція, що набуває значення 1, якщо вузол дій приймає участь в діях користувача і 0 – у іншому випадку; τ, t – поточний час дій у вузлах обробки аналітичних даних.

Умова забезпечення передачі даних між вузлами графу по основних або резервних маршрутах дій користувача для досягнення мети аналітичного дослідження:

$$\forall v_i, v_j \Leftrightarrow \exists v_j \in \Gamma_i, \quad i, j = 1, 2, \dots, n,$$

де Γ_i – множина досяжних вузлів графа з вузла v_i ; n – число вузлів дій, що приймають участь в аналітичному процесі.

Із сформульованих умов витікає необхідна умова стійкості аналітичного процесу, тобто можливість виконання аналітичного процесу для досягнення мети дослідження в усіх вузлах орієнтованого графа. Дана умова полягає в наявності знань про процес АнД у вузлі та наявності маршрутів, які відображають послідовність дій аналітика для досягнення мети дослідження:

$$\forall G(V, L), v_i \in V, l_{ij} \in L \Leftrightarrow \begin{cases} \forall v_i \in V \Leftrightarrow \omega_i(\tau) = 1, \tau \in [0, t); \\ \forall v_i, \forall v_j \Leftrightarrow \exists v_j \in \Gamma_i, i, j = \overline{1, n}. \end{cases}$$

Необхідна і достатня умова стійкості аналітичного процесу, який відображує послідовності дій аналітика щодо досягненню мети дослідження полягає в наявності знань про процес аналітичної діяльності і наявності альтернативних маршрутів передачі інформації:

$$\forall G(V, L), v_i \in V, l_{ij} \in L \Leftrightarrow \begin{cases} \forall v_i \in V \Leftarrow \omega_i(\tau) = 1, \tau \in [0, t); \\ \forall v_i, \forall v_j \Leftarrow \exists v_j \in \Gamma_i^{\geq 2}, i, j = \overline{1, n}. \end{cases}$$

Отже, поточна структура знаходиться на межі стійкості аналітичного процесу, якщо граф отриманої структури зв'язний, та має у своєму складі мости ($N_L \geq 1$) - ребро графа, що сполучає два підграфи, або вузли з'єднання (кореневі вузли) ($N_V \geq 1$):

$$\{K = 1\} \wedge [\{N_V \geq 1\} \vee \{N_L \geq 1\}],$$

де K – це число компонентів графа, а умова $K = 1$ означає, що граф зв'язний; N_V, N_L – число вузлів зв'язності та мостів графа відповідно.

Наявність в структурі моста або вузла зв'язності (кореневого вузлу), які сполучають два підграфи, означає, що усі маршрути дій користувача для досягненню мети дослідження з вершин одного підграфу у вершини іншого включатимуть цей міст або вузол зв'язності. Ця подія істотним чином знижує стійкість аналітичного процесу. Аналіз структур показує, що якщо граф, що описує аналітичний процес, знаходиться на межі стійкості, то система працездатна та виконує встановлений об'єм функцій. Однак, у випадку хоча б однієї відмови моста або вузла зв'язності система переходить у нестійкий стан.

Тому другим кроком запропонованого методу є створення кореневого вузлу графу $Gr(Sc)$ з метою вибору найкращій змінної, виходячи з посилю, що досягнення кінцевих дій можливе лише за стовпцем (кроком n), який містить значення $Z1$ або $Z2$ (Рисунок 3.7).


```

26 def divideset (rows, column, value):
27     split_func=None
28     if isinstance(value,int) or isinstance(value,float):
29         split_func=lambda row:row[column]>=value
30     else:
31         split_func=lambda row:row[column]==value
32
33     set1=[row for row in rows if split_func(row)]
34     set2=[row for row in rows if not split_func(row)]
35     return (set1,set2)

```

Рисунок 3.7 – Завдання умов розбиття множини з метою вибору найкращій змінної

Для цього спочатку необхідно виконати обчислення неоднорідності вхідної множини значень, які містить матриця можливих дій. Мірою неоднорідності множини є ентропія. Для обчислення неоднорідності множини використовувати різні метрики.

Для підвищення якості розбиття скористаємося коефіцієнтом Gini (Gini coefficient) [11]. Коефіцієнт Gini – $K_{Gini}(D)$ є мірою нерівності розподілу деякої величини (D), що приймає значення між 0 і 1, де 0 означає абсолютну рівність (величина приймає лише одне значення), а 1 позначає повну нерівність. Цей коефіцієнт $K_{Gini}(D)$ використовується у деревах рішень при виборі розподілу деякої величини (D). За допомогою коефіцієнта Gini вимірюється двійковий розділ для кожного атрибута – вірогідність появи різних подій P_1 в листі дерева. Коефіцієнт Gini розраховується по формулі [11]

$$K_{Gini}(D) = 1 - \sum_{i=1}^n P_i^2$$

На рисунку 3.8 наведено фрагмент програмного коду розрахунку $K_{Gini}(D)$ при обчисленні приналежності до розбиття.

Обчислення розподілу значень матриці можливих дій сценарію для побудови дерева рішень складається з наступних кроків:

1) На першому кроці обчислюється коефіцієнт Gini – $K_{Gini}(D)$ для кожного значення рядку – тобто вірогідність досягнення кожного з можливих результатів приналежності до певного розбиття (підмножини). Вірогідність

обчислюється шляхом ділення лічильника появ цього результату на загальне число рядків в множині.

```

45 def giniimpurity(rows):
46     total = len(rows)
47     counts = uniquecounts(rows)
48     imp = 0
49     for t1 in counts:
50         p1 = float(counts[t1]) / total
51         for t2 in counts:
52             if t1 == t2:
53                 continue
54             p2 = float(counts[t2]) / total
55             imp += p1*p2
56     return imp
57
58 def entropy(rows):
59     from math import log
60     log2 = lambda x: log(x) / log(2)
61     results = uniquecounts(rows)
62     entr=0.0
63     for t in results.keys():
64         p = float(results[t]) / len(rows)
65         entr = entr - p * log2(p)
66     return entr

```

Рисунок 3.8 – Програмний код розрахунку ймовірності приналежності до розбиття різними методами

2) Далі розраховується вірогідність для кожного рядку методом підсумування. Результатом цього кроку є сумарна вірогідність $\sum_{i=1}^n P_i^2$, яка показує, що для випадково обраного рядка може бути прогнозовано не той результат, який насправді має місце. Слід нагадати, що чим вище вірогідність, тим гірше розбиття. Оскільки вірогідність 0 асоціюється з найкращим результатом, це означає що всі рядки вже розподілені правильно.

3) Оскільки розділення для визначеного атрибуту (наприклад – Z1) розпадається на дві частини D1 і D2, то $K_{Gini}(D)$ для цього розподілу обчислюється за формулою:

$$K_{Gini_{z_1}}(D) = \frac{|D1|}{|D|} K_{Gini}(D1) + \frac{|D2|}{|D|} K_{Gini}(D2)$$

Аналогічно виконується розрахунок $K_{Gini_{z_1}}(D)$ для атрибуту Z2.

4) Наступним кроком є розрахунок ентропії інформації по Шеннону – $H_{gr}(D)$ (ентропія Шенона), яка обчислюється за формулою [12]:

$$H_{gr}(D) = \sum_{i=1}^I P_i \log_2 \frac{1}{P_i} = - \sum_{i=1}^I P_i \log_2 (P_i)$$

Для отримання оцінки того наскільки хороша обрана змінна, алгоритм спочатку обчислює ентропію всієї групи $H_{gr}(D)$, а потім намагається розбити групу за можливими значеннями кожної змінної та обчислює ентропію двох нових груп. Найкраща розбивка визначається на підставі обчислення інформаційного виграшу – різницею між поточною ентропією і середньозваженою ентропією двох нових груп, після чого вибирається та, для якої інформаційний виграш максимальний.

5) Далі алгоритм створює дві гілки: якщо умова істинна, та якщо умова помилкова. Обчислюючи для кожного вузла найкращий атрибут і розщеплюючи гілки, алгоритм будує дерево рішень.

б) Побудова дерева рішень полягає в обчисленні найкращого критерію розщеплення «True» або «False», в якості якого використовується середньозважена ентропія, яка розраховується для кожної пари. Зростання гілки припиняється, якщо інформаційний виграш, отриманий від розщеплення в даному вузлу, виявляється менше або дорівнює нулю (Рисунок 3.9).

```

68 def growtree(rows, scoref=entropy):
69     if len(rows) == 0:
70         return dnode()
71     current_score=scoref(rows)
72     best_gain = 0.0
73     best_criteria = None
74     best_sets = None
75     column_count=len(rows[0])-1
76     for col in range(0, column_count):
77         column_values = {}
78         for row in rows:
79             column_values[row[col]]=1
80         for value in column_values.keys():
81             (set1, set2)=divideset(rows, col, value)
82             p = float(len(set1)) / len(rows)
83             gain = current_score - p * scoref(set1) - (1-p) * scoref(set2)
84             if gain > best_gain and len(set1) > 0 and len(set2) > 0:
85                 best_gain = gain
86                 best_criteria = (col, value)
87                 best_sets = (set1, set2)
88     if best_gain > 0:
89         trueBranch = growtree(best_sets[0])
90         falseBranch = growtree(best_sets[1])
91         return dnode(col=best_criteria[0],
92                     value=best_criteria[1],
93                     tb=trueBranch,
94                     fb=falseBranch)
95     else:
96         return dnode(results=uniquecounts(rows))

```

Рисунок 3.9 – Програмний код побудови дерева рішень

Маючи на увазі той факт, що в процесі побудови дерева рішень може виникнути ситуація коли деякі дані можуть бути втрачені або спотворені, то з метою усунення такої ситуації запропоновано внесення змін до функції

прогнозування наступного кроку, які враховують відсутність даних попереднього етапу – застосовуючи метод оцінки ентропії $H_{gr}(D)$.

За результатом роботи запропонованого алгоритму формується дерево рішень, в якому можливі переходи до різних станів позначено як «True» та «False». Для спрощення безпосередньо процесу вибору наступного кроку дій при формуванні сценарію Анд («True» або «False»), алгоритм доповнено більш зручним механізмом формування семантичних умов переходу у вигляді послідовності операторів «if – then». Семантичною умовою переходу по дереву рішень є вибір коефіцієнту Gini $K_{Gini}(D)$ або ентропії по Шенонну – вірогідності досягнення кожного з можливих результатів приналежності до певного розбиття (Рисунок 3.10).

Для відображення дерева рішень у графічному вигляді бажано інсталиувати відповідний plugin, наприклад, **graphviz** [15], який дає можливість семантичного трактування ймовірних дій користувача. Дерево рішень відображує всі ймовірні кроки дій користувача при виконанні сценарію аналітичної діяльності.

```

114 def classify2(q, tree):
115     if tree.results is not None:
116         return tree.results
117     else:
118         v = q[tree.col]
119         if v is None:
120             tr, fr = classify2(q, tree.tb), classify2(q, tree.fb)
121             tcount = sum(tr.values())
122             fcount = sum(fr.values())
123             tw = float(tcount) / (tcount+fcount)
124             fw = float(fcount) / (tcount+fcount)
125             result = {}
126             for k,v in tr.items(): result[k] = v * tw
127             for k,v in fr.items(): result[k] = v * fw
128             return result
129         else:
130             if isinstance(v, int) or isinstance(v, float):
131                 branch = tree.tb if v >= tree.value else tree.fb
132             else:
133                 branch = tree.tb if v == tree.value else tree.fb
134             return classify2(q,branch)

```

Рисунок 3.10 – Програмний код формування семантичних умов переходу по дереву рішень операторами if–then

На рисунку 3.11 наведено спрощений приклад застосування наведеного підходу для запровадження методів машинного навчання в моделюючому комплексі при формуванні типових сценаріїв обробки інформації, які по'язані

з дослідженнями поведінки гідроакустичних пристроїв в різних морських середах. Такий підхід дозволяє суттєво зменшити кількість помилкових дій користувачів (особливо нових користувачів) при формуванні складних сценаріїв з множиною різноманітних умов використання операторів аналізу даних.

Отже, використання алгоритму побудови дерева рішень це прозорий та дуже наочний спосіб класифікації ймовірних дій користувача при виконанні сценарію аналітичної діяльності, які після навчання видаються як послідовність пропозицій, представлених оператором «if – then» та організованих у вигляді дерева.

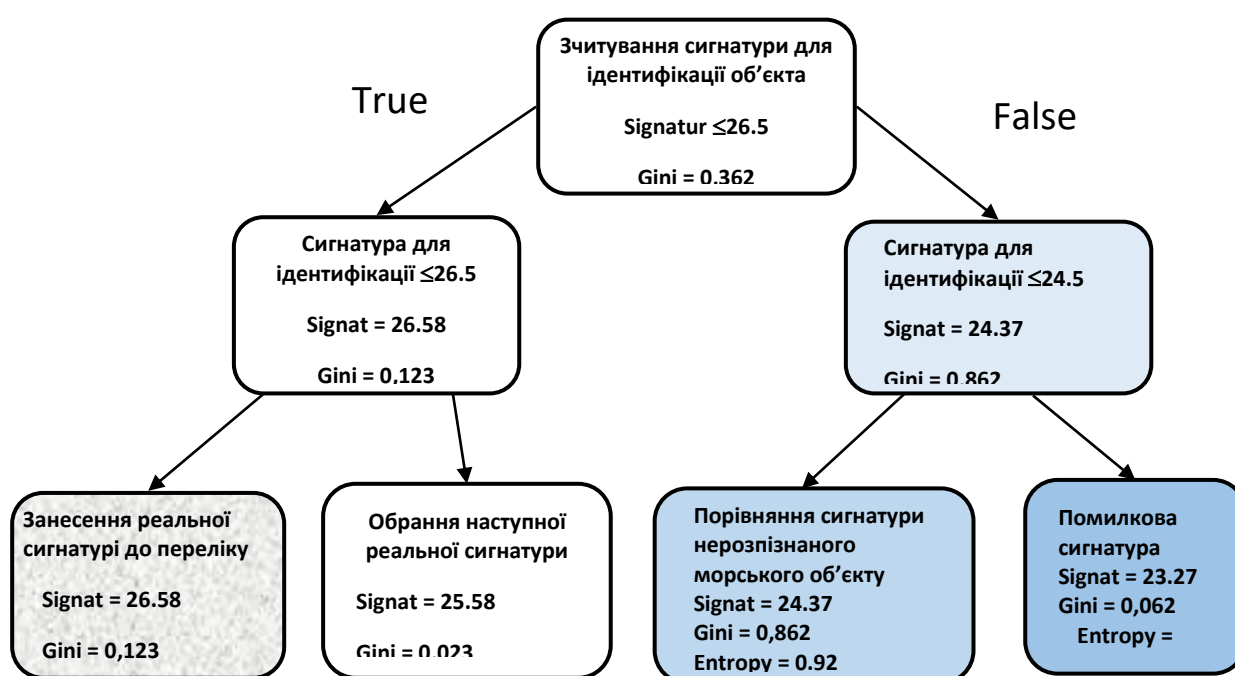


Рисунок 3.11 – Приклад формування дерева рішень

Отже, використання інтелектуального методу визначення наступних подій сценарію АНД дозволяє суттєво зменшити кількість помилкових дій користувачів (особливо нових користувачів) при формуванні складних сценаріїв з множиною різноманітних умов використання операторів аналізу даних. На базі методу розроблений інтелектуальний програмний засіб, що навчається на діях користувача та на основі отриманих в процесі навчання знань представляє дії користувача у вигляді дерева рішень ймовірного сценарію

аналітичного процесу, пропонуючи користувачу найбільш доцільні наступні кроки вирішення завдання АнД. При функціонуванні цього засобу також задіяний процес автоматизованої побудови сценаріїв АнД на основі бібліотеки сценаріїв (прецедентів) для різних предметних областей, що дало можливість в автоматизованому режимі будувати сценарії на основі експертного опису проблемних ситуацій, здійснювати деталізацію сценаріїв та розгалуження їх, та, у свою чергу, суттєво зменшити часові витрати на моделювання та реалізацію сценаріїв АнД.

3.5 Моделювання сценаріїв аналітичної діяльності на основі онтології та прецедентів

При здійсненні АнД фахівець має бути забезпечений не тільки інструментальними засобами аналітики, які підтримують методи On-line аналізу, але бажано мати можливість вибору рішення за допомогою використання експертних знань. Тобто знань, що зберігаються в ІР ПС з урахуванням досвіду вирішення даного завдання фахівцями в минулому на основі вже відомих рішень.

Моделювання сценаріїв АнД на основі прецедентів (Case-based reasoning, CBR) для вирішення нових завдань аналітики в конкретній предметній області [86] дозволяє здійснити формування висновків, що спираються не стільки на логічний висновок при аналізі вхідних даних про наявну ситуацію, скільки на пошук і аналіз прецедентів – випадків формування подібних висновків в минулому [7]. До основних переваг міркувань на основі прецедентів можна віднести можливість використання накопичених знань в ПрО без залучення експертів. Крім того, вже наявні в БЗ ПС сценарії АнД можуть бути використані як контрольні вибірки навчання ПС при побудові нових сценаріїв за використанням інтелектуального методу визначення наступних подій сценарію АнД.

Залучення отології до процесу знаходження аналогій дозволяє зробити міркування по прецедентам більш аргументованими. Звичайно БД прецедентів не містить всі необхідні знання для прийняття конкретного рішення по новій

задачі, однак онтологія дозволяє формалізувати загальні знання предметної області та застосувати їх для вирішення нової задачі. Онтологія також забезпечує семантичну взаємодію між описом прецедентів, які можуть надходити із різних джерел, та формуванням шаблону прецеденту для пошуку ідентичного прецеденту в базі накопиченого досвіду. Знання, що зберігаються в онтології, можуть використовуватися також і для адаптації прецедентів.

Моделювання сценаріїв АнД на основі отології та прецедентів (Case-Based Reasoning, CBR) доцільно застосовувати якщо виконується принцип регулярності – тобто накопичений досвід вирішення завдань може служити відправною точкою процесу пошуку рішення для нових подібних завдань. Ця умова гарантує, що для багатьох проблем в майбутньому буде існувати аналог в минулому досвіді [142].

Процес логічного виводу по прецедентах може бути схематично представлений у вигляді циклу, який складається з наступних етапів:

- формування умов запиту для нового завдання з метою звернення до БД накопичених прецедентів;
- пошук за умовами запиту найбільш схожих прецедентів в існуючій БД прецедентів;
- оцінка знайденого рішення в базі даних прецедентів на предмет відповідності умовам запиту;
- збереження нового рішення в БД прецедентів.

Онтологія в CBR процесах використовується для опису прецеденту в термінах мови OWL, використовуючи класи – $C^{(Ax)}$, екземпляри класів (<Instances>) – $Ex^{(C)}$, відношення між класами (<Object properties>) – $Rel^{(H)}$, атрибути (<Data properties>) – $T^{(A)}$, а також аксіоми (<Axioms>) – $Ax^{(S)}$. Оскільки мови OWL базуються на Description Logics, то вона використовується для процедури логічного виводу при зіставленні вже існуючих прецедентів з новими умовами пошуку.

В БЗ накопичуються і зберігаються прецеденти, які сприймаються як частина знань про предметну область.

Виходячи з вищенаведеного формальну модель системи виводу заснованого на прецедентах, може бути представлено математичним виразом:

$$CBR = \{ Cases, Ont(SD)_{fus}, Reas \},$$

де *Cases* – база прецедентів, накопичених в базі даних системи;

Ont(SD)_{fus} – онтологія (нечітка), яка описує предметну область. Модель онтології описує як чіткі аспекти ПрО, які визначаються аксіомами, так і нечіткі, які визначаються через функцію приналежності.

Reas – алгоритм міркувань по прецедентах – зіставлення умов пошуку з прецедентами, що містяться в БД прецедентів (накопиченого досвіду).

У загальному вигляді нечітка онтологія *Ont(SD)_{fus}* описується виразом

$$Ont(SD)_{fus} = \langle C_{fus}^{(AX)}, Ex_{fus}^{(C)}, Dp_{fus}^{(A)}, Ax_{fus}^{(S)}, Op_{fus}^{(H)}, Mod_{fus}^{(Dp)} \rangle$$

де $C_{fus}^{(AX)}$ – множина класів онтології, включаючи нечіткі;

$Ex_{fus}^{(C)}$ – множина екземплярів нечітких класів (Instances) - прецедентів онтології;

$Op_{fus}^{(H)}$ – множина значень властивостей об'єктів, які описують прецедент;

$Ax_{fus}^{(S)}$ – множина аксіом онтології;

$Dp_{fus}^{(A)}$ – множина значень ступенів (Data properties), які можуть бути додані до об'єкта нечіткої аксіоми;

$Mod_{fus}^{(Dp)}$ – множина «нечітких модифікаторів» функцій приналежності для завдання умов пошуку по прецедентах.

Існує певна аналогія між системами, заснованими на правилах та прецедентах [<http://gaia.fdi.ucm.es/>]. В обох випадках інформаційні ресурси потрібно індексувати, щоб забезпечити ефективний витяг знань.

Застосування CBR заснована на наступних положеннях:

1. Рішення обираються за результатами зіставлення прецеденту із вхідним набором даних, причому вибір засновується на основі знань, що витікають із онтології *Ont(SD)_{fus}*, яка описує прецедент.

2. Для вибору найближчого прецеденту, який відповідає вхідним умовам, використовується система правил $Mod_{fus}^{(Dp)}$.

3. Правила логічного виводу по прецедентах фактично представляють собою шаблони, які описують умови запиту для аналізу вхідних даних та розраховують конкретні значення функції приналежності до прецеденту.

4. Шаблон формується на сукупності чітких аксіом $Ax_{fus}^{(S)}$, що описують прецедент для аналізу, та нечітких значень ступенів параметрів властивостей прецеденту $Dp_{fus}^{(A)}$, що підлягають порівнянню.

5. Застосування правил являє собою ітераційний процес – оскільки шаблон формується на підставі певного уявлення користувача про аналогію між умовами нового запиту до вхідних даних та знаннями про накопичені прецеденти.

6. В процесі зіставлення прецеденту із вхідним набором уявлення користувача може змінюватися, що може змінювати й умови нового запиту $Mod_{fus}^{(Dp)}$.

7. Функція приналежності розраховує ймовірність наближення вхідного набору даних до прецеденту на основі нечіткої логіки (приналежності до цього прецеденту).

Реалізація алгоритму міркувань по прецедентах виконується машиною CBR, для якої необхідно визначити структуру прецеденту. Алгоритм методу міркувань по прецедентах CBR показано на рисунку 3.12.

Структура прецеденту для CBR виводу може бути представлена в наступному вигляді:

$$CASE = \{ IndexCASE, D(IndexCASE), Ef(D(IndexCASE)) \}$$

де $IndexCASE$ – індекс прецеденту для опису вхідних умов;

$D(IndexCASE)$ – множина рішень певної задачі, які можуть бути прийняті на підставі виводу, заснованого на прецедентах;

$Ef(D(IndexCASE))$ – множина оцінок ефективності прийнятого рішення задачі.

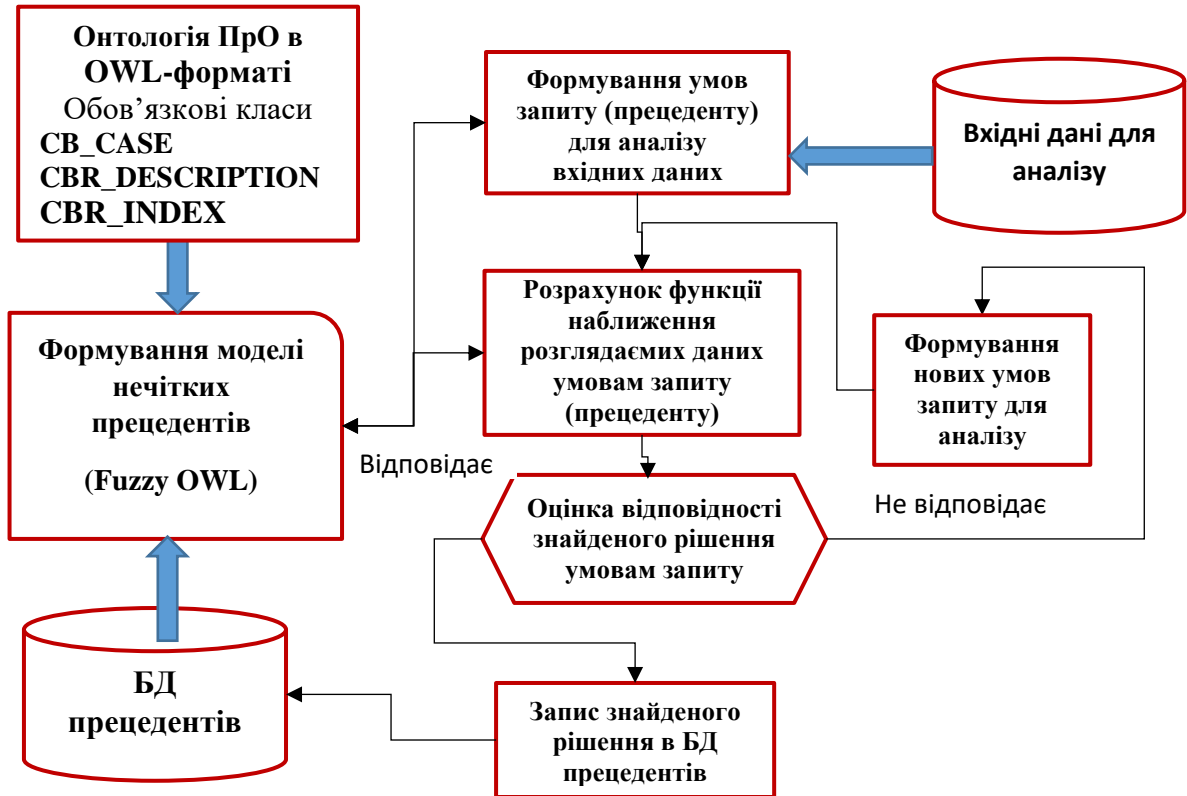


Рисунок 3.12 – Схема алгоритму метода міркувань по прецедентах – CBR

Оскільки прецеденти, на відміну від правил (які оперують із змінними), оперують з конкретними об'єктами класів – $C_{fus}^{(AX)}$ предметної області, а також значеннями властивостей – $Op_{fus}^{(H)}$ цих об'єктів, то початкова ситуація описується виразом:

$$IndexCASE = \{ C_{fus}^{(AX)}, Op_{fus}^{(H)} \}$$

де $C_{fus}^{(AX)}$ – множина об'єктів класів онтології предметної області, включаючи нечіткі;

$Op_{fus}^{(H)}$ – множина значень властивостей об'єктів, які розглядаються для розрахунку функції наближеності.

Реалізація CBR-виводу може виконуватися за допомогою jCOLIBRI – інструментальної платформи для побудови CBR-додатків [68, 69]. jCOLIBRI – це java-фреймворк Мадридського університету, який є безкоштовним і вільно поширюваним. jCOLIBRI дозволяє виконувати обробку прецедентів, представлених в OWL-форматі із застосуванням SWRL-правил (The Semantic Web Rule Language).

Для внесення в БЗ початкового набору прецедентів у середовищі jCOLIBRI може виконуватися редактор OWL-онтології Protégé 5.x, що забезпечує інтеграцію засобів проектування онтології та алгоритмів міркувань по прецедентах – CBR-висновків.

Найважливішим кроком пошуку є оцінка ближнього сусіда. Це спирається на використанні функції подібності для порівняння атрибутів складових і локальних функцій подібності для порівняння простих атрибутів прецедентів при аналізі вхідних даних. jCOLIBRI пропонує і інші методи, які також входять до системи. Наприклад, Expert Clerk та метод фільтрування, який вибирає випадки відповідно до логічних умов над атрибутами. Для текстових додатків CBR jCOLIBRI надає спеціалізовані методи, використовуючи бібліотеки Apache Lucene, GATE.

Машина виводу jCOLIBRI також містить методи оцінки ефективності застосування CBR – це включає стратегії:

- нечіткої онтології (з використанням Fuzzy OWL Plugin);
- набору прецедентів в OWL-форматі.

Застосування машини виводу jCOLIBRI передбачає необхідність налаштування OWL-онтології ПрО на три базових класи:

- CBR_CASE – клас, що містить екземпляри прецедентів;
- CBR_DESCRIPTION – клас, що описує обмеження, які накладаються на прецеденти;
- CBR_INDEX – зберігає структуру прецеденту, тобто класи-нащадки та їх екземпляри, які беруть участь в процесі пошуку відповідного прецеденту.

Рішення, отримане в результаті логічного висновку на основі бази редукційних правил і за вибором користувача в якості правильного, буде занесено в базу прецедентів з вихідними умовами задачі. В результаті даної дії, при вирішенні аналогічної задачі в наступний раз користувач отримає цю рекомендацію як апріорну, тобто отриману на основі аналізу досвіду вирішення такого завдання.

Використання в якості бази знань нечіткої FuzzyOWL – онтології предметної області, набору продукційних правил у сполученні з процесом аналізу прецедентів, дозволяє:

- збільшити гнучкість процесу виведення рекомендації, надаючи користувачу більш широкий вибір варіантів, упорядкованих за ступенем їх релевантності;
- наблизити формалізоване уявлення знань про предметну область до природного для людини варіанту їх уявлення і сприйняття;
- забезпечити отримання користувачем можливості прийняття рішень як з урахуванням аналізу закономірностей конкретної області, так і на підставі досвіду користувачів, що вже стикалися з подібними завданнями.

На перший погляд, може здатися CBR схожі на правила індукції алгоритмів [2] машинного навчання. Як правило налаштування алгоритму, заснованому на прецедентах, починається з інтуїтивного пошуку набору випадків або прикладів, які експерт вважає такими, що їх можна використовувати для навчання.

Далі експерт утворює узагальнення цих прикладів (прецедентів), хоча і неявних, шляхом виявлення спільнот між відібраних прецедентів (випадків у минулому) і вхідними даними сьогодення.

Проте, основна відмінність між неявними узагальненнями в CBR і узагальненнями в правилах індукції полягає в тому, коли робиться узагальнення. Алгоритм правила індукції малює його узагальнення з набору навчальних прикладів, перш ніж цільова проблема ще відома; тобто, він виконує узагальнення.

Головним недоліком методу, заснованому на прецедентах, можна вважати те, що CBR система взагалі ні створює моделей або правил, які узагальнюють попередній досвід. При виборі рішення CBR система розраховує вірогідність наближення вхідного випадку до еталонного прецеденту. На підставі отриманих розрахунків експерт самостійно приймає рішення наскільки вхідні дані співвідносяться до прецеденту, який мав місто у попередньому досвіді.

3.6 Формальна модель сценарію формування висновку на основі нечіткої онтології

При аналізі реальних подій світу доводиться мати справу із знаннями, які, в багатьох випадках, неможливо чітко визначити. В цих випадках моделі, які описують ПрО, засновані на неточній, але об'єктивній інформації про об'єкт. При побудові таких моделей вирішальне значення мають відомості, отримані від експерта, які, в більшій мірі, визначають якісні властивості знань про особливості досліджуваного об'єкта. Опис об'єкта в такому випадку носить нечіткий характер. Саме для вирішення цих проблем американський математик Лотфі Заде в 1965 році розробив апарат нечіткої логіки - "fuzzy logic" [148].

У порівнянні з булевою алгеброю, де 1 являє істину, а 0 – похибку, в нечіткій логіці використовуються також всі дроби між 0 і 1, щоб вказати на часткову істинність [130]. Так запис « μ (високий (X)) = 0,75» говорить про те, що припущення «X - високий» в деякому сенсі на три чверті істинно, а на одну чверть хибно.

Тому адекватним засобом формалізації онтології можуть служити моделі на базі лінгвістичних змінних, нечіткі множини, нечіткі відносини, нечіткі графи і нечіткі дерева, нечіткі обмеження.

У загальному вигляді будь-яка нечітка FuzzyOWL – онтологія може бути представлена наступним чином:

$$I = (I_f, C_f, P_f, D_f, O_f, Mod_f)$$

Де I_f – множина об'єктів класів онтології;

C_f – множина нечітких класів онтології:

$$C_f = \{ C_f^A, C_f^C \}$$

Де C_f^A – множина абстрактних класів, а C_f^C множина конкретних класів онтології;

P_f – множина властивостей об'єктів: P_{f_i}

$$P_f = \{ P_f^A, P_f^C \}$$

P_f^A – множина конкретних властивостей об'єктів, т.е. властивостей типу (Object Property),

P_f^C – множина абстрактних властивостей, тобто властивостей типу даних (Datatype Property);

D_f – множина аксіом онтології:

$$D_f = \{ A_f^{ABox}, A_f^{TBox}, A_f^{RBox} \}$$

A_f^{ABox} – множина тверджень об індивідах,

A_f^{TBox} – множина термінологічних аксіом,

A_f^{RBox} – множина аксіом відносин (ієрархія відносин). Частина аксіом може бути підкласом множини нечітких аксіом, які припускають істинність твердження за певним ступенем.

O_f – множина значень ступенів, які можуть бути додані до об'єкта нечіткої аксіоми:

$$O_f = \{ LD_f, MD_f, ND_f, Var_f \}$$

Де LD_f – множина лінгвістичних змінних,

MD_f – множина ступенів модифікації

ND_f – множина числових значень ступенів,

Var_f – множина змінних

L_f – множина операторів нечітких логік відповідних типів (продукційних правил).

$$L_f = \{ L_f^{Luk}, L_f^{Zad}, L_f^{Goad}, L_f^{Prod} \}$$

L_f^{Luk} – множина операторів логіки Лукасевича,

L_f^{Zad} – множина операторів логіки Заде,

L_f^{Goad} – множина операторів логіки Геделя,

L_f^{Prod} – множина операторів продукційної логіки. Визначаються за допомогою вбудованого відносини `hasSemantics`;

Mod_f – множина «нечітких модифікаторів», тобто функцій модифікації функцій приналежності, нечітких класів і нечітких відносин. Функції можуть бути лінійними або трикутними.

Базовий сценарій формування висновку на основі інтеграції онтології і систем продукції представлений на рис. 4.4.

Взаємозв'язок нечіткої онтології предметної області і набору продукційних правил в процесі логічного висновку здійснюється за допомогою формування запитів до онтології, що генеруються системою аналізу при виконанні набору правил.

Для певного кола аналітичних систем нечітка база знань моделі ПрО може бути зведена до деревоподібної структури з лінгвістичними змінними X_L^k та правилами виводу $(L_l^z)_j$. Отже, модель нечіткої бази знань предметної області KB_{fuz} може бути записана у вигляді математичного виразу:

$$KB_{fuz} = \bigcup_{l,z,j} (L_l^z)_j$$

$$\text{де } (L_l^z)_j = IF (X_1^i = X_1^k) \wedge (X_2^i = X_2^k) \wedge \dots \wedge (X_m^i = X_m^k) THEN X_l^i = X_L^k$$

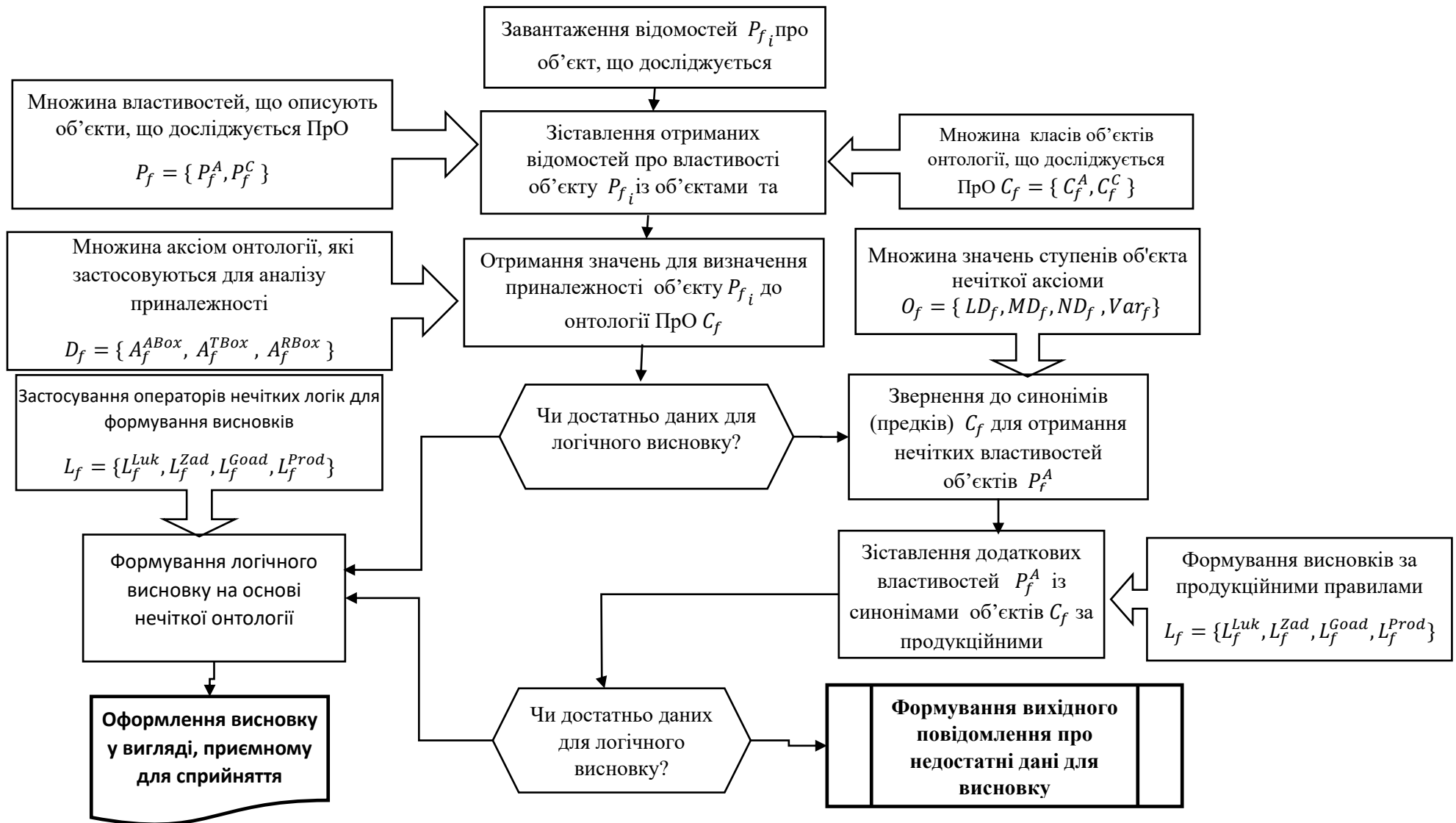


Рисунок 3.13 – Сценарій аналітичної діяльності на основі інтеграції онтології і систем продукції

$(L_l^z)_j$ - j -е продукційне правило виводу ($j = \overline{1, M}$, $M = M_{l_z}$) для визначення z -го терму лінгвістичній змінній з ідентифікатором l ,

$X_{i_{js}}$ – лінгвістична змінна,

$i_{js} = i_{l_z js}$ – ідентифікатор лінгвістичної змінної, ($s = \overline{1, n_j}$, $n_j = n_{l_z j}$) - номер лінгвістичної змінної в лівій частині правила, $X_{i_{js}}^{k_{i_{js}}}$ – терм лінгвістичної змінної $X_{i_{js}}$ з номером $k_{i_{js}}$

$I = \{I_i \mid i = \overline{1, N}\}$ - множина з уведеною на ній ієрархією, що відповідає ієрархії оцінювання ситуації в аналітичній системі

\wedge - оператор кон'юнкцій для визначення z -го терму лінгвістичній змінній.

3.7 Висновки до третього розділу

1. Запропонований теоретичний підхід до вирішення семантичної сумісності (семантичної зв'язаності) процесів аналітичної діяльності (особливо при здійсненні переходів в багаторівневому сценарії) за рахунок застосування онтології ПрО для вирішення конфліктів, пов'язаних з невідповідностями, які виникають при взаємодії кроків сценарію. В основі цього підходу лежить теоретичне обґрунтування вирішення невідповідностей при взаємодії кроків сценарію, насамперед для складних невідповідностей. Визначені та сформульовані умови вирішення невідповідностей між процесами, які визначають переходи між кроками сценарію. Вирішення семантичної сумісності (семантичної зв'язаності) процесів аналітичної діяльності дає можливість застосування всього апарату логіки виводу при вирішенні невідповідностей за рахунок підключення онтологічної моделі ПрО сценарію.

2. На базі запропонованого теоретичного підходу до вирішення семантичної сумісності процесів аналітичної діяльності на виконавчому рівні розроблений алгоритм встановлення взаємодії між двома процесами сценарію, якій забезпечує успішне завершення взаємодії кожним аналітичним процесом.

Це дає можливість перейти до вирішення завдання інтелектуалізації процесу побудови сценарію АнД, розуміючи, що всі переходи між кроками сценарію інформаційно узгоджені, можуть бути здійснені, але у тому числі є можливість зміни порядку надходження переходів між кроками сценарію, якщо це не впливає на загальну семантику процесу, з метою інтелектуалізації побудови аналітичного процесу сценарію.

3. Враховуючи сучасні тенденції щодо інтелектуалізації програмних засобів запропоновано вирішення задачі інтелектуалізації побудови сценарію АнД. Показано, що сценарій АнД можна розглядати як певну структуру представлення знань, яка використовується для опису послідовності пов'язаних подій – у вигляді орграфу. В цьому підході орграф фактично визначає сукупність шляхів досягнення мети в конкретній стереотипній ситуації для заданої ПрО, представленої у вигляді онтології. Тобто, для побудови сценарію АнД з елементами інтелектуалізації має формуватися орієнтований ациклічний граф можливих операцій сценарію. При такому підході задача побудови сценарію АнД фактично зосереджується на вирішенні відомої задачі пошуку найкоротшого шляху в орграфі онтології сценарію від витoku до його стоку з урахуванням накопичених знань про стереотипні дії користувача.

4. Для подання знань в ПС, які використовуються при здійсненні алгоритмів інтелектуалізації процесу побудови сценарію АнД, запропоновано інтегровану модель опису знань, яка на основі моделей задач відображає знання користувача (експерта), дозволяє формалізувати, накопичувати та повторно використовувати досвід щодо способів виконання задач при побудові сценаріїв АнД.

Взаємодія та виконання моделей відбуваються у реальному контексті дієвої аналітики. Розроблено означення контексту, в якому на відміну від існуючих, використовується база знань, яка не залежить від задач, що використовують контекст, а тільки від стану бази знань. В межах цього

визначення розроблена формальна специфікація та методи визначення контексту для факту та класу бази знань.

Використання моделей задач є перспективним і для вирішення комплексу задач менеджменту складної онтології. Зокрема, використання моделей спрощує процеси створення, модифікації, валідації онтології, підвищує її якість.

5. Для вирішення задачі інтелектуалізації побудови сценарію АнД удосконалено математичну модель обчислення розподілу значень можливих подій сценарію та інтелектуальний метод визначення наступних подій сценарію АнД. Математична модель побудована з урахування визначених необхідних і достатніх умов забезпечення передачі даних між вузлами графу по основних або резервних маршрутах дій користувача для досягнення мети аналітичного дослідження та умови стійкості аналітичного процесу, тобто можливість виконання аналітичного процесу для досягнення мети дослідження в усіх вузлах оргграфа сценарію. Для обчислення розподілу значень можливих подій сценарію в моделі використовуються міра нерівності розподілу та ентропія інформації по Шеннону. Удосконалений інтелектуальний метод визначення наступних подій сценарію АнД, який базується на зазначеній математичній моделі та відрізняється від відомих тим, що на основі застосування методів машинного навчання, а саме розвитку методу навчання деревами класифікації і регресії з використанням комбінації метрик оцінок ефективності, та накопичених знань про семантику ПрО - інтегрованої моделі опису знань для визначення кореневого вузлу графу, дозволяє виконувати формування дерева рішень для класифікації можливих наступних кроків сценарію.

РОЗДІЛ 4. МЕТОДИ ВИРІШЕННЯ ЗАДАЧ ВИКОНАННЯ СЦЕНАРІЇВ аналітичної діяльності

4.1 Метод виконання сценаріїв АнД на основі різноманітних Web-сервісів

Формалізовані знання, що накопичуються на рівнях концептуального та об'єктного моделювання сценаріїв АнД, використовуються на виконавчому рівні моделювання сценарію. Як правило, за використання сучасних ПС АнД здійснюється в Web-середовищі, характерною ознакою якого є наявність множини веб-сервісів, включаючи хмарні сервіси, які пропонуються різними провайдерами: Amazon Web Services, Microsoft Azure, Google Cloud Platform, IBM Cloud, Red Hat, Adobe, Oracle Cloud, SAP та інші [1, 19, 56, 77, 112]. Доволі значна частина цих сервісів має вільне розповсюдження і пропонує вирішення певного кола стандартних функцій, таких як: пошуку, витягу даних та знань, верифікації, візуалізації, різні методи обробки даних та інше. Так, тільки Amazon Web Services у розділі «Analytics» пропонує 10 Web-сервісів [254]:

1. AWS Artifact On-demand access to AWS compliance reports.
2. Amazon Athena Query Data in S3 using SQL.
3. Amazon EMR Hosted Hadoop Framework.
4. Amazon CloudSearch Managed Search Service.
5. Amazon Elasticsearch Service Run and Scale Elasticsearch Clusters.
6. Amazon Kinesis Work with Real-time Streaming Data.
7. Amazon Redshift Fast, Simple, Cost-effective Data Warehousing.
8. Amazon Quicksight Fast Business Analytics Service.
9. AWS Data Pipeline Orchestration Service for Periodic, Data-driven Workflows.
10. AWS Glue Prepare and Load Data.

Компанія Google теж пропонує Web-сервіси у напрямку обробки Big Data [112]:

- Analytics Data Warehouse;

- Batch and Stream Data Processing;
- Managed Hadoop & Spark;
- Workflow Orchestration;
- Powerful Data Exploration;
- Intelligent Data Preparation;
- Intelligent Data Preparation;
- Scalable Event Ingestion and Messaging Middleware;
- Build a Serverless Analytics Platform.

Якщо розглядати АнД у веб-середовищі як послідовність виконання типових, в сенсі використання методів пошуку, витягу, верифікації та обробки даних завдань, то неминуче виникає питання використання множини вже існуючих Web-сервісів та рішень для прискорення розробки програмного забезпечення сценаріїв АнД. В цьому контексті сценарій вирішення завдань аналітика можна розглядати як формування послідовності виконання певних функцій обробки даних із залученням вже існуючих у Web-середовищі сервісів і включенням їх до аналітичного додатку.

Метод, якій будується на такому підході, - метод виконання сценаріїв АнД на основі різноманітних Web-сервісів дозволяє динамічно формувати сценарії АнД в залежності від того, які зв'язки та функціональні можливості мають Web-сервіси, які залучаються. Отже, в деяких випадках сценарій роботи аналітика можна розглядати як послідовність Web-сервісів, що корелюються із цілями дослідження. Безумовно, склад Web-сервісів має бути оптимальним, щоб забезпечити правильне виконання завдань аналітика та не ускладнювати роботу програмного застосування в цілому.

Основою запропонованого методу вирішення цих питань є декомпозиція загального сценарію вирішення завдання аналітика - S_c на впорядковану послідовність елементарних завдань - $UsTask_i$, кожне з яких може залучати допоміжний Web-сервіс - $WebS_i$, необхідний для виконання елементарних

завдань. Безумовно, залучаємі Web-сервіси мають корелюватися з функціональним сценарієм [93].

Тоді, виконання сценарію роботи аналітика в Web-середовищі із залученням допоміжних Web-сервісів складається з наступних кроків:

1. Декомпозиція загального сценарію вирішення завдання аналітика Sc на послідовність елементарних завдань (функцій, процедур) - $UsTask_i$ та оцінка можливості залучення Web-сервісу $WebS_i$ для виконання бажаної функції або процедури.

2. Відбір потрібних Web-сервісів із наявних служб для включення їх до загального сценарію АнД.

3. Планування взаємодії (включаючи умови виклику сервісу) відібраних Web-сервісів для реалізації загального сценарію АнД.

4. Безпосереднє виконання послідовності елементарних завдань із залученням відібраних Web-сервісів з метою реалізації загального сценарію АнД.

Після проведення декомпозиції загальний сценарій АнД складається з послідовності виконання функціональних завдань (наприклад, обробки даних, обчислювальних та аналітичних функції, процедур тощо)

$$Sc = \{ UsTask_1, UsTask_2, \dots, UsTask_i \}$$

Кожне функціональне завдання характеризується певним набором ознак (параметрів), що описують особливості їх виконання (джерела, формати даних, обмеження тощо). Кожен Web-сервіс теж описується деяким набором параметрів (наприклад, назва Web-сервісу, виконувана функція, формати даних, взаємозв'язки з іншими Web-сервісами тощо). Такі набори параметрів формують свій метаопис – MD_{SD} (метадані), як для опису функціональних завдань, так і Web-сервісів, залучаємих для їх виконання. З урахуванням особливостей Про. Специфікація метаописів MD_{SD} як функціональних завдань, так і Web-сервісів заснована на стандарті Дублінського ядра [140] і визначає метадані у вигляді

$$MD_{SD} = \langle At^{Func}, D^{Name} \rangle$$

де At^{Func} – являє атрибут опису функції, який визначає завдання або Web-сервіс, що може бути залучений для її виконання;

D^{Name} – дані, що визначають ім'я та інші характеристики, необхідні для визначення взаємодії між функціональними процедурами та Web-сервісами, що їх реалізують.

Таким чином, кожне елементарне завдання $UsTask_i$ може бути описане певним набором метаданих з конкретними значеннями At^{Func} та D^{Name} саме для цієї задачі, включаючи й Web-сервіси, що залучаються.

Формування метаданих для опису компонентів загального сценарію аналітичної діяльності може бути представлена наступним математичним виразом:

$$Sc_{MD} \ni (UsTask_1, \dots, UsTask_i) \equiv \langle Task_i, MD_{SD_i} \rangle$$

де Sc – загальний сценарій вирішення завдання аналітика – визначає набір функцій та процедур, які потрібні для вирішення задачі;

$UsTask_j$ – i -те елементарне завдання у складі загального сценарію аналітичної діяльності;

$Task_i$ – назва i -ого елементарного завдання у складі загального сценарію аналітичної діяльності;

MD_{SD_i} – метаопис i -ого елементарного завдання, потрібний для пошуку відповідних Web-сервісів у Web-середовищі.

Кореляція між завданням користувача і сценарієм (що відповідає завданню цього користувача) описуються відношення еквівалентності:

$$UsTask_k \sim Sc_k$$

Набір Web-сервісів описується як:

$$WebS_{Sc_k} \ni WebS_1, WebS_i = \langle UsTask_i, MD_{SD_i} \rangle$$

Де $WebS$ – множина Web-сервісів, які можуть залучатися для вирішення задач користувача;

$WebS_i$ – i -ий Web-сервіс;

$UsTask_i$ – назва елементарного завдання, яке визиває i – ий Web-сервіс;
 MD_{SD_i} – метаопис i – того Web-сервісу.

Кореляція між сценарієм і Web-сервісами описується відношенням включення ($WebS_1$ включені в сценарій):

$$Sc_k \supset (WebS_1, \dots, WebS_d)$$

Взаємодія компонентів загального сценарію вирішення завдання користувача та Web-сервісів, які залучаються для виконання сценарію показано на рисунку 4.1. Рисунок також відображує кореляцію взаємовідносин між компонентами загального сценарію вирішення завдання аналітика та множиною Web-сервісів, які залучаються для реалізації сценарію.

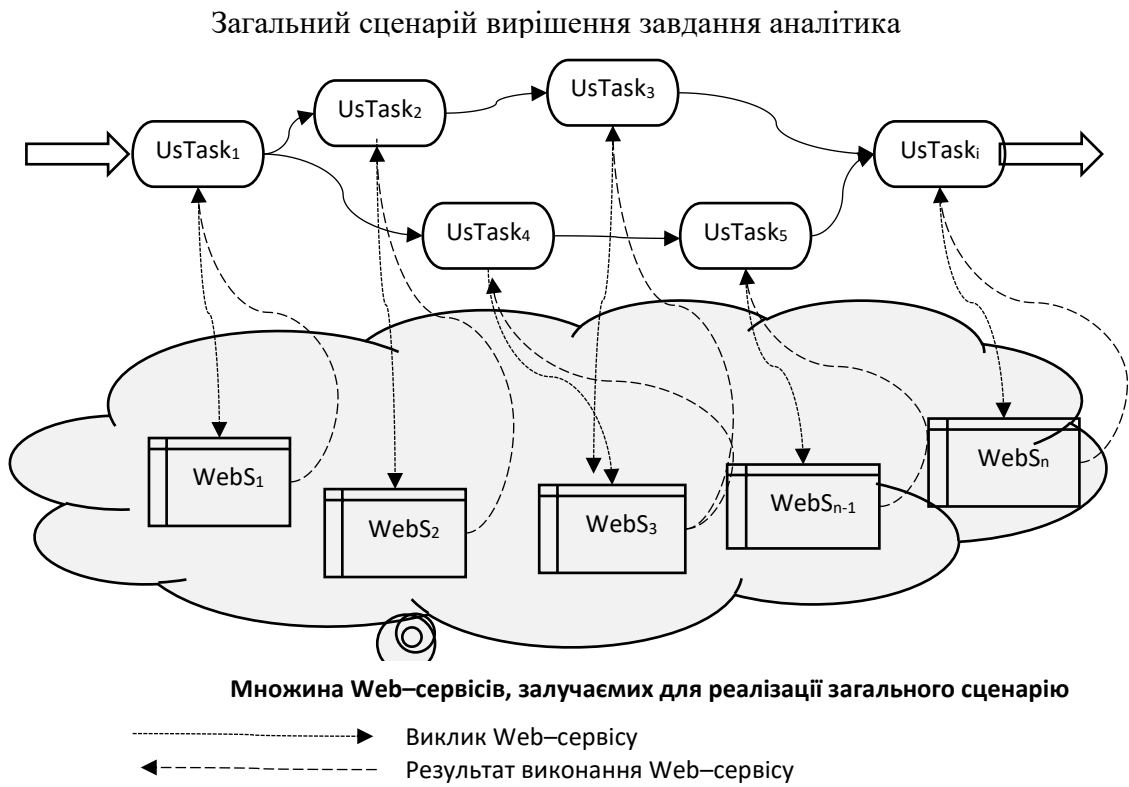


Рисунок 5.1 – Кореляція взаємовідносин між загальним сценарієм та Web – сервісами

Для створення програмної архітектури високого рівня, орієнтованої на вирішення завдань аналітика, можна застосовувати службу Web Services Composition (WSC) [33]. В службі Web Services Composition Dustdar розрізняють

наступні моделі формування саме Web-сервісів, які залучаються до виконання сценарію роботи аналітика:

- Статичну або динамічну композицію – Static and Dynamic Composition;
- Керовану сервісну композицію – Model Driven Service Composition;
- Декларативну сервісну композицію – Declarative Service Composition;
- Автоматизовану та ручну композицію – Automated and Manual Composition;
- Композицію Web-сервісів на основі контексту – Context-based Service Discovery and Composition.

Моделі із статичною, динамічною, керованою, декларативною та автоматизованою композицією формування складу Web-сервісів базуються на механізмах побудови репозиторію Web-сервісів – service repository, в якому зберігаються дані про вже виконані сценарії конкретних завдань із залученням Web-сервісів при їх вирішенні.

Композиція Web-сервісів на основі контексту, тобто семантичні Web-служби [85] забезпечують відкритий, розширений, семантичний фреймворк (framework), що пристосований для опису і публікацій смислового змісту та який покращує сумісність, автоматизованої служби, виявлення та доступу до знань в Internet [5]. Слід підкреслити, що архітектури SOA підтримує всі моделі формування Web-сервісів в конкретному програмному додатку.

В загалі, процес формування динамічного сценарію заснований на механізмі Service Composition, який підтримується архітектурою SOA [127], та являє собою набір окремих повторно використовуваних сервісів. Автоматизована композиція Web-сервісів складається з чотирьох етапів: дослідження, вибору, планування та виконання.

Формальні описи, що використовуються в запропонованому підході, дозволяють формування сценарію виконання аналітичної діяльності на основі Web-сервісів, вибір яких засновано на механізмі порівняння параметрів метаописів Web-сервісів із параметрами функціональних завдань, які підлягають

вирішенню. Таким чином, запропонований підхід і забезпечує можливість виконання завдань кінцевого користувача шляхом динамічного формування послідовності Web -сервісів.

4.2 Методи та моделі формування сценаріїв аналітичної діяльності на порталах знань

В даний час спостерігається тенденція до систематизації та структуризації знань різних ПрО методом побудови спеціалізованих веб-порталів знань (knowledge management portals), які спрямовуються на потреби певного кола кінцевих користувачів АНС [134,135]. Розглянемо застосування методу виконання сценаріїв АНД на основі різноманітних Web-сервісів на прикладі функціонування порталу знань.

4.2.1 Модель функціонування порталу знань на основі сценарного підходу

Розробка моделі функціонування веб-порталу знань на основі сценарного підходу має на меті опис типових процесів, які формують ймовірний сценарій роботи користувача (users workflow) – $UF(SD)$, що відбивають уявлення кінцевого користувача про типові аналітичні дії в процесі роботи з інформаційними матеріалами для вирішення аналітичних завдань в певній ПрО. Слід нагадати, під сценарно-цільовим підходом розуміється сукупність методів опису та організації АНД (для багатофакторних ПрО), застосування яких сприяє генерації різних сценаріїв досягнення цілей аналітичного дослідження при жорстко заданих обмеженнях. З позиції сценарно-цільового підходу Web-портал складається з наступних компонент:

1. Онтології, яка визначає основні об'єкти ПрО (включаючи тематику контенту Web-порталу) та їх взаємозв'язки - $Ont(SD)$
2. Інформаційних ресурсів, потрібних для здійснення АНД в повному обсягу - Informational resources – $IR(SD)$

3. Сукупності сценаріїв Анд, що реалізують процеси вирішення множини аналітичних задач, притаманних визначеній ПрО – Production workflow – $Scen(SD)$.

4. Множини сервісних функції, $PSer(SD)$, .

Концептуальна модель функціонування Web-порталу – $CM(Por)$ обумовлена вкладеністю функцій та методів у сценарії виконання аналітичних дій - $Scen(SD)$, з чого випливає наявність на порталах множини сервісних функції – $PSer(SD)$, які враховують зв'язки між об'єктами ПрО при виконанні аналітичних функцій та процедур обробки даних та знань.

Таким чином, концептуальна модель веб-порталу знань аналітичної системи – $CM(Por)$ на основі сценарного підходу може бути описана математичним виразом:

$$CM(Por) = \{ Ont(SD), UF(SD), FScen(SD), PSer(SD), IR(SD) \}$$

де $Ont(SD)$ – онтологічна модель – задає опис основних сутностей (понять і відносин в межах онтологічної моделі), таксономії та рубрикації контенту ПрО на семантичному рівні. Онтологія описується у вигляді класів об'єктів, екземплярів класів, їх властивостей та відносин між класами та властивостями, включаючи опис IP, необхідних для здійснення Анд в цілому;

$UF(SD)$ – сукупність спеціальних інтерфейсів кінцевого користувача, які визначають структуру знань (контенту) Web-порталу, відображують множину типових цілей дослідження, множину аналітичних задач $An(SD)$, які їх реалізують. Інтерфейс кінцевого користувача надає права доступу користувачеві до IP та знань Web-порталу, а також вибору відповідних сценаріїв вирішення аналітичних завдань, включаючи можливість застосування альтернативних методів досягнення поставленої мети;

$FScen(SD)$ – модель функціональних сценаріїв Анд являє собою сукупність сценаріїв, що пов'язана з множиною типових цілей дослідження. Компонента $FScen(SD)$ надає користувачу можливість генерації сценарію – $FScen_k^j$ послідовності регламентованих дій (послідовності взаємопов'язаних

сервісних функції та процедур) для вирішення конкретної аналітичної задачі - $An(p)_k^j$, з урахуванням багатофакторності та заданих обмеженнях на дані, джерела, інформаційні ресурси, часові обмеження тощо;

$IR(SD)$ – модель інформаційних ресурсів аналітичній ПС, що зберігаються як безпосередньо в ПС, так і територіально-розподілених ресурсів, розташованих в інших доменах WWW-простору. Фактично модель інформаційних ресурсів описує метадані ПрО у вигляді RDF-графу, застосовуючи сервіси, які обробляють метадані та знаходять потрібні знання для вирішення аналітичних завдань.

$PSer(SD)$ – множина сервісних функцій та процедур, за рахунок яких реалізуються сценарії аналітичної діяльності.

Оскільки сучасні аналітичні системи мають справу з різноманітними за природою типами даних (структуровані, неструктуровані, текстові, метадані, гіпертекстові), то сервісні функції та процедури $PSer(SD)$, в свою чергу, поділяється на відповідні підмножини функцій спеціалізованої обробки даних:

$$PSer(SD) = \{Ser_{st}, Ser_{Fuzzy}, Ser_{TM}, Ser_{DL}, Ser_{ont}, Ser_{Con}, M_{date}\}$$

де Ser_{st} – процедури та сервіси обробки структурованих даних, які представлені реляційними та електронними таблицями, базами даних – Data Mining (класифікація, кластеризація, визначення відхилень, прогнозування, регресія, візуалізація та інші);

Ser_{TM} – процедури та сервіси інтелектуального аналізу текстових документів з метою отримання інформації з колекцій текстових документів – text mining, включаючи текстові процесори (категоризація документів, класифікація документів, навчання з учителем)

Ser_{Fuzzy} – процедури та сервіси обробки нечіткої логіки для вирішення задач аналізу нових ринків, оцінці політичних рейтингів, виборі оптимальних стратегій, аналізі перспектив інноваційного розвитку, знаходження функції приналежності визначення вхідних і вихідних змінних;

Ser_{DL} – засоби формування правил логічного виводу;

Ser_{ont} – сервіси та програмні застосунки корегування та управління онтологією;

Ser_{con} – сервіси управління контентом та правами доступу

M_{date} – метаопис (параметри) інформаційних ресурсів, аналітичних задач, сервісних функцій та процедур, а також правил виводу. Завдяки метаопису здійснюється вибір потрібних сервісних функцій та процедур для динамічного формування сценаріїв аналітичної діяльності.

В основні метаописів лежать параметри p_i , які забезпечують реалізацію механізмів публікації сервісу, його пошуку, підключення і використання в аналітичній задачі $An(p)_k^j$. Характеристики параметрів наведено у таблиці 4.1.

Таблиця 4.1.

Назва параметру - p_i	Опис атрибута
Загальні параметри метаопису інформаційних ресурсів	
Name IR	Ім'я ресурсу (дані, знання, сервіс, функція, процедура) – містить формальне ім'я, за яким даний ресурс відомий в репозиторії
Identifier IR	Унікальний ідентифікатор ресурсу, що міститься в репозиторії. Розглядається також як номер функціонального засобу, процедури, сервісу
Subject	Визначає предметну область, до якої належить інформаційний ресурс (функціональний засіб, процедура, сервіс)
Description	Текстовий опис інформаційного ресурсу (сервісу, процедури розрахункової формули тощо)
Параметри, що визначають умови застосування	
Function	Опис функції, яку виконує сервіс при формуванні сценарію вирішення аналітичній задачі - $An(p)_k^j$
Terms of Use	Умови застосування сервісу (функції, процедури) при генерації сценарію
Parameter	Параметри налаштування функції для залучення до сценарію
Data Type	Визначає тип даних, з яким оперує сервіс, функція, процедура
Data Source	Налаштування на адресу джерела даних, з якими оперує сервіс (функція, процедура)

Назва параметру - p_i	Опис атрибута
Result	Налаштування умов отримання результату виконання сервісу (функції, процедури)
Параметри, що встановлюють зв'язки	
IN_net,	Назва програми або сервісу, який здійснив визів сервісу для залучення до сценарію виконання аналітичних дій або розрахунків
OUT_net	Назва програми або сервісу, до якого повертаються результати виконання сервісу
Address	Адреса, що визначає місцезнаходження сервісу, який залучається до сценарію виконання аналітичних дій або розрахунків

Включення метаописів до компоненти $PSer(SD)$ дозволяє автоматизувати процес динамічного формування сценаріїв аналітичної діяльності $FScen_k^j$ у вигляді графу з підключенням різноманітних сервісів, розрахункових формул та інших функціональних засобів, базуючись на параметрах та характеристиках конкретної аналітичній задачі $An(SD)_k^j$.

Множина сервісних функцій та процедур - $PSer(SD)$ має зберігатися у незалежному сховищі – репозиторії, яке може наповнюватися в процесі вирішення нових аналітичних задач, що застосовують нові методи або процедури. Репозиторії не має зв'язків з сховищем типових сценаріїв виконання – $FScen(SD)$.

Пошук необхідних сервісних функцій та процедур для формування сценарію $FScen_k^j$ вирішення аналітичній задачі $An(p)_k^j$, може базуватися на порівнянні параметрів метаописів сервісів $PSer(SD)$ з вимогами до параметрів конкретній аналітичній задачі $An(p)_k^j$, на базі якої будується сценарій. Тобто, до сценарію, який будується, включаються лише ті сервісні функції та процедури, у яких є спільні параметри або області значень параметрів перетинаються.

Математично умови вибору сервісу на основі перетинання параметрів метаопису сервісу та вимог аналітичної задачі можна відобразити наступним чином

$$An(p)_k^j \ni p_{Ank}^j, p_{Ank}^j = \langle Ser_k^j, p_{Ser_k}^j \rangle$$

Де $An(p)_k^j$ – j-та аналітична задача, що належить до k-ої предметної області, яка характеризується параметром p_{Ank}^j

Ser_k^j - назва сервісу, значення параметрів якого $p_{Ser_k}^j$, перетинається із значенням параметру p_{Ank}^j j-ої аналітичній задачі.

Спираючись на таку концептуальну модель порталу можна використовувати методи для автоматизації процесу динамічного формування сценаріїв виконання аналітичних завдань в залежності від настання конкретних умов.

4.2.2 Формальна алгебраїчна система виконання сценаріїв на порталі знань

Одним із підходів виконання сценаріїв АНД є підхід, якій використовує метод формальної алгебраїчної системи, який дозволяє здійснити виконання сценаріїв в процесі функціонування порталу знань. Цей підхід дозволяє коректно і швидко сформуванню процес виконання сценарію $FScen_k^j$ порталу знань.

Формальну алгебраїчну модель формування сценаріїв порталу $FScen(SD)$ можна визначити за допомогою множини A , яка позиціонується як частково визначена функція

$$y = F(x_1, \dots, x_n), (y, x_1, \dots, x_n) \in A,$$

що є n -арної частковою операцією на множині A .

A також через систему $U_A = \langle A, \Omega \rangle$, яка складається з множини A та визначена на сукупності частинних операцій

$\Omega = \{F_s^{ns}\} (s = 1, 2, \dots)$, яка є частиною універсальної алгебри з сигнатурою Ω .

Дві однотипні універсальні алгебри $U_A = \langle A, \Omega \rangle$ та $U_B = \langle B, \Omega' \rangle$ з основними множинами A і B , в яких задані відповідно сигнатури Ω і Ω' вважаються однотипними, якщо можливе встановити таку взаємну відповідність між сигнатурами Ω і Ω' , при якій будь-яка операція $\omega \in \Omega$ та відповідна їй операція $\omega' \in \Omega'$ буде n -арними з одним и тем же n .

Гомоморфні та ізоморфні відображення по визначенням мають наступні умови існування:

- якщо для будь-яких елементів a_1, \dots, a_n та довільної n -арної операції $F \in \Omega$ виконується співвідношення $\phi(F(a_1, \dots, a_n)) = F(\phi(a_1), \dots, \phi(a_n))$, де $\phi(a_i) = b_i$ і $b_i \in B$ ($i = 1, \dots, n$), то відображення $\phi: A \rightarrow B$ є гомоморфним відображенням алгебри U_A в алгебру U_B ,

- якщо між основними множинами A та B встановлюється взаємоднозначна відповідність, то відображення ϕ називається ізоморфним відображенням, а алгебри U_A і U_B називаються ізоморфними,

- якщо $\pi(x_1, \dots, x_n), x_1, \dots, x_n \in A$ n -місткий предикат $\Pi = \{\pi^{ns}_s\}$, де ($s = 1, 2, \dots$) сигнатура предикатів, то система $U_A = \langle A; \Omega; \Pi \rangle$ називається універсальною алгебраїчною системою. Між алгебраїчними системами також можна встановити гомоморфні та ізоморфні відображення.

Алгебраїчна модель PWF представляється формалізмом, що описує кожен бізнес-процес (WF) ПрО, і складається з визначення елементів ПрО, операцій над ними, відповідних аксіом і теорем, що дозволяють описати процес обслуговування WF інформаційними ресурсами.

Елементи, які подаються на вхід та вихід ПС представляють дані алгебраїчної моделі PWF. На фізичному рівні дані представлені значеннями різних параметрів і характеристик ПрО, формулами, межами значень, а також безпосередньо WF, які можуть надходити як вхідні дані на вхід PWF.

В алгебраїчній моделі PWF виділяють наступні операції [14]:

Елементарні операції є кончиною звичайних алгебраїчних операцій відповідно до їх визначення:

-Додавання $+$ - позначає процес підсумовування значень елементів WF або метаописів.

-Множення $*$ - позначає пошук твори значень елементів WF або метаописів.

Множинні операції:

\subseteq - нестроге включення.

\subset - суворе включення.

$\not\subset$ - заперечення включення.

\in - приналежність.

\notin - неналежність.

Логічні операції - це аналоги теоретико-множинних операцій алгебри відносин $[]$.

Визначення 1. Нехай дано WF_1 і WF_2 , в алгебраїчній моделі PWF визначені наступні логічні операції:

Операція об'єднання: $WF_3 = WF_1 \cup WF_2$

Операція перетину: $WF_3 = WF_1 \cap WF_2$

Операція різниці: $WF_3 = WF_1 \setminus WF_2$

Операція композиції: $WF_3 = WF_1 \circ WF_2$

Операція еквівалентності: $WF_3 \sim WF_1$

Операція доповнення: $WF_3 = WF_1^d$

Композиційні операції:

Операція паралельного з'єднання

Операція паралельного з'єднання - Z_{np} , це операція, яка описує процес зв'язування WF в PWF за допомогою простої операції диз'юнкції - \vee .

Узагальнена схема операції паралельного з'єднання представлена на рисунку 4.2 –. На варіанті схеми, позначеному індексом «а», показано об'єднання WF при паралельному з'єднанні за допомогою оператора \vee , що дозволяє реалізувати логічне «або» при розгляді маршруту паралельного з'єднання WF.

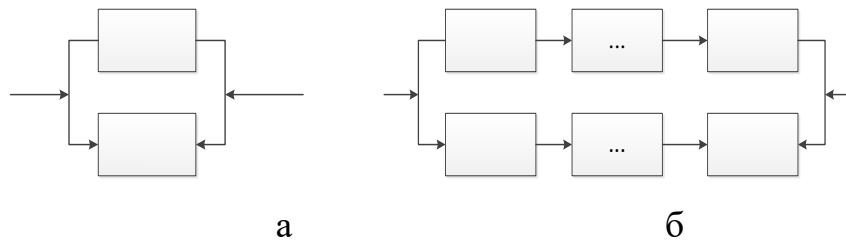


Рисунок 4.2 – Схема паралельного з'єднання

На варіанті схеми, позначеному індексом «б», показано паралельне з'єднання множини послідовно з'єднаних WF. Таким чином, необхідно визначити, як об'єднати WF спочатку послідовно, потім паралельно. Пропонується для операції паралельного з'єднання кінцевого безлічі WF використовувати як оператор диз'юнктивну нормальну форму, де кожна елементарна кон'юнкція є об'єднанням WF в кожній гілці, а сама диз'юнктивна нормальна форма - форму операції паралельного з'єднання розрахунків.

Операція послідовного з'єднання

Операція послідовного з'єднання - Z_{nc} , описує процес зв'язування WF в PWF за допомогою простої операції кон'юнкції - \wedge .

Узагальнена схема операції послідовного з'єднання представлена на рисунку 4.3:

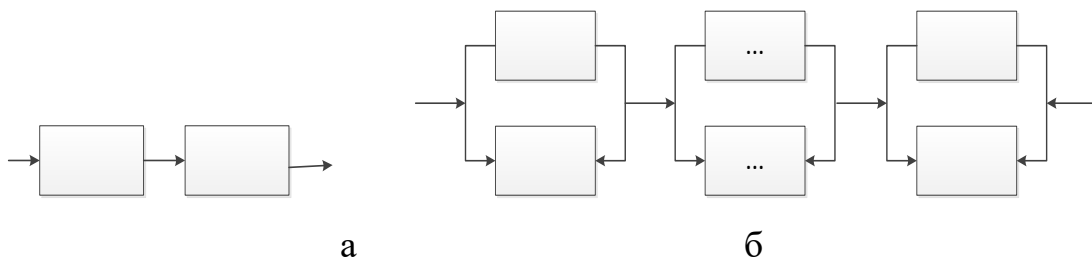


Рисунок 4.3 – Схема послідовного з'єднання

На схемі, позначеної індексом «а» (Рисунок 4.2а), показано варіант об'єднання WF при послідовному з'єднанні за допомогою оператора \wedge , що дозволяє реалізувати логічне «і» при розгляді маршруту послідовного з'єднання WF. Схема, представлена на рисунку 4.2 з індексом «б», позначає послідовне з'єднання множини WF, що дозволяє поєднати WF спочатку паралельно, а потім послідовно.

Запропонований механізм алгебраїчної моделі дозволяє здійснювати комбіноване з'єднання WF, використовуючи кон'юнктиву нормальну форму, де кожна елементарна диз'юнкція являє збідніння WF в кожен паралельний блок, а сама кон'юнктива нормальна форма - форму операції послідовного з'єднання WF.

Операція логічного об'єднання

Операція логічного об'єднання - O_{Π} дозволяє з'єднувати WF, зв'язок між якими заданий через проміжний WF (Рисунок 4.4).

Операція логічного об'єднання дозволяє об'єднати два WF, якщо існує логічний зв'язок цих WF з третім. В цьому випадку операція O_{Π} розглядається як проста операція композиції.

Операція інверсії

Операція інверсії - O_i забезпечує заміну порядку виконання WF на протилежний при з'єднанні часткових WF до загального. Операція інверсії ґрунтується на операторі інверсії - $i-1$. (Рисунок 4.5)

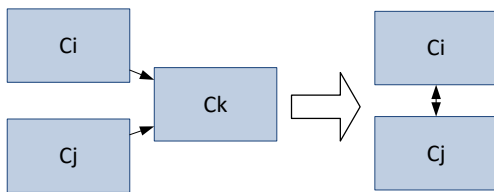


Рисунок 4.4 – Схематичне зображення операції логічного об'єднання

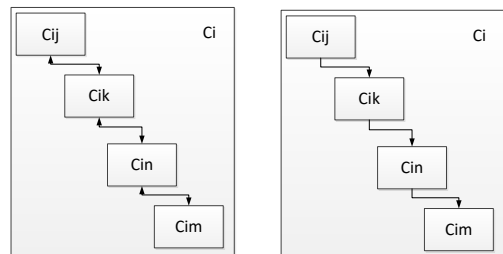


Рисунок 4.5 – Схематичне зображення операції інверсії

Із схеми видно, що послідовність часткових WF можна міняти місцями при з'єднанні в загальний WF. Слід зазначити, що послідовність WF можна змінювати тільки в тому випадку, коли дані не передаються з одного WF в інший, вхідні дані попереднього WF не є вхідними даними для наступного WF.

Операція суміщення - O_c дозволяє з'єднати WF, зв'язку між якими в явному вигляді не існує, враховуючи деякі параметри (Рисунок 4.6).

Операція суміщення дозволяє вибрати з множини функціональних сервісів WF ті, які пов'язані з конкретним WF для перетворення їх в послідовність загального порядку організації WF. Операція O_c базується на порівнянні значень відповідних метаописів WF, для випадків якщо значення метаопису одного WF збігається зі значенням метаопису іншого WF, це означає, що ці WF необхідно з'єднати в один і той же загальний WF:

$$\begin{cases} M_m(C_i) = M_m(C_j) \\ M_k(C_i) = M_k(C_j) \end{cases} \Rightarrow M_m(C_i) = C_j \otimes C_i$$

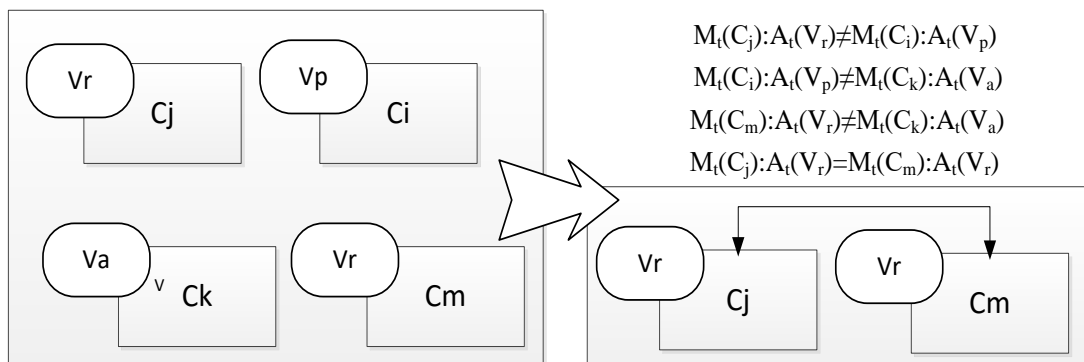


Рисунок 4.6 – Схематичне зображення операції суміщення

Процес суміщення двох workflow для оптимізації їх структури, виконується на базі метаописів ПрО та складається з наступних кроків:

1-й крок – порівняння метаописів різних workflow для їх суміщення:

$$M_1: Atr_1(C_1) = "Value_1" \Leftrightarrow Atr_1(C_2) = "Value_1"$$

2-й крок – перевірка коректності порівняння метаописів:

$$"Value_1" = "Value_1" \Rightarrow$$

3-й крок – визначення та оцінка результату виконання операції суміщення двох workflow:

$$\Rightarrow \begin{cases} R_1 \otimes R_2 \\ R_1 \in R_2 \end{cases}$$

4.2.3 Метод динамічного формування сценаріїв на порталі знань

Вищенаведені моделі дозволяють дуже чітко формувати виконавчий процес PWF (інженерні розрахунки) порталу знань з конкретних функціональних сервісів для визначеної інженерної задачі. Тобто описані моделі дозволяють формування статичних (детермінованих) сценаріїв виконання послідовності розрахунків для певних інженерних завдань. Як правило, такі завдання характеризуються детермінованими, чітко визначеними послідовностями виконання інженерних розрахунків, які потребують використання конкретних сервісів та формул. Але на практиці кінцевий користувач доволі часто зіштовхується з ситуацією, коли в процесі інженерних розрахунків (при настанні певних умов) може кардинально змінюватися послідовність виконання розрахунків.

Тому виникає потреба коректно та динамічно формувати послідовності виконавчих дії на порталі – PWF, безпосередньо, в процесі його функціонування (тобто при настанні певних умов) з об'єднанням відповідних WF для підтримки динамічного порядку їх виконання.

Пропонується процес формування динамічного PWF порталу знань виконувати за допомогою методу впорядкованого дерева, що спростить процедуру модифікації Wf. Враховуючи, що впорядковане дерево – це дерево з коренем, в якому визначено порядок проходження дочірніх вузлів, застосування методу впорядкованого дерева для представлення PWF дозволить визначити послідовність виконання сценаріїв WF.

Процес формування PWF порталу знань методом впорядкованого дерева (в більш загальному вигляді) може бути представлений 3-ма етапами:

- На першому етапі необхідно відкинути з множини $R^{\text{ч}}$ всі WF, в яких жоден з p_{kj}^3 не дорівнює ніякому з $p_{lq}^{\text{ч}}$.
- На другому етапі необхідно виконати процедуру порівняння значення $p_{lq}^{\text{ч}}$, щоб відсікти підмножину WF, які мають спільні параметри, але множини їх значень не перетинаються.

– На етапі три проводиться спрощення формули WF з метою зменшення використання ресурсів порталу.

Дерево формування PWF з використанням описаного підходу зображено на рисунку 4.7. Відсічені вузли показують WF, які були відкинуті при переборі, кореневий вузол – PWF, листами якого є WF, які у нього включаються.

Метод формування дерева PWF складається з наступних етапів.

Етап 1. Порівняння параметрів WF

На першому етапі необхідно провести включення WF до загального на основі порівняння та знаходження тих WF, параметри яких співпадають з параметрами PWF. Далі необхідно провести аналіз множини WF та обирати з неї ті WF, параметри яких співпадають з параметрами WF за правилом:

$$P_d^3 = \bigcup_i^m P_i^4$$

При чому можливе виконання $P_i^4 \cap P_j^4$, де $i, j = \overline{1, m}$

В результаті отримуємо підмножину множини WF яка задовольняє правилу та дерево PWF вершинами другого рівні якого є елементи з цієї підмножини.

Етап 2. Перевірка значень параметрів WF

Часткові WF обираються для об'єднання в PWF на базі порівняння параметрів, які відповідають цим WF, а порівняння відбувається на базі входжень або рівності областей допустимих значень цих даних.

Таким чином, на етапі перевірки значень параметрів WF відбувається порівняння значень параметрів WF з множини WF, що була сформована після 1-ого етапу з значеннями відповідних параметрів PWF.

$$M(p_{ik}^3) \langle \rangle M(p_{jt}^4), \text{ при умові, що } p_{ik}^3 = p_{jt}^4$$

Результатом такого порівняння можуть бути наступні варіанти:

1. Області співпадають:

- $M(p_{ik}^3) = M(p_{jt}^4)$, у цьому випадку область значень k -ого параметру i -ого PWF співпадає з областю значень t -ого параметру j -ого часткового, тобто даний WF включається як вузол у впорядковане дерево PWF.

2. Одна область включає іншу:

- $M(p_{jt}^4) \subset M(p_{ik}^3)$, у цьому випадку область значень t -ого параметру j -ого WF включає область значень k -ого параметру i -ого PWF, тобто даний WF включається як вузол у впорядковане дерево PWF,

- $M(p_{ik}^3) \subset M(p_{jt}^4)$, у цьому випадку область значень k -ого параметру i -ого PWF включає область значень t -ого параметру j -ого WF, тобто даний WF включається як вузол у дерево PWF, але продовжується аналіз значень параметра для інших WF. З метою пошуку WF, область значень параметру якого в об'єднанні з даним буде формувати область значень параметру PWF.

3. Області перетинаються

- $M(p_{ik}^3) \cap M(p_{jt}^4)$, у цьому випадку область значень k -ого параметру i -ого PWF перетинається з областю значень t -ого параметру j -ого WF, тобто даний WF включається як вузол у дерево PWF, але продовжується аналіз значень параметра для інших WF. З метою пошуку PWF, область значень параметру якого в об'єднанні з даним буде формувати область значень параметру PWF: $M(p_{jt}^4) \cup M(p_{ik}^3) = M(p_{hd}^4)$

4. Області не перетинаються

- $M(p_{ik}^3) \neq M(p_{jt}^4)$, у цьому випадку область значень k -ого параметру i -ого PWF не перетинається з областю значень t -ого параметру j -ого WF, тобто даний WF не включається у впорядковане дерево сценарію розрахунку PWF,

Етап 3. Спрощення часткового розрахунку

На третьому етапі відбувається спрощення формули WF. Мінімальна формула може бути отримана в результаті спрощення початкового виразу, який представляє WF через процедуру послідовного застосування властивостей операцій та правил спрощення формул.

Отримане дерево сценаріїв – PWF може бути збережено у БЗ як шаблон для періодичного застосування з можливістю його модифікації. Така модифікація дерева сценаріїв може бути виконана за рахунок використання стандартних операцій над деревами.

Представлення інженерної задачі у вигляді сукупності динамічних сценаріїв PWF з деревоподібною структурою дозволяє обробляти кожну вершину цього дерева, як окремий сценарій, паралельно, обходячи гілки дерева, які є незалежними одна від одної. Такий підхід дозволяє мінімізувати час на динамічне формування сценаріїв - PWF для інженерного порталу.

4.3 Метод семантичної верифікації знань Web-контенту

Дуже актуальною особливо для розподілених Web-орієнтованих аналітичних ПС з компонентами інтелектуальної обробки інформації є проблема верифікації знань є. Одним з методів, за допомогою якого виконується верифікація знань, є використання онтологічної моделі для аналізу інформації, яка автоматично вилучається з розподілених різнорідних джерел [227]. Застосування методу семантичної верифікації знань дозволяє суттєво підвищити рівень довіри до Web-контенту.

Модель інтелектуалізації аналізу Web-контенту [136, 194] заснована на семантичній верифікації знань, що містяться в базі знань інформаційної системи та які описуються онтологією ПрО. Тобто, аналіз здійснюється на основі інформації, яка автоматично вилучається з розподілених різнорідних джерел, розташованих у Web-просторі.

З урахуванням специфіки Web-орієнтованих ПС модель онтології має такий вигляд:

$$Ont(SD) = (C, \leq_C, Rel, \sigma, \leq_R, Ex, \leq_I, Ax)$$

де C – множина класів (концептів) предметної області;

Rel – множина відношень між класами та екземплярами предметної області,

\leq_c – виконує часткове впорядкування множини C (концептів предметної області);

σ – сигнатура - набір операцій, предикатів і відносин, прийнятих в даній системі логічного виводу;

\leq_R – частково впорядкована ієрархія відношень;

E_x – множина екземплярів;

A_x – множина аксіом, які інтерпретують поняття предметної області у вигляді тверджень, що не викликають сумнів.

Семантична верифікація – процес перевірки достовірності знань, поданих за допомогою онтологічної моделі, заснованої на дескриптивних логіках, та сформованих на основі інформації, що видобувається з розподілених різнотипних джерел, як результат гібридного виведення.

Онтологічне гібридне виведення знань – це видобуток нових знань з інформації, яка вилучається з розподілених різнорідних джерел як за SPARQL-запитом, або автоматично при настанні певних умов. Гібридне виведення знань припускає застосування різних типів виведення в одному процесі, а саме:

- виведення на знаннях, закладених у структурі онтології;
- виведення, що базується на знаннях про екземпляри онтології;
- математичне обчислення з урахуванням обмежень на властивості, які мають числові визначення;
- логічне виведення за правилами Semantic Web Rule Language (SWRL) [138];
- виведення у відповідь на SPARQL-запит- або RDQL-запит;
- отримання знань із зовнішніх джерел (в тому числі, від Web-сервісів).

SPARQL (*PARQL Protocol and RDF Query Language*) — мова запитів до даних, представлених по моделі RDF [132], а також протокол для передачі цих запитів і відповідей на них. SPARQL є рекомендацією консорціуму W3C і одною з технологій семантичної павутини [159]. Представлення SPARQL-точок

доступу (SPARQL endpoint) є рекомендованою практикою при публікації даних у всесвітній павутині.

Головна мета семантичній верифікації – надання безперечного свідчення достовірності дескрипційних тверджень, які формують знання.

Метод базується на таких складових:

1. Теза або знання – *VerKnowl*, що подається у вигляді множини дескрипційних тверджень, достовірність яких слід обґрунтувати. Твердження подаються у формі аксіом в термінах словника, заданого компонентом бази знань *TBox*, який визначає таксономію ПрО:

Теза або знання записуються математичним виразом

$$VerKnowl := a: C, (a, b): Rel, \text{ або } a \neq b,$$

де C – концепт (сутність) онтології певній ПрО - *Ont(SD)*.

$(a, b) \in Ex$ – екземпляри онтології, за участю яких здійснюється процедура семантичного виведення;

Rel – семантичне відношення між екземплярами або концептами онтології, задане дієсловом,

2. Аргументи (*Arg*) – визначає конкретні умови виведення іноді у вигляді відомих тверджень або аксіом чи фактів, що можуть застосовуватися для виведення нового твердження *VerKnowl*

3. Пояснення (*Explanation*) – спосіб організації обґрунтованого логічного зв'язку між тезою та аргументами, демонстрація шляху гібридного виведення певного знання.

Для виводу нового знання висунута теза - *VerKnowl* потребує пояснення. Для цього застосовуються аргументи $Arg_1, Arg_2, \dots, Arg_n$, на підставі яких логічно виводиться нове знання - *VerKnowl*. В цьому виведенні вираз *VerKnowl* вважається об'єктом пояснення, а $Arg_1, Arg_2, \dots, Arg_n$, – базисом пояснення, над яким виконуються певні дії

$$\{Arg_1, Arg_2, \dots, Arg_n\} \rightarrow VerKnowl.$$

Згідно зі способами формування знань «дії» поділяються на прості (отримання знання від явного джерела знань) або складні (виведення знання з наявних передумов шляхом застосування певного елемента виведення – правила, формули або запиту).

Для кожного екземпляра, який міститься в базі знань онтології, в залежності від його типу, визначається множина атрибутів, що описують даний екземпляр, наприклад:

- дата (date_of_action);
- правило виведення (has_inference_rule);
- джерело дії (has_source);
- формула, за якою відбулось обчислення (calculated_by_formula);
- та інші, які ставлять у відповідність даній дії множину об'єктів верифікаційної онтології.

Верифікація нових знань, отриманих від явного джерела або виведеного шляхом застосування правил, може бути зведена до верифікації окремих дескрипційних тверджень за рахунок надання безперечного свідчення достовірності дескрипційних тверджень або їх спростування. В процесі верифікації може виникнути ситуація, яка потребує зміни або доповнення онтології новими сутностями, які більш повно відображують ПрО. Онтологічну зміну можна визначити у вигляді онтології нової онтології $Ont(SD)_{new}$, що відрізняється від оригінальної версії $Ont(SD)_{old}$. Онтологічна операція зміни є функцією $Chenge_{opr}$:

$$Chenge_{opr}: Ont(SD)_{old} \rightarrow Ont(SD)_{new}$$

Метод семантичної верифікації припускає три основні операції зміни онтологічної бази знань $Ont(SD)$,:

- додавання нового визначення $VerKnowl$ до онтології $Ont(SD)$;
- модифікація вже існуючих онтологічних визначень;
- видалення визначень, які є зайвими для відповідної онтологічної бази знань.

Під визначенням розуміються всі можливі компоненти онтологічної моделі $Ont(SD)$, а саме: концепти, властивості, відношення, екземпляри, аксіоми. Процес змін в $Ont(SD)$ може бути представлений у вигляді відповідних операцій змін

$$\{Change_{opr1}, Change_{opr2}, \dots, Change_{opr_m}\} \rightarrow Ont(SD)_{new}$$

Метод семантичної верифікації знань (Рисунок 4.7) являє собою ітеративний процес, заснований на прослуховуванні подій, та формуванні

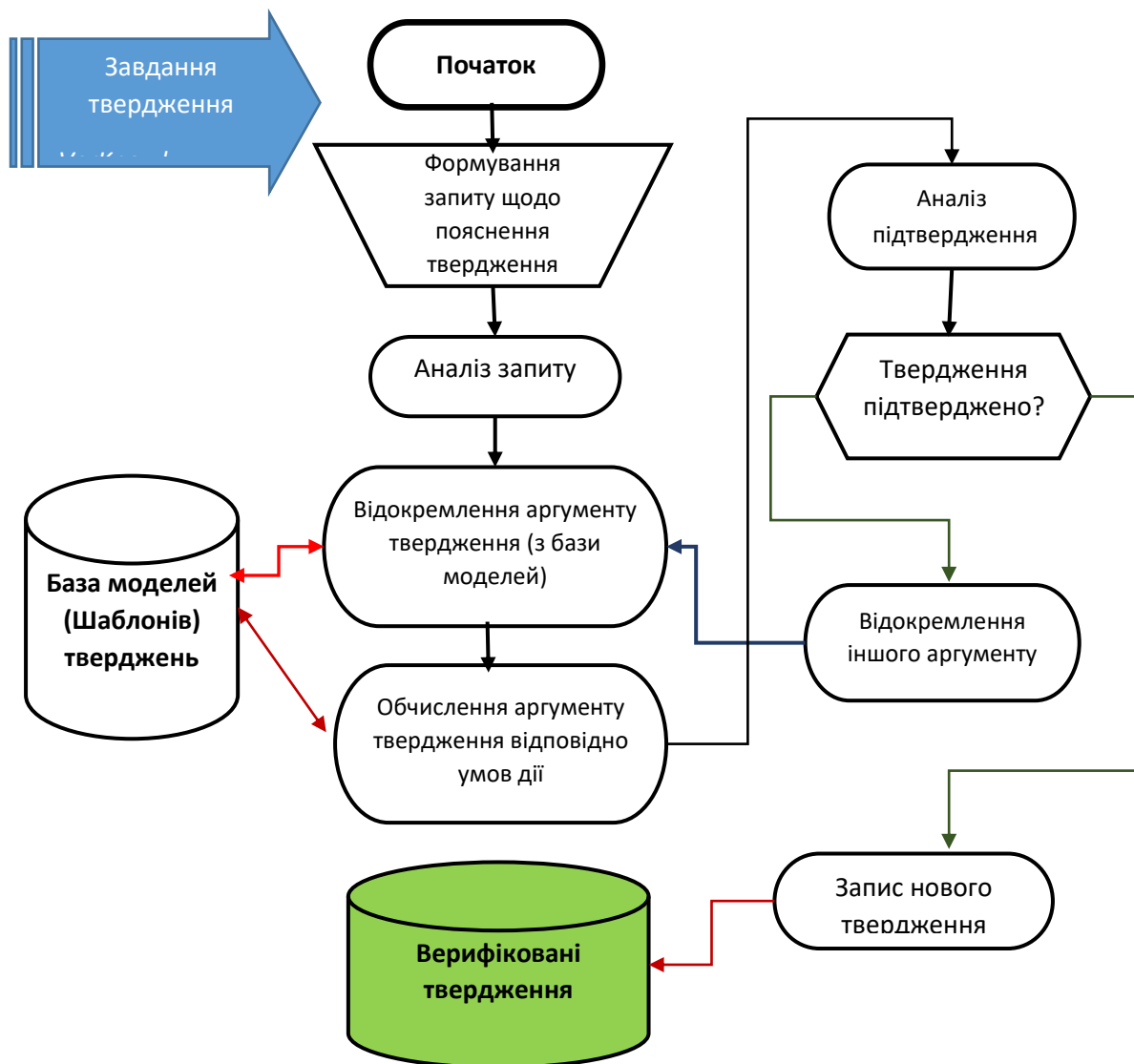


Рисунок 4.7 – Процес семантичної верифікації знань для Web-орієнтованих систем

пояснень шляхом дедуктивного виведення, який спирається на знання, що містяться у багатовимірній деревоподібній структурі. Результатом семантичної верифікації є віднесення нових знань до онтології, із наданням їм певних ідентифікаційних характеристик.

У вершинах деревоподібної структури знаходяться множина тверджень, представлених у вигляді триплетів $\langle \text{суб'єкт} - \text{предикат} - \text{об'єкт} \rangle$. Існує тільки один вузол N_{root} , який є коренем деревоподібної структури, та а якому знаходиться теза – $VerKnowl$, що підлягає верифікації. Інші вузли містять аргументи з множини Arg дерева, які можуть міститись в k піддеревах

$$Exp = \{ \{N_{root} \}, Exp_1, Exp_2, \dots, Exp_k \} \quad k \geq 0,$$

Оскільки відносний порядок піддерев T_1, \dots, T_k є фіксованим, дерево є впорядкованим. На основі тверджень, що знаходяться у вузлах деревоподібної структури, відбувається виведення тверджень, що знаходяться у батьківських вузлах. Гілки позначають переходи від одного стану інформаційної моделі до іншого.

Формування пояснення твердження – $VerKnowl$ полягає у покроковому виконанні аналізу верифікаційного компонента бази знань (рис. 3.1.), автоматичному вилученні елемента підтвердження для кожного кроку виведення твердження та розміщенні відповідного елемента в загальній структурі. Інтерпретація результатів верифікації полягає у визначенні еквівалентності деревоподібних структур. Для цього застосовуються два базових поняття: повної синтаксичної еквівалентності деревоподібних структур та семантичної еквівалентності деревоподібних структур.

Дві деревоподібні структури Exp_1 та Exp_2 вважаються повністю синтаксично еквівалентними, якщо виконуються такі умови:

По-перше, кількість вершин – n дерева Exp_1 еквівалентна кількості вершин дерева Exp_2 , тобто виконується рівняння $|Exp_1| = |Exp_2| = n$

По-друге, кількість гілок дерева Exp_1 еквівалентна кількості гілок дерева Exp_2

$$Br_{Exp1} = Br_{Exp2}$$

Дві деревоподібні структури Exp_1 та Exp_2 вважаються *семантично еквівалентними*, якщо виконуються такі умови:

Кількість вершин дерева Exp_1 еквівалентна кількості вершин дерева Exp_2 , а семантика деревоподібні структури Exp_2 , яка визначається кількістю еквівалентних гілок Br_{Exp1k} відповідає за семантикою структури Exp_1

$$Br_{Exp1k} = sem Br_{Exp2k} \text{ для всіх } 1 \leq k \leq l, \text{ де } l - Exp_1 \text{ або } Exp_2$$

Для практичного встановлення ступеня еквівалентності об'єктів метод семантичної верифікації гібридний, застосовує такі оцінки подібності.

Міра подібності об'єктів через рефлексивну та симетричну функцію:

$$Sim(a, b) = \sum_{k=1}^n W_k Sig_k(Sim_k(a, b) - 0,5)$$

Де (a, b) – об'єкти, що порівнюються:

W_k - вага коефіцієнту подібності, обчислена для k -й властивості об'єктів порівняння (a, d) , $W_k \geq 0$

Sig_k - функція, що виконує обчислення кожного з коефіцієнтів подібності

$Sim_k(A, B)$ – множина синтаксичних і семантичних оцінок подібності об'єктів, що обчислюється за формулою:

$$Sim_c(\lambda, \lambda) = \frac{\sum (C, C) \in (\lambda, \lambda) Sim_c(C, C)}{\max(|\lambda|, |\lambda|)}$$

Обчислення синтаксичної подібності об'єктів для знань, які містяться у текстових документах) заснована на формулі Левенштайна [273]. Для розрахунку відстані Левенштейна між словами текстових документів використовують простий алгоритм, в якому використовується матриця розміром $(n + 1) * (m + 1)$, де n і m - довжини порівнюваних подібностей).

$$Sim_{synt}(L_i, L_j) := \max(0, \frac{\min(|L_i|, |L_j|) - ed(L_i, L_j)}{\min(|L_i|, |L_j|)}) \in [0,1]$$

де L_i, L_j – лексичні одиниці,

$ed(L_i, L_j)$ – лексична відстань між двома об'єктами.

Обчислення семантичної подібності об'єктів здійснюється:

– за синонімами, які визначають класи об'єктів, що вимагає наявність

тезаурусів або словників, – бінарна функція sim_w ;

– за відмінними властивостями об'єктів:

$$Sim_u(a, b) = w_p sim_p(a, b) + w_f sim_f(a, b) + w_a sim_a(a, b),$$

де sim_p, sim_f, sim_a – коефіцієнти подібності властивостей-частин,

властивостей-атрибутів, властивостей-функцій;

– за семантичними відношеннями (sim_n) між об'єктами, що будується на порівнянні семантичних оточень класів об'єктів:

$$N(a^o, r) = \{c_i^o\}, \text{ таке що } \forall d(a^o, c_i^o) \leq r,$$

де N – семантичне оточення класу a^o ,

c_i^o – множина класів, від яких відстань d до класу a^o менше або дорівнює радіусу семантичного оточення класу r .

Метод семантичної верифікації знань дозволяє:

– виконувати процедури підтвердження достовірності знань, використовуючи онтологію ПрО;

– виявляти зміни в механізмах виведення знань;

– у випадку внесення змін до механізмів виведення знань у ПС, перевіряти достовірність існуючих знань з існуючою онтологією;

– перевіряти ідентичність формування знань.

4.4 Модель користувача інтегрованої платформи моделювання сценаріїв аналітичної діяльності

Ефективність використання КС є похідної моделі користувача (аналітика), яка повинна враховувати всі вимоги щодо наявності зручних та зрозумілих сценаріїв роботи, пошуку та обробки інформації з метою отримання нових знань.

Отже, основу моделі користувача – $Ont(SD)_{us}$ складають онтології, що містять опис структури та типології відповідних мережевих ресурсів інтегрованої платформи корпоративного Web-порталу. Крім того, онтологічна модель, я ніяка інша, відображує семантику взаємовідношень між компонентами моделі.

В загальному виді онтологічна модель користувача (аналітика) може бути описана наступним чином:

$$Ont(SD)_{us} = \langle C^{(SD)}, Ex^{(C)}, Rel^{(H)}, T^{(A)} \rangle$$

де: $C^{(SD)}$ – підмножина класів, що описує ПрО, $Ex^{(C)}$ – множина екземплярів наявної онтології, $Rel^{(H)}$ – підмножина відношень між класами та екземплярами онтології, $T^{(A)}$ – підмножина атрибутів, яка визначає властивості класів та екземплярів.

Онтологічна модель користувача інтегрованої платформи моделювання систем керування знаннями - $Ont(SD)_{us}$ включає класи понять, що відносяться до організації аналітичній діяльності, які зв'язані відношеннями різних типів. Різні властивості кожного поняття описуються на основі атрибутів понять і обмежень, що накладаються на область їх значень.

$$C^{(SD)} = \langle C_{ps}^{(SD)}, C_{com}^{(SD)}, C_{st}^{(SD)}, C_{an}^{(SD)}, C_{tem}^{(SD)}, C_{con}^{(SD)}, C_{doc}^{(SD)}, C_{stu}^{(SD)} \rangle$$

Детальний розгляд концептів предметної області, яка характеризує інтегровану платформу на базі корпоративного Web-порталу, а також функціональних завдань, призводить до виділення наступних основних класі:

$C_{ps}^{(SD)}$ – персона. До цього класу відносяться поняття, пов'язані з особистими даними користувачів корпоративного порталу. Атрибутами цього

класу – $T_{ps}^{(A)}$ є персональні дані (прізвище, ім'я) рівень кваліфікації, рівень освіти, напрям аналітичної діяльності та інші властивості.

$C_{com}^{(SD)}$ – компетенції. До цього класу відносяться поняття, пов'язані з правами доступу користувачів до інформаційних ресурсів корпоративного порталу: співробітники корпорації, адміністратори, контент-менеджери, уповноважені персони інших організацій. Атрибутами цього класу $T_{com}^{(A)}$ – є опис прав доступу до інформаційних ресурсів порталу (контенту): рівень володіння математичними методами обробки даних, рівень знання інформаційних технологій, знання іноземних мов тощо читання, створення контенту, видалення контенту, редагування контенту, зміна стану контенту та інші. За допомогою моделі компетенцій можна будувати *профіль користувача*. У процесі роботи на порталі взаємодіють наступні групи учасників – користувачів:

- адміністратор – управляє інтегрованою платформою в цілому та всіма її частинами, включно з операційною системою, базами даних та знань який контролює структуру та права доступу до порталу;
- контент-менеджери, які контролюють зміст матеріалів та доступ користувачів до них в межах своїх зон відповідальності (груп сайтів);
- експерти-аналітики – мають доступ для розміщення матеріалів на визначених сайтах відповідно до їх зон відповідальності, компетентності та призначених їм повноважень;
- авторизовані користувачі – зареєстровані в системі з правом звертатися до загальнодоступної та персоніфікованої інформації корпоративного Web-порталу;
- неавторизовані користувачі. Користувачі порталу з правом звертатися до загальнодоступної інформації.
- координатор порталу – здійснює загальний службовий контроль за його роботою.

На порталі повинно бути реалізоване розділення доступу до контенту й функцій стосовно загальнодоступного публічного сайту та внутрішньої (персоніфікованої) частини порталу.

$C_{an}^{(SD)}$ – напрям діяльності. У цей суперклас входять поняття, що описують аналітичну або науково-дослідницьку діяльність корпорації. До атрибутів напрямів діяльності корпорації - $T_{an}^{(A)}$ відносяться: повна назва напрямку діяльності, скорчена назва напрямку, код за міжнародною класифікацією, англійську назву та інші властивості.

$C_{tem}^{(SD)}$ – тематична група – рубрикатор. Цей клас визначає множину понять

$C_{tem}^{(SD)n} = \{ C_{tem_i}^{(SD)1}, C_{tem_i}^{(SD)2}, \dots, C_{tem_i}^{(SD)n} \}$ – тематичних груп, які формують

тематичну спрямованість – рубрикатор інформаційних ресурсів корпоративного Web-порталу. Атрибутами цього класу є: назва тематичної групи, підгрупа, сайт, права доступу користувача. На основі тематичних груп можна визначати зв'язки асоціативності між елементами контенту, які наповнюють тематичну групу.

$C_{con}^{(SD)}$ – структура контенту. Цей клас описує ієрархічно-мережеву модель організації контенту за визначеною тематикою $C_{tem}^{(SD)}$. Клас $C_{con}^{(SD)}$ має розвинуту структуру підкласів – $C_{con}^{(SD)n} \ni \{ C_{sub_{con_i}^{(SD)1}}, C_{sub_{con_i}^{(SD)2}}, \dots, C_{sub_{con_i}^{(SD)n}} \}$ для встановлення бінарних зв'язків асоціативності між елементами контенту. Атрибутами цього класу - $T_{con}^{(A)}$, є: назва, опис, статті, матеріали, дата і мова публікації, тип документу, URL-адреса документу (файлу), посилання на джерело інформації при розміщенні на порталі матеріалів інших сайтів.

Виділяються різні області асоціативності для елемента контенту:

- *найближче коло* тематично-асоціативного контенту елемента;
- *помірне* коло тематично-асоціативного контенту;
- *широке*, тобто узагальнююче, коло тематично-асоціативного контенту.

$C_{doc}^{(SD)}$ – документація. Цей клас служить для опису різного роду документації, такої як ДСТУ, ISO, закони, постанови, інструкції, та інші. До атрибутів класу - $T_{doc}^{(A)}$ відносяться: назва, опис, дата публікації та мова.

$C_{stu}^{(SD)}$ – навчальні матеріали. Цей клас містить відповідні підручники, довідники, методичні матеріали і т.п. До атрибутів класу – $T_{stu}^{(A)}$ відносяться: назва, автор, опис, ключові слова, дата публікації і мова.

Підмножина відношень – $Rel^{(H)}$ визначає зв'язки між класами онтології, екземплярами класів та їх властивостями – атрибутами. У моделі зовнішнього користувача актуальні наступні відношення:

- асоціативні відношення;
- структурні відношення - <частина-ціле>;
- відношення наслідування;
- відношення <клас-дані>.

Асоціативні відношення використовується для встановлення зв'язку між класами, встановлюється відношення типу <зв'язаний з>, наприклад, між напрямом діяльності $C_{an}^{(SD)}(i)$ та рубрикатором тематичної групи – $C_{tem}^{(SD)}(j)$, де j – тематична група

$$Rel_{ij}^{(H)} = \{ C_{an}^{(SD)}(i) \times C_{tem}^{(SD)}(j) \}$$

Структурні відношення (частина-ціле). Наприклад, відношенням <частина-ціле> описується зв'язок між класом <Документація> - $C_{doc}^{(SD)}(i)$ та <тематична група> $C_{tem}^{(SD)}(j)$

$$Rel_{ij}^{(H)} = \{ C_{doc}^{(Ax)}(i) \in C_{tem}^{(SD)}(j) \}$$

Відношення наслідування використовується для передачі атрибутів і відношень від батьківського класу - $C_i^{(SD)}$ до дочірнього - $C_j^{(SD)}$.

$$Rel_{ij}^{(H)}(T_m^{(A)}) = a_i, r_i \mid T_{doc}^{(A)}(C_i^{(SD)}) \rightarrow a_i, r_i \mid T_{doc}^{(A)}(C_j^{(SD)})$$

Відношення виду <клас-дані> використовується для зв'язки конкретних екземплярів понять з класом.

$$Rel_m^{(H)}(C_m^{(SD)}) = Ex^{(C)}(C_m^{(SD)}) \in C_m^{(SD)}$$

Для керування роботою КС визначено базові, службові типи даних. Ці дані входять у загальну таксономію онтології під розділом *Constructs*. Прикладом службових даних є типи атрибутів (*Attribute Types*) та Ролі (*Roles*). Визначено такі базові типи атрибутів як *String*, *Number*, *Date*, *List*, *Tuple*. А також похідні від них. Наприклад такими похідними типами від типу *String* є *Name* (назва) та *Description* (опис). Типи атрибутів не можуть використовуватися як самостійні сутності. Для групування даних використані такі базові типи як *Сутності* (*Entities*), *Відношення* (*Relations*) та *Моделі* (*Models*).

Такий підхід дозволяє з одного боку розглядати службові конструкції, сутності, відношення та моделі як різновид загальних сутностей (і реалізувати спільні для них функції, наприклад, підтримку атрибутів), а з іншого – враховувати їх відмінності.

4.5 Висновки до четвертого розділу

1 Показано як формалізовані знання, що накопичуються на рівнях концептуального та об'єктного моделювання сценаріїв Анд, використовуються на рівні розробки комп'ютерної моделі виконання сценарію. Як правило, Анд здійснюється в Web-середовищі, характерною ознакою якого є наявність множини Web –сервісів, включаючи хмарні сервіси, значна частина яких має вільне розповсюдження і пропонує вирішення певного кола стандартних функцій, таких як: пошук, витяг даних та знань, верифікація, візуалізація, різні методи обробки даних та може бути задіяна. З метою використання множини вже існуючих Web–сервісів та рішень, які корелюються з функціональним призначенням сценарію, яке описується його семантикою, для забезпечення моделювання сценаріїв Анд запропоновано метод виконання сценаріїв Анд на основі різноманітних Web-сервісів, які реалізує процесу формування динамічного використання Web-сервісів. Метод заснований на виборі веб-сервісів за використанням механізму порівняння параметрів метаописів веб-

сервісів із параметрами функціональних завдань сценарію, які задаються його метаописом.

2. З метою більш наочного розуміння методу виконання сценаріїв АнД на основі різноманітних Web-сервісів був розглянутий та показаний процес його застосування при моделюванні сценаріїв аналізу задач обробки даних за використанням порталу знань. Визначено модель функціонування Web-порталу знань на основі сценарного підходу, в основі якої лежать формальні описи типових дій вирішення аналітичних завдань в певній ПрО користувачем порталу знань. Розглянута формальна алгебраїчна модель виконання сценаріїв, яка описує послідовність дій сценарію, які складаються з визначення елементів ПрО та операцій над ними, що дозволяють описати процес обслуговування виконання сценарію інформаційними ресурсами. Модель дозволяє чітко формувати виконавчий процес сценарію порталу знань з конкретних функціональних сервісів для визначеної задачі, що в свою чергу дозволяє здійснити динамічно (автоматично) формування процесу виконання статичних (детермінованих) сценаріїв послідовності розрахунків для певних завдань за використанням методу впорядкованого дерева в більш загальному вигляді.

3. Важливою задачею при виконанні сценаріїв АнД особливо для розподілених Web-орієнтованих програмних систем з компонентами інтелектуальної обробки інформації є задача верифікації знань, які використовуються сценарієм, та повнота та несуперечливість яких обумовлюють ефективність вирішення задач аналітики. Вирішення цієї задачі розглянуто за використанням методу семантичної верифікації знань, якій базується на процесі перевірки достовірності знань, поданих за допомогою онтологічної моделі ПрО, заснованої на дескриптивних логіках. Метод дозволяє:

- виконувати процедури підтвердження достовірності знань, використовуючи онтологію ПрО;
- виявляти зміни в механізмах виведення знань;

- у випадку внесення змін до механізмів виведення знань у ПС, перевіряти достовірність існуючих знань з існуючою онтологією;
- перевіряти ідентичність формування знань.

Розділ 5. Комп'ютерна моделююча система моделювання сценаріїв АнД

5.1 Структура та архітектура комп'ютерної моделюючої системи (інтегрованого програмного інструментального середовища) для тестування і оцінки запропонованого теоретичного підходу моделювання сценаріїв АнД

Проектування комп'ютерної моделюючої системи - комп'ютерного середовища для моделювання сценаріїв АнД (КС) є досить складним питанням, яке включає необхідний мінімумом вимог, а саме:

- застосування сучасних технологій та інструментальних засобів проектування для інтеграції різноманітних програмних застосувань, включаючи засоби аналізу великих даних в сполученні з онтологічними засобами опису знань;
- наявність гнучкої моделі опису знань і наукової інформації (контенту) для різних за фахом та рівнем підготовки фахівців, а також його перегляду, пошуку інформації та знань за визначеною тематикою;
- наявності зручних та функціональних засобів для редагування наповнення та актуалізації контенту;
- інтеграція різних систем на єдиному корпоративному веб-порталі для уніфікації доступу користувачів до контенту;
- наявність гнучких механізмів адміністрування та управління правами доступу користувачів до інформації та знань;
- широке застосування веб-технологій, які забезпечують роботу різноманітних програмних застосувань через стандартні браузері без необхідності створення спеціального клієнтського програмного забезпечення.

Враховуючи сучасні тенденції, проектування конкурентоспроможних інформаційних систем, заснованих на знаннях, функціонування таких систем має здійснюватися на основі моделей, в яких взаємодія між об'єктами будується на

базі веб-технологій. Таки технології спираються на інтернет протоколи, побудовані відповідно до принципів та стандартів відкритих систем (Open System Interconnection, OSI) [62 64,].

Такий спосіб реалізації ПС дозволяє користувачам працювати в системі через стандартні браузері без спеціального клієнтського програмного забезпечення, а розробникам – здійснювати інтеграцію різних програмних застосувань на єдиному Web-порталі.

Аналіз сучасних методів формалізованого опису та побудови сценаріїв Анд на базі комп'ютерних технологій – «Схемно-рекурсивний метод» [4], «Матричний метод опису та формування сценарію» [5], «Метод «сценарних областей» [7], включаючи й методи сценарного аналізу каскадних події [6], дозволяє зробити такий загальний висновок - жоден з методів не охоплює вісь процес моделювання з верху до низу. Для формалізації багаторівневих сценаріїв сучасних систем організаційного управління найбільш ефективними є гібридні методології, які спираються на графічні методи моделювання сценаріїв у сполученні з семантичною їх інтерпретацією на базі онтологічної технології. Запропонований підхід дозволяє представляти та моделювати Анд «зверху до низу» починаючи з моделі інформаційної взаємодії різних за фахом і повноваженнями експертів та закінчуючи детальної проробкою сценаріїв обробки даних та отримання результатів аналізу для різних ПрО.

Моделювання сценаріїв Анд вимагає використання технологій, які забезпечують швидкий доступ до реалізованої функціональності та передбачає відпрацювання наступних етапів:

- 1) побудова опису ПрО;
- 2) побудова сценаріїв Анд;
- 3) виконання сценаріїв аналітичної діяльності.

На кожному з етапів вирішується свій перелік функціональних задач, що, у свою чергу, визначає функції відповідних компонентів КС та задачі, які вони повинні вирішувати (Рисунок 5.1), а саме:

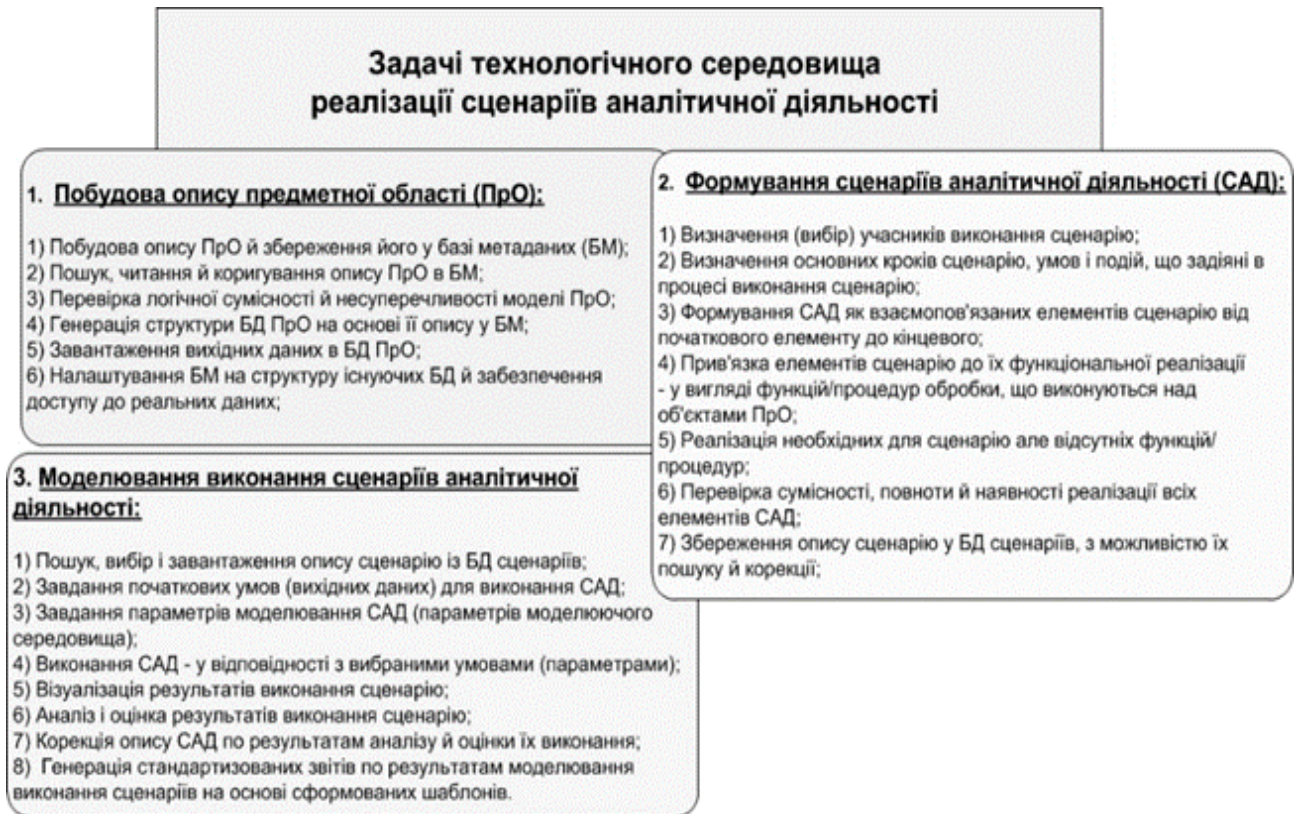


Рисунок 5.1 – Задачі КС моделювання сценаріїв Анд

1. Побудова опису предметної області передбачає реалізацію компонентами КС наступних функцій:

- 1) побудову опису ПрО та збереження його у базі метаданих (БЗ);
- 2) пошук, читання та коригування опису ПрО в БЗ;
- 3) перевірку логічної сумісності та несуперечливості моделі ПрО;
- 4) генерацію структури БД ПрО на основі її опису в БЗ;
- 5) завантаження вихідних даних у БД ПрО;
- 6) налаштування БЗ на структуру існуючих БД і забезпечення доступу до реальних даних.

Формування сценаріїв Анд передбачає рішення наступних функціональних задач з реалізацією відповідних функцій у КС:

- 1) визначення (вибір) учасників виконання сценарію;
- 2) визначення основних кроків сценарію, умов і подій, що задіяні в процесі виконання сценарію;
- 3) формування сценарію як взаємопов'язаних елементів сценарію від

початкового елемента до кінцевого;

4) прив'язки елементів сценарію до їхньої функціональної реалізації - у вигляді функцій/процедур обробки, що виконуються над об'єктами ПрО;

5) реалізації необхідних для сценарію але відсутніх функцій/процедур;

6) перевірки сумісності, повноти та наявності реалізації усіх елементів сценарію;

7) збереження опису сценарію у БД сценаріїв, з можливістю їхнього пошуку, корекції та повторного виконання.

Моделювання виконання сценаріїв АнД передбачає реалізацію наступної функціональності КС:

1) пошук, вибір і завантаження опису сценарію з БД сценаріїв;

2) задання початкових умов (вихідних даних) для виконання сценарію;

3) задання параметрів моделювання сценарію (параметрів моделюючого середовища);

4) виконання сценарію відповідно до вибраних вхідних умов (параметрів);

5) візуалізація результатів виконання сценарію;

6) аналіз і оцінка результатів виконання сценарію;

7) корекція опису сценарію за результатами аналізу і оцінки їхнього виконання;

8) генерація стандартизованих звітів за результатами моделювання сценаріїв на основі сформованих шаблонів.

Виконання наведених вище етапів моделювання сценарію носить циклічний характер. При рішенні задач на кожному з етапів можливе повернення на будь-який із попередніх етапів для внесення необхідних змін і повторне проходження наступних етапів для відпрацювання проведених змін.

Така специфіка моделювання сценаріїв АнД висуває наступні вимоги до формування КС:

— реалізація всієї функціональності сценарію відбувається в єдиному інтегрованому технологічному середовищі;

— реалізація функцій окремих етапів АНД відбувається у виділених функціональних підсистемах, доступ до яких забезпечується через спільний інтегрований графічний інтерфейс;

— взаємодія між основними функціональними підсистемами відбувається через загальну інформаційну базу;

— реалізація функціональності КС забезпечуються в розподіленому середовищі та в режимі багатокористувацького доступу.

Рішення наведених вище задач КС вимагає наявності відповідних програмних засобів, які функціонують у рамках єдиної інтегрованої інструментальної системи моделювання.

Структуру КС наведено на рисунку 5.2.

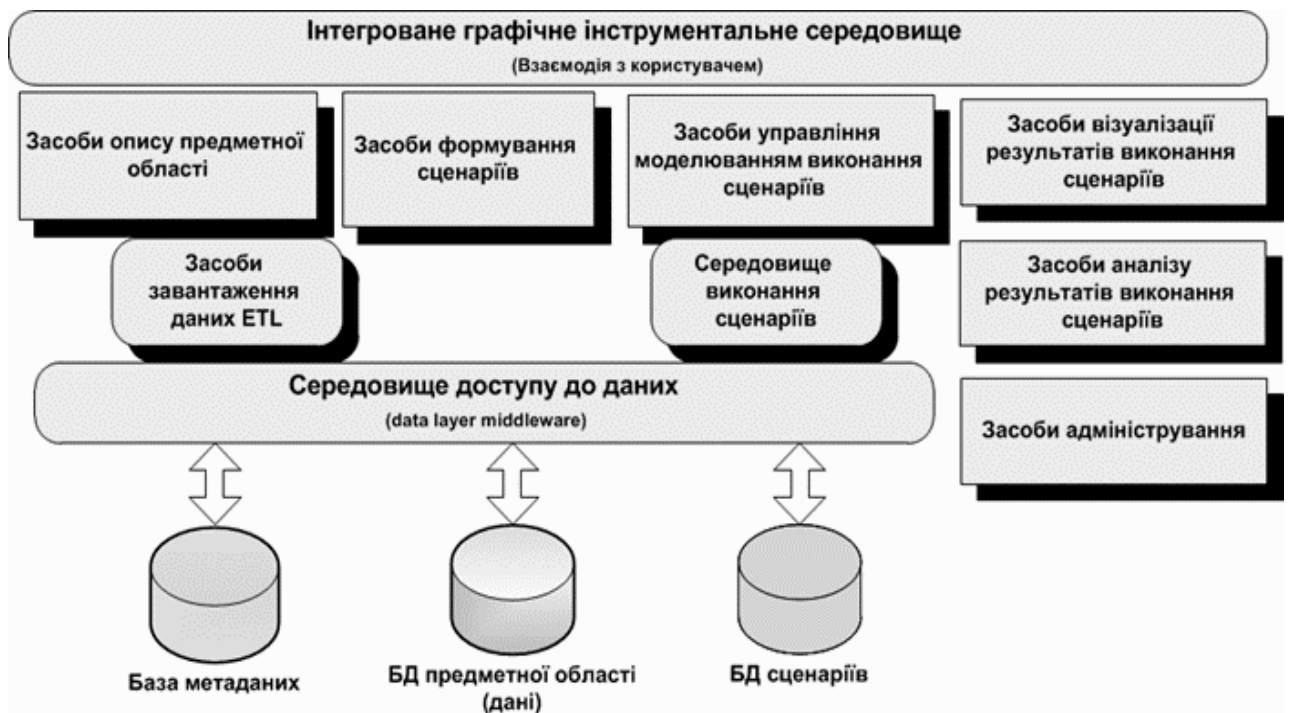


Рисунок 5.2 – Структура КС моделювання сценаріїв АНД

Взаємодія користувачів з КС відбувається за допомогою інтегрованого графічного інструментального середовища, яке надає можливість виконувати основні операції АНД, використовуючи єдиний графічний інтерфейс. Виконання основних операцій моделювання сценаріїв і доступ до відповідних інструментальних засобів КС відбувається через єдиний графічний інтерфейс.

Реалізація функцій кожного з етапів моделювання забезпечується наступними інструментальними засобами КС:

1) опис предметної області — побудова опису ПрО та збереження його в БЗ;

2) побудова сценаріїв — побудова сценаріїв Анд та збереження їхнього формалізованого опису в БД типових сценаріїв;

3) управління моделюванням виконання сценаріїв — засоби управління вбудованим у КС внутрішнім моделюючим комплексом для виконання сценаріїв Анд. Здійснюють управління процесом імітаційного моделювання виконання сценаріїв.

Доступ до інформаційної бази КС забезпечується середовищем доступу до даних, яке є програмним забезпеченням проміжного шару (data layer middleware). Налаштування КС на конкретні бази даних виконується засобами адміністрування за рахунок налаштування відповідних параметрів середовища доступу до даних.

Дані, що необхідні для моделювання сценаріїв, зберігаються в БД ПрО. Завантаження нових даних із зовнішніх джерел забезпечується інструментальними засобами завантаження даних (ETL).

Результати моделювання виконання сценаріїв за допомогою засобів візуалізації результатів виконання сценаріїв можуть бути представлені в різних формах, таких як:

— табличне представлення, з агрегованими даними по різним групам, з можливістю довільної фільтрації і групування даних;

— графіки різних форм, у дво- та тривимірних форматах;

— просторове представлення - використання геоінформаційних можливостей представлення інформації;

— мультимедійні форми - представлення результатів у відео і звуковій формі.

Засоби аналізу результатів виконання сценаріїв забезпечують аналіз

відповідності одержаних результатів заданим цілям аналітичного дослідження на основі визначених і формалізованих критеріїв оцінки.

Засоби адміністрування забезпечують можливість формування КС шляхом конфігурування різних функціональних засобів і налаштування необхідних параметрів для моделювання сценаріїв АнД.

Виходячи з запропонованого підходу проведено огляд існуючих мов високого рівня та інструментальних програмних засобів формального опису моделей сценаріїв сучасної аналітики, включаючи й графове представлення, а також визначення їх функціональних переваг, сценарних характеристик та недоліків для різних структур організаційного управління, у тому числі й для багаторівневих структур. За результатами аналізу запропоновано створення комп'ютерної моделюючої системи (інтегрованого програмного інструментального середовища) для тестування і оцінки запропонованого теоретичного підходу моделювання сценаріїв АнД як Web-портал, якій є захищеною точкою взаємодії з різноманітною інформацією, представленою базами даних та знань, які підтримуються Web-технологіями, програмними застосуваннями. Web-портал має централізований вхід і спеціальні засоби для зручної навігації по інформаційних ресурсах та знанням відповідної ПрО. Framework для інтеграції різнорідних інструментів моделювання сценаріїв АнД складається з наступних програмних засобів та мов високого рівня (Рисунок 5.3):

1) Засоби формування та адміністрування веб-порталу комп'ютерного моделювання сценаріїв АнД.

2) Графічний редактор для моделювання сценаріїв інформаційної взаємодії верхнього рівня – **BizAgi Process Modeler Version 3.6**. (open-source Modeler) [271,272], який підтримує стандарт ISO/IEC 19510 (November 2013) – Business Process e Model and Notation, Information technology. Окрім візуального відображення, сценарії може бути експортовано в базу даних у форматі XML або XPDЛ та конвертовано у OWL-модель онтології для подальшого аналізу.

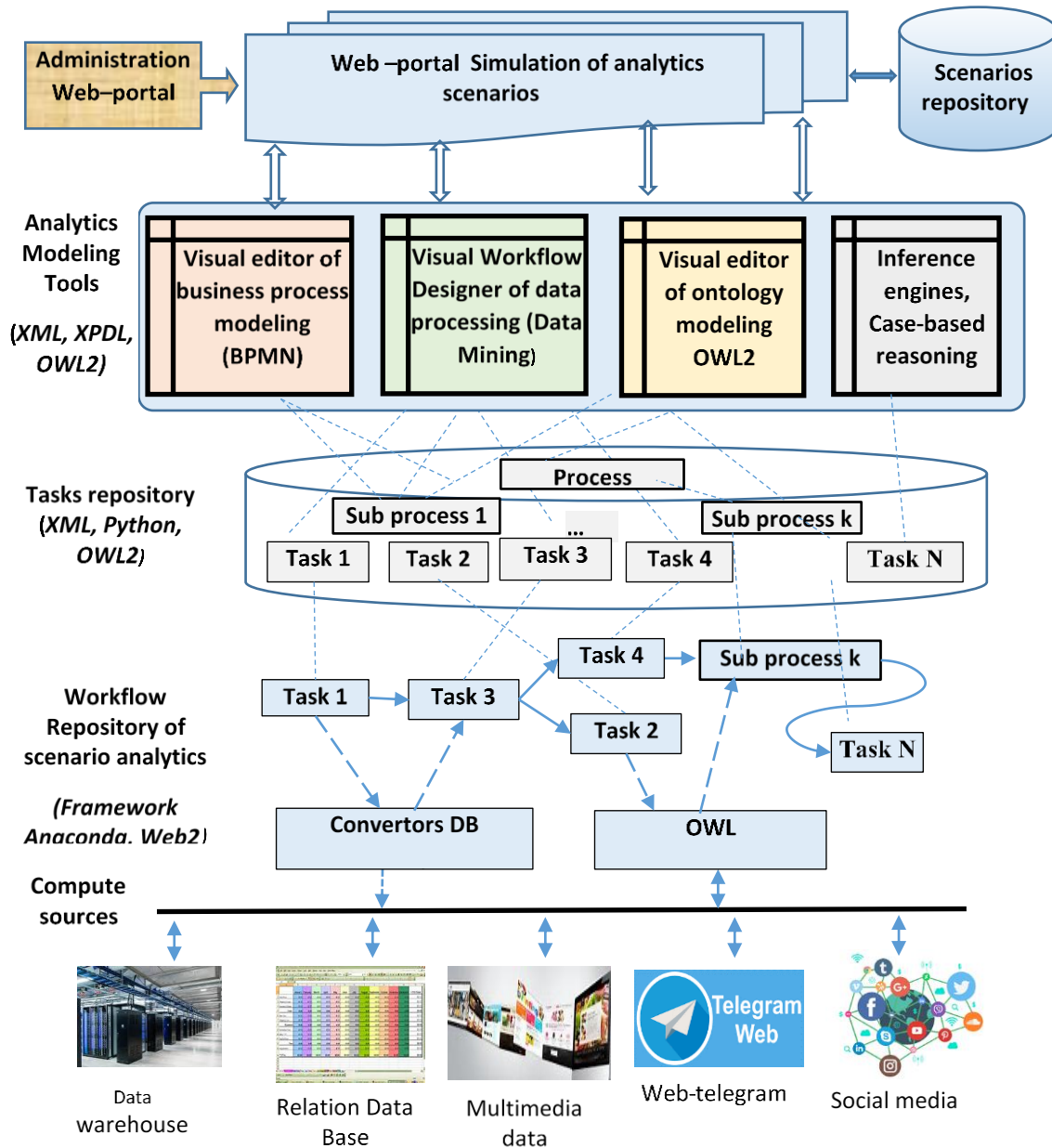


Рисунок 5.3 – Framework для інтеграції різномірних інструментів моделювання сценаріїв АНД

3) Студія та графічний редактор для моделювання сценаріїв та процедур обробки широкого спектру даних аналітики (структурованих, слабоструктурованих, неструктурованих включаючи методи DataMining, TextMining та WebMinin) Це open-source RapidMiner Studio та Visual Workflow Designer for Data Scientists Програмна платформа Technical University of Dortmund – більш 500 віджетів [120, 121]. Результатом дослідження в RapidMiner є виконувана модель сценарію обробки та аналізу даних в форматі XML-коду.

4) Редактор та фреймворк для моделювання онтології – Protégé 5 - Free, open-source ontology editor and framework for building intelligent systems [119].

5) Машина логічного виводу на прецедентах для сфер діяльності, заснованих на накопиченому досвіді – jCOLIBRI 1.0 in a nutshell. A software tool for designing CBR systems.

6) Мова програмування для опису та виводу знань при моделюванні сценаріїв пошуку знань та асоціацій в Web-просторі – OWL2 Web Ontology Language Document Overview (Second Edition) W3C Recommendation 11 December 2012: URL.

7) Об'єктно-орієнтована мова високого рівня Python 3.6.4. для безпосереднього формування інтерфейсів кінцевого користувача та роботи з бібліотеками NumPy для роботи з багатовимірними масивами. Бібліотека WSGI — інтерфейс шлюзу з веб-сервером (Python Web Server Gateway Interface), пакети для доступу до різних СУБД: PostgreSQL, Oracle, Sybase, Firebird (Interbase), Informix, Microsoft SQL Server, MySQL та sqlite

8) Фреймворк та система управління середовищем Anaconda (відкрите програмне забезпечення більш 1500 програмних модулів) для широкомасштабній обробки даних, аналітики, застосування методів машинного навчання. різних програмних продуктів. Anaconda підтримує мови програмування Python та R.

Запропонований підхід дає низку переваг - починаючи з суттєвого спрощення самого процесу моделювання та верифікації сценаріїв Анд, і, закінчуючи, реалізацією моделі в інтегрованому середовищі з отриманням кодів (scripts) виконуваних задач, функції та операції, а також формуванням інтерфейсу кінцевого користувача.

Верхній рівень архітектури інструментальних засобів моделювання утворюють різноманітні програмні додатки. В якості редактору онтології обрано відомий редактор онтологій Protégé 5. – розроблений Стенфордським університетом [109], який відноситься до візуальних редакторів та дозволяє

підтримати всі фази життєвого циклу онтології відповідно до вимог ISO/IEC 15288:2002 [58] – від розробки семантичної мережі та створення бази знань.

Доступ користувачів до інформаційних ресурсів здійснюється за допомогою пошукового механізму, засобів навігації та візуалізації, а також шляхом безпосереднього обміну знаннями між користувачами (наприклад, шляхом обміну посиланнями на ресурси).

Користувач системи моделювання застосовує редактор онтології Protégé для створення та модифікації онтології. Головні функції цього програмного компонента наступні:

- експорт та імпорт онтології,
- створення, модифікація та знищення сутностей онтології,
- збереження сутностей як таксономії, реалізація успадкування атрибутів,
- редагування атрибутів,
- створення, модифікація та знищення відношень,
- створення, модифікація та знищення обмежень на атрибути.

Взаємодія користувача з проміжним рівнем додатків здійснюється за допомогою мов запитів і маніпулювання даними (наприклад, SPARQL, DL, SQL та інші). Особливе місце в процесі виведення знань відводиться машині логічного виведення на основі мови OWL 2 та дескрипційної логіки – DL. В якості інструментального засобу моделювання та інтелектуального аналізу даних обрано програмну платформу RapidMiner, яку в дослідженнях компанії Gartner за 2014 [117] та 2017 роки [76] визнано лідером в області Advanced Analytics. RapidMiner реалізує більш 400 різноманітних операторів обробки як структурованих, так і неструктурованих даних (включаючи текстові) [110]. Вихідні дані та результати аналізу можуть бути представлені різними форматами CSV, Excel, XML, SAS, Access, AML, ARFF, XRFF, Database, SPSS, Stata, Sparse, dBase, C4.5, BibTeX, DasyLab, URL, включаючи роботу з хмарними сховищами та управління процесами і групами об'єктів. Система підтримує

експорт та імпорт онтології та усіх пов'язаних з нею наборів даних (фактів, моделей) як окремих контекстів моделювання.

Сервіс навігації та пошуку інформації по ресурсах порталу містить:

- репозиторій, в якому зберігаються метадані – каталог ресурсів, призначений для реєстрації та подальшої каталогізації всіх інформаційних ресурсів, що входять до Web-порталу, а також забезпечення інформаційної та функціональної бази для ефективного пошуку серед зареєстрованих ресурсів, зокрема з використанням банку інформаційних об'єктів;

- карту Web-порталу, призначену для відображення основних змістовних розділів порталу, яка пропонує користувачу огляд основних функціональних можливостей порталу;

- метапошукову систему, що здійснює пошук інформації та знань серед інформаційних джерел Web-порталу, а також із використанням зовнішніх інформаційних джерел через конвертори перетворення даних.

До складу інтегрованої платформи моделювання систем керування знаннями при проведенні АНД входять також засоби адміністрування та управління контентом.

Успішне функціонування Web-порталу великою мірою залежить від правильності вибору програмно-технічної платформи. Для цього мають бути виконані такі вимоги:

- відповідність міжнародним стандартам відкритих систем, зокрема стандартам розроблення, супроводу й документування – ISO/IEC 15288:2002Б, ISO/IEC 9126-1:2001 та іншим;

- відповідність міжнародним стандартам в галузі керування якістю – ISO/IEC TR 9126-2 (Software engineering – Product quality);

- інтегрованість, що забезпечує існування розвинених технологічних засобів інтеграції з іншими прикладними системами та базами даних;

- адаптованість, що означає наявність засобів налаштування корпоративного порталу під функціональні вимоги конкретної організаційної

системи, а також підтримує технології перенесення рішень з одної платформи до іншої.

Запропонована архітектура побудови інструментальних засобів моделювання на базі корпоративного Web-порталу надає можливість існуючими технологічними засобами за мінімальними витратами вирішити питання створення інтегрованої платформи, яка поєднує засоби моделювання систем управління знаннями, існуючі інформаційні ресурси аналітика із засобами інтелектуального аналізу даних, в системі з однією точкою доступу.

Основний економічний ефект від впровадження корпоративного Web-порталу досягається за рахунок широкого залучення засобів створення та редагування онтології – редактор Protégé з відкритим кодом, а також засобів інтелектуального аналізу даних на основі Web-технологій – RapidMiner.

Надважливою перевагою запропонованої архітектури є зниження трудомісткості і тривалості створення, а також підвищення якості та функціональних можливостей сучасних аналітичних систем.

5.2 Структура семантичної інтерпретації знань при моделюванні сценаріїв АнД

Для певного кола аналітичних ПС, заснованих на онтологічному моделюванні, структура семантичної інтерпретації знань складається з таких основних компонент:

- база фактів, подій, що накопичилися в базі знань ПрО;
- онтології предметної області;
- моделей, які семантично інтерпретують події, що відбуваються в ПрО та впливають на формування сценаріїв АнД, виходячи з настання певних подій;
- типових сценаріїв АнД, виходячи з настання певних подій.

В такої моделі КС база фактів містить множину даних, яка визначає об'єкти та події зовнішнього світу, необхідні для вирішення аналітичних задач. Зрозуміло, що всі факти мають бути семантично інтерпретовані, тобто подані як об'єкти певних класів, визначених онтологією. Інформація в базі фактів має бути

впорядкована за часом та змінами, що дозволяє відслідкувати стан бази на будь-який момент часу. Отже, база фактів, онтологія та моделі у сукупності утворюють базу знань ПС, яка має семантичну інтерпретацію, що потрібна для формування на її основі сценаріїв Анд. В цьому контексті онтологія фактично являє собою таксономію класів ПрО. Це створює можливість однозначного трактування усіх об'єктів з бази фактів, визначення єдиної структури слотів та типів властивостей для них. Крім того, онтологія містить відношення між класами, певну множину аксіом та правила логічного виводу.

Моделі задач визначають яким чином вони вирішуються у системі в контексті визначених правилами семантичних відношень, та дозволяють класифікувати і семантично інтерпретувати об'єкт, встановити відношення з іншими об'єктами. Оскільки КС має справу з розподіленими ІР, база знань містить посилання на зовнішні джерела, наприклад, БД, документи, Web-сервіси та ін. Важливим зовнішнім джерелом для надходження інформації до системи є коментарі та зміни внесені вчителем.

Важливим компонентом КС є сервіси моделювання, які надають додаткові послуги та забезпечують функціонування КС. Визначені такі сервіси:

– **Засоби управління процесом моделювання.** Деякі моделі активуються не іншими моделями, а подіями зовнішнього світу, або певними часовими правилами, або ж за наявності визначених фактів в інформаційній базі. Активація таких моделей здійснюється менеджером запуску моделей. Менеджер виконання використовує модель координації, що визначає які задачі зараз вирішуються, обирає та активізує моделі для їх вирішення, координує та планує використання спільних ресурсів, призупиняє виконання низько пріоритетних задач. Альтернативою до використання моделі координації є боротьба моделей, що працюють як автономні агенти за ресурси. У процесі своєї роботи моделі використовують сервіси, що виконують базові операції у системі.

– **Постачальник інформаційних послуг.** Він приймає від моделі семантично- інтерпретований ідентифікатор потрібного факту і забезпечує

доступ до цього факту, який зберігається у локальній базі знань або зовнішніх джерелах. Реалізація постачальника приховує від моделі механізми пошуку та різні формати даних якими оперують зовнішні джерела. В свою чергу постачальник використовує знання про джерела фактів, методи їх пошуку і оцінки релевантності і базується на відповідних моделях.

– **Брокер взаємодії моделей.** Забезпечує взаємодію моделей, пошук потрібної моделі для активації.

Враховуючи наведені позначення, визначимо тип БЗ як абстрактний тип даних T_{BKN} , поданий кортежем, що містить тип бази фактів T_{BFC} , тип онтології T_{ON} та тип множини моделей \hat{T}_{MD} :

$$T_{BKN} = (T_{BFC}, T_{ON}, \hat{T}_{MD}). \quad (4.1)$$

Екземпляр бази знань t_{BKN} у кожен момент часу подано структурою $t_{BKN} = (t_{BFC}, t_{ON}, \hat{t}_{MD})$, в якій $t_{BFC}, t_{ON}, \hat{t}_{MD}$ відповідають екземплярам бази фактів, онтології, та множини моделей.

База фактів t_{BFC} – це множина фактів про об'єкти та події зовнішнього світу та відношення між ними:

$$t_{BFC} = (\hat{t}_{FC}, \hat{t}_{RF}). \quad (4.2)$$

Елементами множини \hat{t}_{FC} виступають екземпляри даних, що мають тип факту - T_{FC} , $TypeName(T_{FC}) = "Fact"$.

З іншого боку, ці елементи також мають один з типів T_{pid} , який є похідним від T_{FC} : тобто,

$$\exists R_{isa} (TypeEn(T_{pid}), TypeEn(T_{FC})). \quad (4.3)$$

Кожен факт та відношення є семантично- інтерпретованим, тобто його тип визначено в онтології:

$$\begin{aligned} \forall t_{FC}^i : Entity(t_{FC}^i) \neq \emptyset \\ \forall t_{RC}^i : Entity(t_{RC}^i) \neq \emptyset. \end{aligned} \quad (4.4)$$

Тип онтології містить типи множини визначень класів \hat{T}_{CL} , та відношень між ними – \hat{T}_{RCL} :

$$T_{ON} = (\hat{T}_{CL}, \hat{T}_{RCL}). \quad (40.5)$$

Кожен клас T_{CL} визначається набором атрибутів \hat{T}_{SL} , правил \hat{T}_{RU} , обмежень \hat{T}_{CS} що визначені на цих атрибутах:

$$T_{CL} = (\hat{T}_{SL}, \hat{T}_{RU}, \hat{T}_{CS}). \quad (4.6)$$

На рівні онтології j -тий атрибут A_i^j сутності E_i подається атрибутивним відношенням, що поєднує цей атрибут з іншою сутністю E_k :

$$R_{atr}(A_i^j, E_k). \quad (4.7)$$

так, що для кожного екземпляру атрибуту a_i^j його значення визначається екземпляром e_k .

Кожен атрибут задано типом його значень T_{VSL} , множинами правил та обмежень, що діють на рівні атрибуту – $\hat{T}_{RUS}, \hat{T}_{CSS}$:

$$T_{SL} = (T_{VSL}, \hat{T}_{RUS}, \hat{T}_{CSS}). \quad (4.8)$$

Значення атрибуту t_{SL} належить множині допустимих значень $RgSL$:

$$\forall i: t_{SL}^i \in RgSL. \quad (4.9)$$

Обмеження T_{CS} визначає відображення множини значень атрибутів у множину булевих значень $\{true, false\}$:

$$T_{cs}: \hat{T}_{SL} \rightarrow \{true, false\}. \quad (4.10)$$

Правило T_{RU} визначає відображення однієї підмножини значень атрибутів в іншу. З кожним правилом T_{RU}^j пов'язані дві підмножини A_{bas}^j, A_{der}^j атрибутів класу, що не перетинаються:

$$\begin{aligned} T_{RU}^j: A_{bas}^j &\rightarrow A_{der}^j, \\ A_{bas}^j \cap A_{der}^j &= \emptyset. \end{aligned} \quad (4.11)$$

Тип відношення між класами T_{RCL} задано на впорядкованій послідовності типів класів, які воно сполучає: $(T_{CL}^1, T_{CL}^2, \dots, T_{CL}^n)$.

Тип факту бази знань T_{FC} є супертипом для типів класів T_{CL} та відношень T_{RCL} :

$$\begin{aligned} &\exists R_{isa}(TypeEn(T_{CL}), TypeEn(T_{FC})), \\ &\exists R_{isa}(TypeEn(T_{RCL}), TypeEn(T_{FC})). \end{aligned}$$

5.3 Визначення та формування словника відносин та зв'язування класів

Редактор Protégé 5. має можливість опису різноманітних відносин (зв'язків між) класами, що допомагає вирішення задачі несуперечності між класами, що особливо цінне при побудові онтології великого розміру для складної ПрО. В редакторі Protégé 5. реалізовані такі характеристики властивостей.

– Inverse (зворотне) - якщо клас $C_A^{(Ax)}$ пов'язані з класом $C_B^{(Ax)}$, то для цих класів вірне зворотне твердження $C_B^{(Ax)} \rightarrow C_A^{(Ax)}$.

– Functional (функціонал) - якщо класі $C_A^{(Ax)}$ має таку властивість, то для даного класу може існувати тільки один клас $C_B^{(Ax)}$, пов'язаний з класом через цю властивість.

– Inverse functional (зворотне функціональне) - якщо клас $C_A^{(Ax)}$ має таку властивість, то може існувати безліч класів, пов'язаних з класом через Inverse functional властивість.

– Transitive (Транзитивне) - якщо клас $C_A^{(Ax)}$ пов'язаний транзитивною властивістю з класом $C_B^{(Ax)}$, а клас $C_B^{(Ax)}$ пов'язаний теж транзитивною властивістю з класом $C_C^{(Ax)}$, то вірне твердження, що клас $C_B^{(Ax)} \rightarrow C_C^{(Ax)}$ А

– Simmetric (симетричне) - якщо клас А пов'язаний симетричною властивістю з класом В, то клас В теж пов'язаний цією симетрією з класом А.

– Asymmetric (асиметричне) - якщо клас $C_A^{(Ax)}$ пов'язаний асиметричною властивістю з класом $C_B^{(Ax)}$, то він не може бути пов'язаний тією властивістю з класом $C_A^{(Ax)}$.

– Reflexive (рефлексивне) - якщо клас $C_A^{(Ax)}$ має рефлексивну властивість, то він пов'язаний цією ж властивістю з собою.

– Irreflexive (іррефлексивне) - якщо клас $C_A^{(Ax)}$ має іррефлексивну властивість, то він може бути пов'язаний з класом $C_B^{(Ax)}$, але не може бути пов'язаний з самим собою.

– Кожна властивість-зв'язок може мати кілька характеристик.

Відносини $Rel^{(H)}$ (типи зв'язків) між класами та екземплярами онтології встановлюються за допомогою закладки <ObjectProperty>, що розташована на інструментальній панелі редактора Protégé.

За допомогою закладки <ObjectProperty> виконуються наступні дії:


1. Формування підмножини відношень між класами у визначеній предметної області – додавання типів відношень до списку в <owl:topObjectProperty>.

2. Зв'язування класів $C^{(Ax)}$ та формування підмножини аксіом - $Ax^{(S)}$ за допомогою предикатів <owl:topObjectProperty>.

3. Формування відношень між екземплярами класів $Ex^{(C)}$ за допомогою предикатів, які містяться в <owl:topObjectProperty>

5.3.1 Формування підмножини відношень та створення словника

Для формування підмножини відношень в редакторі Protégé 5 необхідно обрати закладку <Object Property>. У робочому полі редактора необхідно зайти у вікно з назвою <Object Property hierarchy> та перейти на стрічку <owl:topObjectProperty>. На інструментальній панелі відкриваються засоби для створення відношень. Для опису нового типу відношень необхідно натиснути

кнопку . Далі у вікні <Create a new OWLObjectProperty> потрібно ввести назву типу відношення, наприклад, <Has_connect_with_device> (Рисунок 5.4). Після виконання відповідних дій, передбачених процесом, у підмножини <owl:topObjectProperty> з'являється новий тип відношень. Повторюючи цей процес для ПрО можна сформувати всю підмножину типів відношень для обраної ПрО.

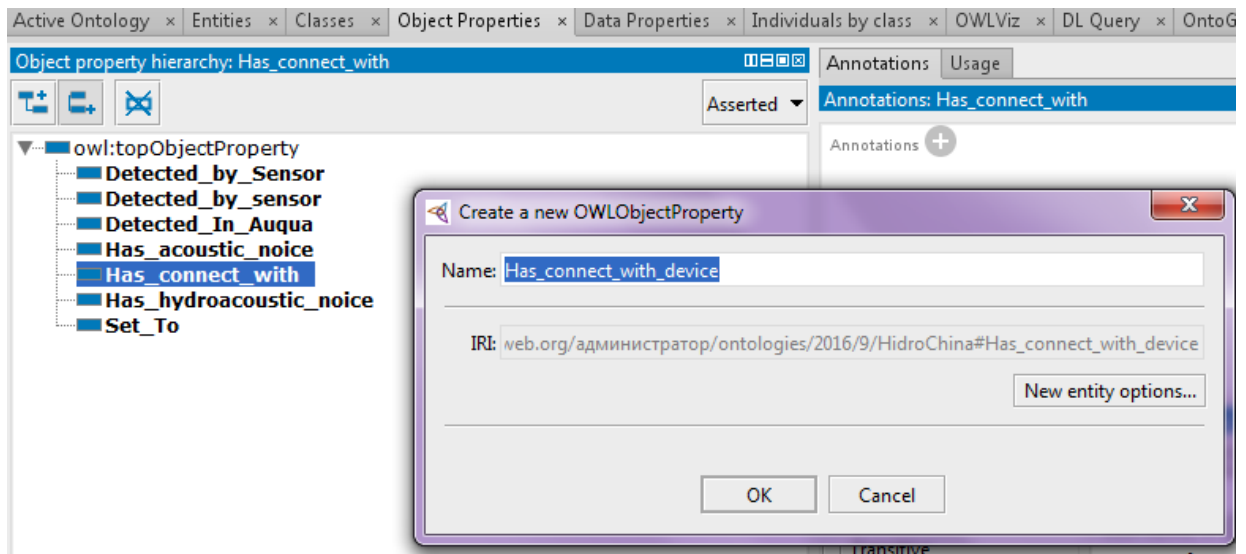


Рисунок 5.4 – Приклад додавання нового типу відносин онтології до словника

Повторюючи цей процес можна сформувати повну підмножину типів зв'язків між класами ПрО.

Слід нагадати, що сформована підмножина типів зв'язків <owl:topObjectProperty> відбиває особисте уявлення розробника про функціонування моделі онтології ПрО на поточний момент часу.

5.3.2 Формування підмножини зв'язків між класами та визначення аксіом

Створений зв'язок між класами описується RDF-графом

<суб'єкт – предикат - об'єкт>

Суб'єкт $C_{si}^{(Ax)}$ (<i>Subject</i>) (клас _i)	Предикат $Rel^{(H)} Rel_{ij}^{(h)}$	Об'єкт $C_{oj}^{(Ax)}$ (<i>Object</i>) (клас _j)
--	--	--

	(дієслово _{ij})	
--	---------------------------	--

Слід нагадати, що у редакторі Protégé 5 для визначення аксіом Про використовуються дві функції:

- Domains (intersection) – домен пересічення, означає, що суб'єкти таких тверджень повинні належати до розширення класу вказаного опису класу. Тобто його властивості можуть розповсюджуватися на кілька об'єктів класу.

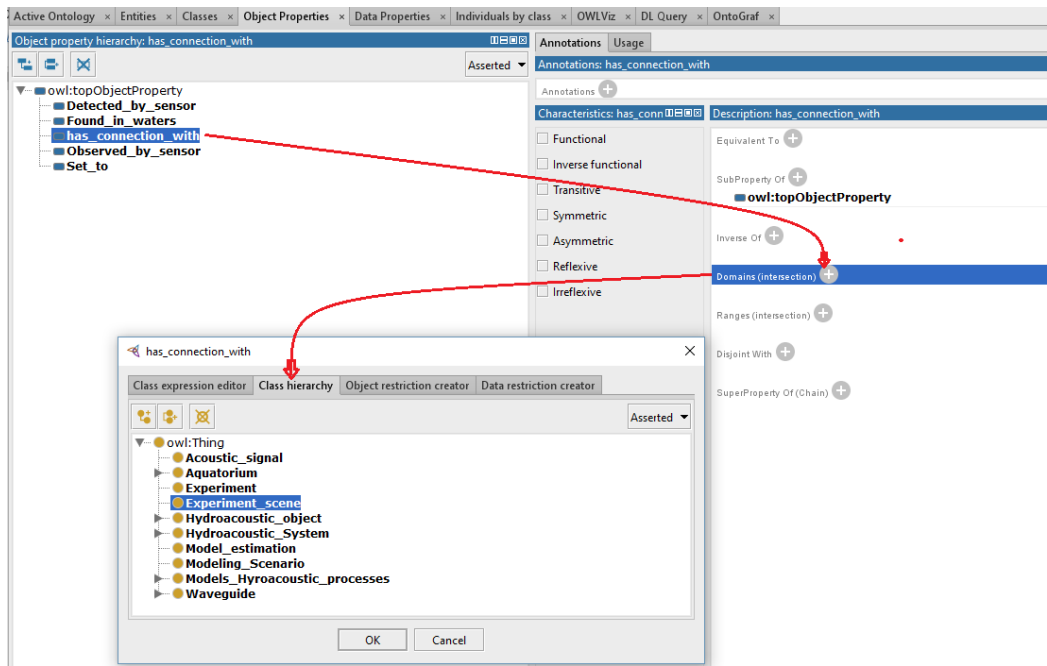
- Ranges (intersection Взагалі сенс процедури <Ranges (intersection)> (Діапазони пересічення), якщо примірник є об'єктом відносини, використовуючи цю властивість, тоді він повинен бути членом цього класу. Кілька записів розглядаються як пересічення.

Для створення підмножини аксіом необхідно обрати закладку <ObjectProperty> в інструментальній панелі редактора Protégé 5. У підмножині <owl:topObjectProperty> слід обрати потрібний тип відношень, який зв'язує два класи, наприклад, <Experiment_scene> пов'язана з <Aquatorium> відношенням <has_connection_with> .

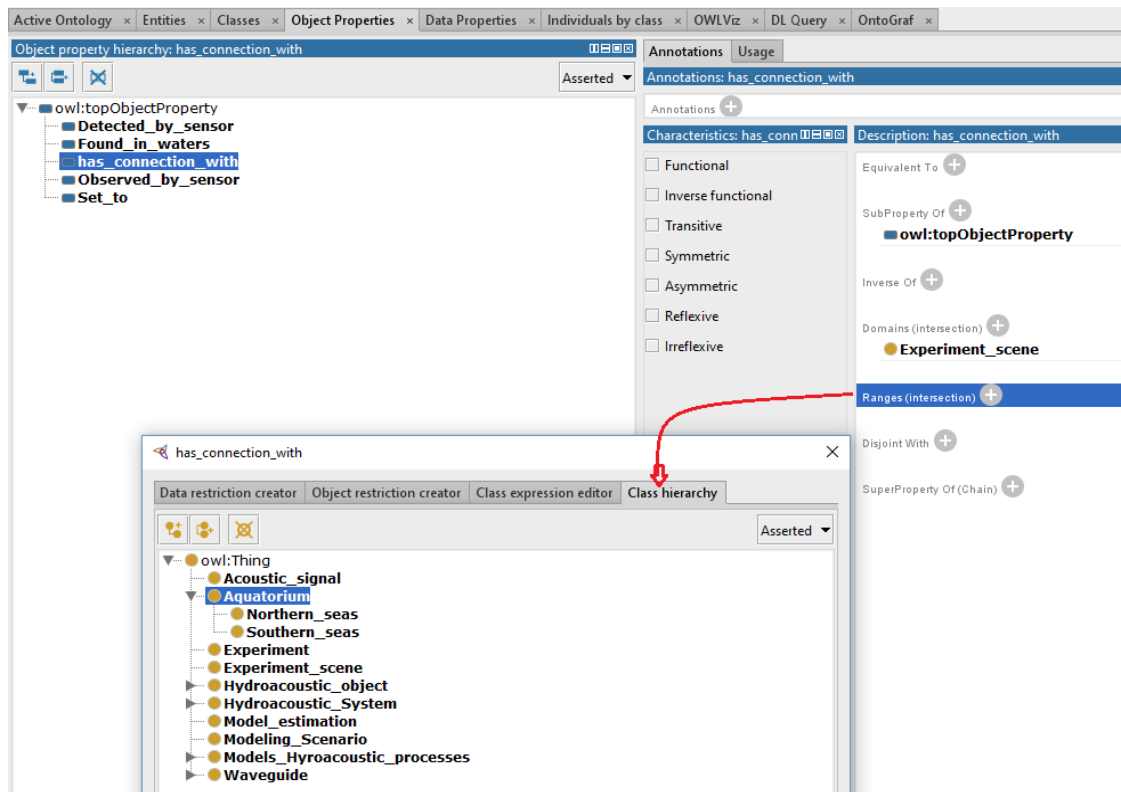
Для цього на вкладці <ObjectProperty> у вікні <Description> потрібно натиснути «Add» (+) біля функції <Domains (intersection)>. При натисканні «Add» (+) розкривається окреме вікно, де потрібно обрати закладку <Class hierarchy> (рисунок 5.5a) та визначити клас, якому надається обрана властивість, наприклад, класу <Experiment_scene>.

Далі потрібно вибрати другий клас натиснувши на "+" біля функції <Ranges (intersection)> у вікні <Description>. Крім того, у вікні, що відкривається, потрібно вибрати клас (<Aquatorium>) і натиснути кнопку ОК (рисунок 5.5b). Після виконання відповідних дій створено нове відношення між класами та зберігається в моделі онтології Subject Domain.

Під аксіомою розуміються твердження, що вводиться в онтологію в готовому вигляді, з якого можуть бути виведені інші твердження.



a)



b)

Рисунок 5.5 – Приклад встановлення зв'язку між двома класами

Аксиоми бувають 3-х типів:

- Відносини типу «дочірній клас». Ці аксиоми мають вигляд « $C \setminus \text{subset } D$ », де C і D - це довільні (можливо складні) класи. Аксиоми цього типу зв'язують два класи (поняття) певними відносинами $Rel^{(H)}$.

– Відносини типу «екземпляр класу». Вони мають вигляд «a: C», де «a» позначає об'єкт, а C - довільний клас.

– Відносини типу «екземпляр властивості». Вони мають вигляд «(a, b): P», де «a, b» позначають два об'єкти, а P - довільне властивість.

Механізм створення аксіом типу «дочірній клас» або «екземпляр класу» аналогічний механізму зв'язування класів.

5.3.3 Формування відношень (зв'язків) між екземплярами класів

Екземпляри класів $Ex^{(C)}$ - це конкретні об'єкти досліджуваної предметної області, які відносяться до певного класу $C_i^{(Ax)}$. Процес формування екземплярів класів виконується через закладку <Individuals by class>, яка записується тегом виду

<owl:NamedIndividual rdf:about=" ...">

Процес створення окремого екземпляру примірника класу складається з наступних кроків:

- 1) Вибір класу, застосовуючи закладку <Classes> або <Entities>.
- 2) Створення окремого екземпляру обраного класу через закладку <Individuals by class>.
- 3) Введення значення або значень екземпляру цього класу застосовуючи функцію <Velu>

Для формування взаємовідносин між двома екземплярами класів в Protégé 5 необхідно перейти на вкладку <Entities> на панелі інструментів. У відкритому вікні <Individuals by type> з'являється ієрархія класів, яка має екземпляри. Далі в ієрархії класів (наприклад, <Helicopter>) потрібно вибрати екземпляр класу і відкрити його (наприклад, <BoeingCH-47>). Після цього в робочому полі відкриваються два вікна <Description> та <Property Assertions>. Для формування відносин між обраними екземплярами потрібно натиснути "+" біля функції <Object Property Assertions>.

Після цього відкриється нова форма для введення двох полів <Enter Object Property name> та <Enter Individual name> (Рисунок 5.6). Наприклад, екземпляр

класу <Helicopter → VoeigCH-47> характеризується високим рівнем акустичного шуму, граничні значення якого фіксуються в екземплярі класу <Noise → Big_acoustic_noise>.

У стрічці <Enter Object Property name> потрібно ввести відношення <Has_acoustic_noise>, а в полі <Enter Individual name> ввести назву екземпляра <Big_acoustic_noise>, а потім натиснути ОК.

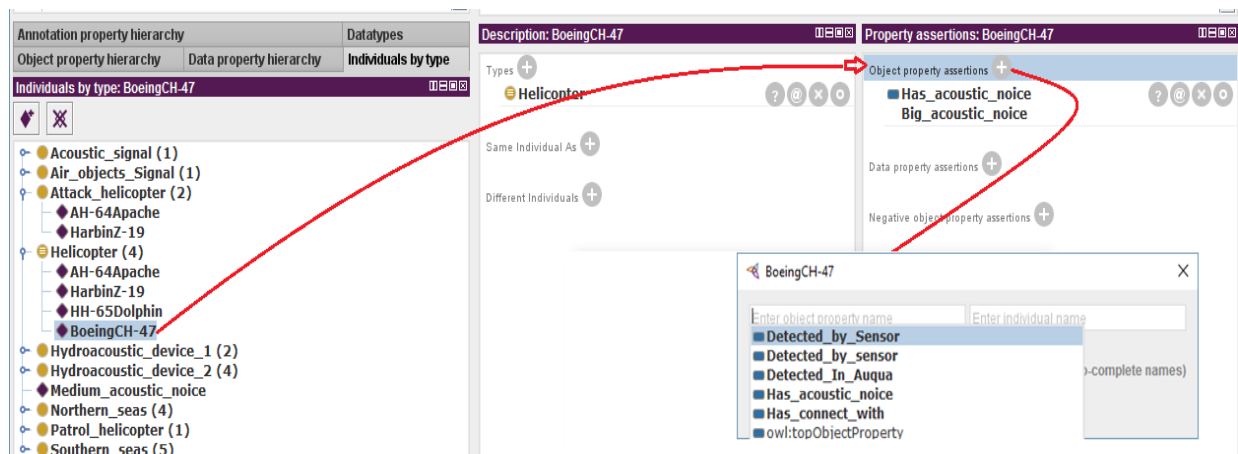


Рисунок 5.6 – Приклад встановлення зв'язку між окремими екземплярами класів

Механізм створення аксіом типу «екземпляр властивості» аналогічний механізму формування відношень між екземплярами класів.

5.4 Висновки до розділу

1. З метою визначення структури та архітектури комп'ютерної моделюючої системи - комп'ютерного середовища для моделювання сценаріїв АнД (КС) визначені вимоги до неї, задачі, які вона повинна виконувати та функції. Запропоновані структура та архітектура КС у вигляді веб-порталу з використанням максимально програмних засобів с відкритим кодом, визначені основні мови, які використовуються для забезпечення процесу моделювання сценарію АнД.

2. Розглянута структура семантичної інтерпретації знань КС, яка застосовується при моделюванні сценаріїв АнД та яка складається з таких основних компонент: база фактів, подій, що накопичилися в базі знань ПрО; онтологія ПрО; моделі, які семантично інтерпретують події, що відбуваються в

ПрО та впливають на формування сценаріїв Анд, виходячи з настання певних подій; типові сценарії Анд, виходячи з настання певних подій. Наявність онтології ПрО в цій структурі створює можливість однозначного трактування усіх об'єктів з бази фактів, визначення єдиної структури слотів та типів властивостей для них.

Визначені такі важливим компоненти КС як сервіси моделювання, які надають додаткові послуги та забезпечують функціонування КС.

Розглянуте вирішення задачі опису різноманітних відносин (зв'язків між) класами шляхом визначення та формування словника відносин та зв'язування класів, що забезпечує несуперечності між класами, та особливо цінне при побудові онтології великого розміру для складної ПрО. Вирішення цієї задачі важливе для забезпечення коректності структури семантичної інтерпретації знань КС.

4. Розглянута модель користувача (аналітика) КС, яка повинна враховувати всі вимоги щодо наявності зручних та зрозумілих сценаріїв роботи, пошуку та обробки інформації з метою отримання нових знань для забезпечення процесу моделювання. Для керування роботою КС визначено базові, службові типи даних.

Розділ 6. Аналіз переваг застосування онтологічних моделей для формування сценаріїв та оцінка якості та ефективності моделювання сценаріїв аналітичної діяльності

Для визначення ступеня досягнення мети дослідження необхідно проаналізувати переваги використання онтологічних моделей для вирішення проблеми побудови програмних систем, здатних до адаптації змін середовища функціонування. Використання онтологічних моделей впливає на різні аспекти процесу створення та експлуатації ПС. Задачу аналізу переваг будемо вирішувати як задачу оцінки ступеня підвищення якості програмного забезпечення в результаті використання онтологічних моделей на всіх етапах його життєвого циклу.

6.1 Оцінка якості моделі онтології та бази знань на якість сценарію аналітичної діяльності

При побудові моделі онтології $Ont(SD)_{IS}$ складної ПрО, як правило, можуть виникнути недоліки у структурі та змісті онтології, які суттєво впливають на якість функціонування створеної онтології в цілому. Тому виникає необхідність запропонувати вимоги, яким повинна відповідати онтологічна модель, що забезпечить логічність висновків, однакову відповідь на однакові, але по-різному сформульовані пошукові запити.

В якості основної вимоги можна розглядати вимоги щодо цілісності онтологічної моделі – $Ont(SD)_{IS}$. Поняття цілісності є інтегрованим та включає такі методи її оцінки:

- Виявлення надмірності онтології - $Ont(SD)_{IS}$;
- Оцінка зв'язності графа онтології - G ;
- Оцінка відсутності взаємозаперечень в онтологічній моделі - $Ont(SD)_{IS}$;
- Оцінка незалежності онтологічної моделі від синтаксису мов дескрипційної логіки для формування правил логічного виводу.

Надмірність онтології – $Ont(SD)_{IS}$ полягає у наявності дублюючих структур: понять і тверджень виражених через зв'язки між цими поняттями. Онтологія вважається внутрішньо узгодженою, якщо виконуються наступні умови:

1) Класи $C^{(Ax)}$, екземпляри класів та їх властивості мають унікальні імена, які не повторюються.

2) Всі підкласи $Csub^{(Ax)}$ в таксономії ПрО пов'язані відношеннями виду «IS-A», а на верхньому рівні таксономії міститься лише один базовий клас – $Csub_i^{(Ax)} \in C_i^{(Ax)}$.

3) Всі класи $C^{(Ax)}$ онтології містять хоча б один об'єкт-екземпляр $Ex^{(C)}$ певного класу – $Ex_i^{(C)} \in C_i^{(Ax)}$, що дає можливість забезпечення функціональності фреймової моделі представлення знань.

4) Аксиоми $Ax^{(s)}$, також, повинні бути логічно узгоджені між собою.

5) Всі обробники повідомлень понять онтології на момент їх виклику мають певні атрибути $T^{(A)}$ (конкретні значення формальних параметрів). Тобто для досягнення результату базовий набір понять – Entities (класів - $C^{(Ax)}$, аксіом - $Ax^{(s)}$), екземплярів - $Ex^{(C)}$), але їх має бути достатньо, щоб створювати складні запити до бази знань - Knowledge base (SD) та здійснювати логічні висновки.

Незважаючи на наявність контрольованої надмірності онтологічна модель - $Ont(SD)_{IS}$ повинна мати властивість розширюваності, що передбачає постійне наповнення онтології з можливістю розширення її словників – таксономії $\langle owl:Thing \rangle$; а також словника взаємозв'язків моделі ПрО $\langle owl:topObjectProperty \rangle$, так і словника властивостей $\langle owl:topDataProperty \rangle$.

Зв'язність графа онтології – G це властивість, яка означає що між будь-якими двома вершинами V онтографа (класами, екземплярами) існує простий зв'язок, тобто поняття онтології логічно узгоджені між собою. З боку машини логічного виводу (Resoner) зв'язність свідчить про те, що всі елементи знаходяться в межах досяжності Resoner і можуть бути використані при формуванні логічного виводу.

Орієнтований граф G називається зв'язним (слабозв'язним), якщо відповідний йому неорієнтований граф є зв'язним. Виходячи з означення зв'язності двох вершин, можна говорити про бінарне відношення зв'язності σ , задане на множині $V(\sigma)$. В основу задачі мінімізації структури графу онтології G покладено типову оптимізаційну задачу теорії графів про відшукування мінімального кістяка, яка полягає у пошуку кістяка мінімальної ваги у зваженому графі. Задача забезпечення узгодженості в структурі графу ефективно розв'язується методом резолюцій.

Оцінка відсутності взаїмозаперечуючих тверджень в онтологічній моделі $Ont(SD)_{IS}$ передбачає необхідність перевірки класів онтології – $C^{(Ax)}$ та відношень між класами – $Rel^{(H)}$ на наявність протиріч між ними.

При внесенні змін до онтології (доповненню її новими сутностями, модифікації, видалення сутностей або їх властивостей) – необхідно виконувати перевірку на цілісність, тобто відсутність в отології дублюючих та (або) взаїмозаперечуючих тверджень. Таку процедуру можливо реалізувати через механізм виявлення протилежних за змістом відгуків за рахунок їх текстового порівняння (співставленням) в випадку послідовного логічного інвертування одного з тверджень-відгуків методом резолюції. В випадку співпадиння прямого та інверсного тверджень отримується сигнал про порушення цілісності та необхідності усунення протиріччя тверджень. В випадку виявлення взаємопротиріч конфлікт може бути усунений шляхом видалення того твердження, важливість якого менша.

Метод оптимізації онтології передбачає задачу оптимізації її структури та змісту:

1) усунення паралельних ребер, дублювання вершин з однаковими параметрами та інших особливостей структури графу онтології, які можуть порушити її цілісність і знижують ефективність функціонування інтелектуальних систем (задача оптимізації структури онтології);

2) усунення циклів в онтологічній моделі – для більш глибокого сприйняття онтологічної моделі кількість циклів повинно бути мінімальним. Рекомендується, щоб такі цикли були відсутні зовсім.

3) оптимізація змістової частини онтології з метою збільшення її швидкодії та інформаційної насиченості за заданих обмежень на фізичний об'єм пам'яті системи.

Розв'язування цих задач рознесено у часі, причому, щоб зберегти цілісність онтології, по-перше виконують її структурну перевірку і лише потім – оптимізацію змістової частини послідовною редукцією її графу до виконання вимог вибраних критеріїв через максимізацію суми ваг вершин та ребер такого графу.

Існує деяка класифікація підходів до оцінки якості онтології:

1. OntoMetric – метрика, яка зосереджується на оцінці досягнення цілей моделі онтології і вивчення її характеристики.

2. OntoClean – формальна оцінка повноти та прозорості таксономії онтології з позиції кінцевого користувача. Виконується із залученням експертів предметної області.

3. EvaLexon – здійснення порівняння термінів зі словника онтології з текстами на природній мові, та визначення наскільки точно висновки про те, чи охоплює дана онтологія більшу частину понять вхідного тексту.

4. Natural Language Application metrics – допомагає оцінити зміст онтології за допомогою різних метрик.

5. OntoManager – метрика, яка має на меті оцінку онтології з позиції її відповідності заданої предметної області.

Останнім часом отримала поширення метрика оцінки онтології з позиції сприйняття її людиною. До таких метрик відносяться глибина онтології [289], яка в свою чергу поділяється на абсолютну глибину, середню глибину та максимальну глибину.

Абсолютна глибина онтології – m обчислюється як сума довжин всіх шляхів онтологічного графа – G , де шляхом – $P \in$ будь-яка послідовність з'єднаних між собою вершин, що починається від кореневої вершини і закінчується листом графа.

$$m = \sum_J^p N_{j \subset p}$$

Де $N_{j \subset p}$ - довжина кожного шляху – j з множини шляхів P у графі - G .

Середня глибина графа онтології. Дорівнює абсолютної глибини поділеній на кількість шляхів в графі.

$$m = \frac{1}{n_{p \subseteq G}} \sum_J^p N_{j \subset p}$$

Де $N_{j \subset p}$ – довжина кожного шляху – j з множини шляхів P у графі - G , а $n_{p \subseteq G}$ – кількість всіх шляхів графу.

Максимальна глибина. Дорівнює максимальній довжині шляху.

Чим більше глибина – m , тим важче граф онтології піддається сприйняттю.

При оцінці глибини моделі онтології з практичної точки зору найбільш значущими можна вважати дві метрики:

1. медіана глибини $m_{50\%} = \overline{N_{j \subset p}}$

де $N_{j \subset p}$ – медіана глибини графа G (можливе значення глибини графа, яке ділить ранжирувану сукупність шляхів графа на дві рівні частини:

50% «нижніх» одиниць ряду даних матимуть значення довжини шляху не більше, ніж медіана – $\overline{N_{j \subset p}}$,

«верхні» 50% - значення довжини шляху не менше, ніж медіана $\overline{N_{j \subset p}}$.

2. 90% line глибини $m = P_{90}(N_{j \subset p})$

де $P_{90}(N_{j \subset p})$ — 90 % глибини графа – G - можливе значення довжини шляху P графа, яка не перевищують довжини 90% шляхів графа.

Наявність циклів в моделі онтології дуже ускладнює її сприйняття. Бажано, щоб їх не було взагалі. Тому при оцінці онтології на циклічність застосовуються дві метрики для вимірювання ступеня циклічності графа:

3. Кількість різних циклів в графі. У хорошій онтології воно має дорівнювати нулю.
4. Кількість вершин, що входять в який-небудь цикл поділене на кількість вершин в графі $m = \frac{N_{v \in C}}{n_G}$

де n_G – загальна кількість вершин графа онтології – G ,

C — множина вершин графа, які входять в хоча б один цикл

$N_{v \in C}$ — кількість вершин графа – G , які входять до будь-якого циклу.

Чим значення метрики циклу менше, тим краще для онтології $Ont(SD)_{IS}$.

Бажане, щоби оптимальне значення метрики циклу m_{opt} дорівнювалося нулю

$$m_{opt} = \frac{N_{v \in C}}{n_G} = 0$$

Для оцінки моделі онтології застосовуються також і інші метрики такі як ширина. Аналогічно до метрики «глибина» метрика ширини онтології теж розділяється на абсолютну ширину, середню ширину та максимальну ширину.

Абсолютна ширина прирівнюється до суми кількості вершин для кожного рівня ієрархії по всіх рівнях

$$m = \sum_j^L N_{j \in L}$$

Де $N_{j \in L}$ – кількість вершин на рівні – j з множини рівнів L у графі – G .

Середня ширина розраховується як абсолютна ширина, поділена на кількість рівнів ієрархії.

$$m = \frac{1}{n_{L \subseteq G}} \sum_j^L N_{j \in L}$$

Де $N_{j \in L}$ – кількість вершин на рівні – j з множини рівнів L у онтологічному графі – G , $n_{p \subseteq L}$ – кількість всіх рівнів графа.

Максимальна ширина прирівнюється до кількості вершин на найбільшому за кількістю вершин рівні. Отже, чим максимальна ширина буде менше, тим краще з точки зору когнітивної ергономіки буде онтологія.

З точки зору когнітивної ергономіки доцільно розглядати не ширину кожного конкретного рівня, а її відношення з сусідніми рівнями

$$m = \frac{1}{n_{L \subseteq G} - 1} \sum_{J=2}^{n_{L \subseteq G}} \frac{N_{j \subseteq L}}{N_{j-1 \subseteq L}}$$

$N_{j \subseteq L}$ – кількість вершин на рівні J

А також медіана відношення ширини сусідніх рівнів

$$m_{50\%} = \frac{\widetilde{N}_{j \subseteq L}}{N_{j-1 \subseteq L}}$$

де $\frac{\widetilde{N}_{j \subseteq L}}{N_{j-1 \subseteq L}}$ – медіана відносини ширини сусідніх рівнів ділить на дві рівні частини: 50% «нижніх» одиниць ряду даних матимуть значення відносини ширини не більше ніж медіана, а «Верхні» 50% - значення відносини ширини не менше, аніж медіана.

Стандарт ISO/IEC 9126 (Software engineering — Product quality) [57] пропонує чіткі методологічні підходи щодо оцінки якості створююмого програмного продукту, включаючи й якість моделі (Quality model). Спираючись на Стандарт ISO 9126 для оцінки якості моделі онтології можна сформулювати такі критерії:

Функціональна придатність онтології є однією з базових характеристик онтології. Функціональна придатність залежить від повноти онтології, тобто наскільки точно вона описує специфіку Про та задач, які у ній виникають. В свою чергу, повнота онтології залежить від вміння давати правильні відповіді на запити до неї, а це залежить від того, чи вміє система оцінювати новизну знань, що пропонується додавати до онтології.

Критерієм якості функціональної придатності побудованої онтології Q_{KB} буде відсоток нетривіальних (ненульових) N_{KB}^{Cor} правильних відповідей на запити до онтології

$$Q_{KB} = \frac{N_{KB}^{Cor}}{N_{KB}} 100\%$$

Зручність використання (usability). Онтологія повинна бути зрозумілою (understandability) для кінцевого користувача, зручною у навчанні (learnability) та використанні, а також привабливою для користувачів (operability).

Коректність (або достовірність) функціонування БЗ – це відсоток достовірно розв’язаних задач. Це основна характеристика якості БЗ

$$Q_{KB}^{cor} = \frac{N_{KB}^{cor}}{N_{KB}} 100\%$$

Зручність супроводу (maintainability) відображається зручністю тестування (testability) та ефективністю виправлення, удосконалення або адаптації структури (analyzability та змісту онтології БЗ залежно від змін у зовнішньому середовищі застосування, а також у вимогах і функціональних специфікаціях замовника.

В разі, якщо онтологічна модель не відповідає критеріям оцінки її якості, здійснюють її структурну перевірку та оптимізацію змістової частини послідовною редукцією її графу до виконання вимог вибраних критеріїв, причому розв’язування цих задач розноситься у часі, щоб зберегти цілісність онтології.

6.2 Оцінка якості та ефективності сценарію, побудованого на онтології

Ефективність застосування запропонованих моделей та методів моделювання сценаріїв АНД визначається оцінкою ефективності самого процесу моделювання сценарію. Визначення такої оцінки з використанням онтологічної моделі для багатofакторного аналізу на відміну від однофакторного потребує побудови алгоритмів, що зможуть надати можливість однозначного вибору

сценарію на основі ієрархічної за структурою моделі онтології ПрО, на основі якої будуються сценарії.

Сьогодні існує багато методів вибору необхідних за якістю та ефективністю сценаріїв з множини можливих чи допустимих сценаріїв, які побудовані на основі аналізу складу та характеру факторів, що впливають на процес планування сценарію. В практиці цього планування виділяються ситуації, що відрізняються числом факторів, на основі яких приймають рішення та визначають принципові відмінності процедур побудови необхідних сценаріїв.

Розглянемо один з прикладів оцінки якості та ефективності сценарію Анд. Одною з досить складних та актуальних задач, що потребують побудови ефективних сценаріїв для їх вирішення, є задача побудови та подальшої оптимізації сценаріїв для аналізу бюджетного процесу на розгалуженій мережі. Доречи задача «оптимізація сценаріїв збору потокової інформації на розгалуженій мережі» є одною з найважливіших задач обробки великих об'ємів даних [1].

Найбільш важливим інструментом для вирішення проблем бюджетного процесу є фінансовий аналіз регіональних бюджетів. В умовах кризисного стану економіки в країні та регіонах, підвищення ролі регіонів та регіональних бюджетів у вирішенні економічних та соціальних задач, проблеми стійкості регіональних бюджетів дуже актуальні.

Фінансовий аналіз бюджету держави дозволяє підводити підсумки за певний проміжок часу, визначати ефективність бюджетних витрат, прогнозувати стан бюджету на наступний квартал або рік та приймати рішення, що сприятимуть покращенню фінансового стану бюджету держави. Одним з показників фінансового стану вважається рівень фінансової стійкості бюджету. Стійкість бюджету дозволяє робити висновки про можливість нормального функціонування суб'єкту публічної влади та про міцність його фінансової основи діяльності.

Головним результатом аналізу фінансової стійкості бюджету є отримання оцінки стійкості для кожного суб'єкту аналізу за певний період часу та висновку про стан стабільності бюджету для цього суб'єкту. На основі цих результатів можна робити загальні висновки про фінансовий стан бюджету держави, робити прогнози про стан на майбутні періоди та приймати рішення щодо дій, які направлені на покращення економічної ситуації для досягнення стабільності та незалежності бюджету держави.

Розробка програмного забезпечення для проведення фінансового аналізу бюджету держави та визначення стійкості бюджету дозволяє значно скоротити час проведення аналізу та полегшує роботу державних установ та органів влади [2].

Загальна структура мережі понять, що визначає онтологію ПрО для вирішення задачі аналізу бюджетного процесу на розгалуженій мережі, досить складна за структурою, оскільки вона може складатися із значної кількості сутностей.

Один з найбільш поширених підходів до вирішення таких задач характеризується тим, що в процесі побудови сценарію на основі онтології ПрО, яка описується графом, при визначенні переваги для оцінки якості та ефективності сценаріїв, як правило, розглядається лише один фактор. При цьому всі інші фактори не враховуються.

Інший шлях відрізняється тим, що сценарій аналізу формується з урахуванням двох або більше факторів, безумовно, які можуть бути враховані в різній мірі.

Процес знаходження якісних та ефективних сценаріїв на базі онтології, та які відповідають вимогам за одним фактором, можна описати за допомогою стратегії, що використовує ієрархічну структуру графу – онтологію ПрО.

Для цієї стратегії можна виділити такі основні критерії оцінки якості сценаріїв:

- 1) по вектору пріоритетів, відповідно до ієрархії онтології;

- 2) з оцінюванням прибутків та витрат від реалізації кожного сценарію;
- 3) з урахуванням експертних оцінок;
- 4) за результатами впливу від реалізації сценаріїв.

Але для побудови необхідних за якістю та ефективністю сценаріїв для багатофакторного аналізу, потребує застосування інших методів аналізу. Одним з таких методів є урахування переваг впливу певних факторів на значення оцінки сценарію. При цьому, вибір сценарію може виконуватися як з визначенням переваги одного із факторів, так і на основі знаходження компромісних рішень між декількома чи всіма факторами.

Автоматизація вирішення таких задач потребує побудови алгоритмів, що зможуть надати можливість однозначного вибору сценарію на основі ієрархічної за структурою моделі онтології ПрО, на основі якої будуються сценарії.

Тому необхідно розробити моделі та алгоритми вибору сценарію на основі існуючої моделі онтології ПрО, яка описується ієрархічною мережею. При цьому ставиться задача вибору найбільш ефективних сценаріїв шляхом виявлення та використання найбільш відповідних сценаріїв до умов конкретного запиту.

На даний час існує декілька методів вирішення задачі побудови сценаріїв на основі моделі онтології ПрО. Ці методи використовують різноманітні математичні моделі для обчислення. Найбільш розповсюдженою є модель побудови сценаріїв на основі опису структури зберігання даних у вигляді розгалужених мереж. На основі цих моделей та їх різноманітних модифікацій будуються сучасні інформаційно-аналітичні та інформаційно-пошукові системи [3]. Така модель притаманна для задач аналізу регіональних бюджетів та враховує їх особливості.

Як вже зазначалося у графічному вигляді онтологія представляється як мережа, вершини якої позначені поняттями (сутностями) ПрО, а ребра позначають зв'язки між ними. Базовими є такі ієрархічні зв'язки типу «клас - підклас» та «частина - ціле», які визначають структуру розгалуженої структури мережі зберігання інформації. Таким чином, онтологія представляє собою опис

структури (таксономії) ПрО, забезпечуючи уявлення сукупності понять і зв'язків між ними.

Для вибору необхідних за якістю та ефективністю сценаріїв (з множини можливих чи допустимих) найбільш ефективними є методи опису та відображення сценарію, засновані на теорії графів. При цьому, як правило, найбільш ефективні – описи сценаріїв у вигляді ієрархічних графів, що дозволяє виконувати оцінку якості сценарію як за критерієм ваги, так і за кількістю ребер, включених до опису сценарію графу.

Відомо чимала кількість методів оцінки онтології. Ці методи розглядають деякі конкретні аспекти, але більшість роблять це неповно. Важливим фактором для оцінки онтології є те, з якої точки зору вона розглядається. Це залежить від того, в якій предметної області вона застосовується і що стоїть в пріоритеті для користувача. Тому необхідно визначитися, який із методів є найбільш відповідним для конкретного випадку.

На сьогодні не має моделей які мали би кількісні та якісні переваги одних над іншими. Тому розробка нових алгоритмів та методів, що базуються на поєднанні у собі переваги різних підходів у побудові моделей, є досить актуальною.

Структурний підхід до розгляду онтології є найбільш відповідним до поставленої задачі, оскільки, представлення структури у вигляді графа, дає можливість виміряти її властивості за допомогою метрики, по якій можна буде визначити її якість та сформулювати рекомендації щодо її поліпшення.

Важливим аспектом структурного підходу є необхідність оцінки ефективності побудови онтологічної моделі на основі оцінки скорочення витрат пошуку по графу, що описує онтологію, у порівнянні з неспрямованим пошуком шляху, який відповідає сценарію, що будується на основі онтології, описаної у вигляді графу.

Для оцінки скорочення витрат пошуку по графу, що описує онтологію, можна провести оцінки ефективності, ґрунтуючись на припущеннях, заснованих

на деяких формальних оцінках різних типів графів, для яких досить просто і точно можна оцінити значення обраних для порівняння характеристик. Ці оцінки можуть бути побудовані на різниці між повним і частковим проходженням по графу, який є графічним відображенням аналізованої онтології при описі сценарію, якій будується на її базі.

Основою для таких процедур може бути раніше отримана інформація про проходження по графу при реалізації інших запитів, що частково збігаються з поточними, тобто інформації, що отримана на попередніх проходженнях по графу. Така інформація зберігається у відповідній базі знань для кожної конкретної онтології, а значить, і для певного відповідного їй графу. При цьому для визначення оцінки ефективності підходу в цілому, не зменшуючи ступеня спільності, можна розглядати деякі характерні типи графів для опису онтології. Це дає можливість визначити граничні оцінки для ефективності підходу в цілому.

Розглянемо спрощену ієрархію (Рисунок 6.1). Для неї характерні і важливі зв'язки тільки на основі відносини предок і нащадок. Найбільш простий випадок – лінійна спрощена ієрархія.

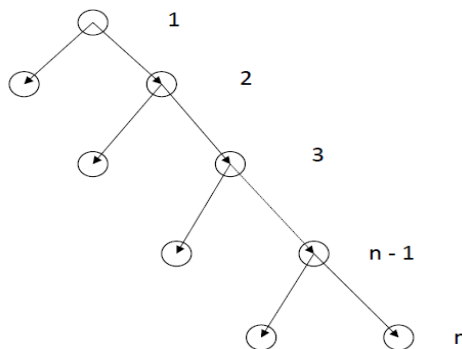


Рисунок 6.1 – Спрощена ієрархія графу онтології

Для неї можна визначити наступні ймовірні співвідношення визначають,

$$P(A) = 1 \gg P(B) = 1/2 * n, \quad (2)$$

де A - апріорно відомий напрямок на основі передісторії пошуку; B - напрямок пошуку без попередньої інформації.

Тоді кількість перевірок відповідності за запитом на графі буде:

$$K(A) = n \text{ та } K(B) = 2 * n$$

Навіть на такому спрощеному графі наявність попередньої інформації значна скорочує час проходження по вузлах графу онтології.

Розглянемо більш складну структуру – парну ієрархічну структуру графа, який описує онтологію ПрО. Цей граф характеризується не більше ніж двома поділами напрямку на кожному з вузлів графа для кожного рівня ієрархії (Рисунок 6.2).

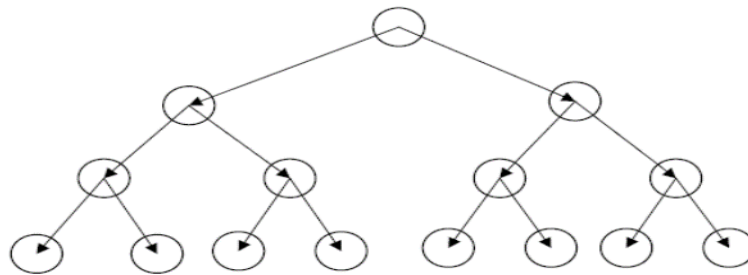


Рисунок 6.1 – Парна ієрархія ($k = 2$) графу онтології

В даному випадку при подвійному розгалуженні графа отримаємо, що ймовірність вибору потрібного напрямку на графі дорівнює $P(B) = \frac{1}{2 * n}$

де n – кількість рівнів у ієрархії.

У разі розгалуження k на кожному рівні $P(B) = \frac{1}{k * n}$

Тоді в загальному випадку

$$1 / k * n < P(B) < 1 / 2 * n .$$

Для дихотомічного ієрархічного графу можна визначити, що

$K_{ur} = n$ – кількість рівнів ієрархії,

$$K_{du} = \sum_{i=1}^n 2^i \text{ – кількість дуг}$$

$$K_{uz} = \sum_{i=1}^n 2^{i-1} \text{ – кількість вузлів}$$

$Kv_m = 2 * n$ – кількість можливих маршрутів.

При n середня кількість невдалих маршрутів прагне до $q = \frac{Kv_m}{2}$.

Тоді використання апріорної інформації підвищує ефективність до q раз.

У загальному випадку можна розглядати таку ситуацію, що для запиту до онтології, що складається з n рівнів, вже відомо на підставі аналізу попередніх запитів збережених в базі знань, результати проходження для m рівнів, де $m < n$.

Тоді ефект від використання раніше отриманої інформації – E_{min} для цього випадку буде становити

$$E_{min} = 2n / (2n - 2(n - m))$$

Таким чином кількість перевірок при проходженні графа можливо знизити в E раз.

Як показано у роботі Міллера, оцінки онтології припускають, що кількість зв'язків у одного концепту в повно пов'язаному графі, що описує онтологію не повинно перевищувати 9 [4]. Таким чином, можна вважати, що у більшості реальних випадків сумарна кількість вхідних і вихідних ребер не перевищує 9.

Виходячи з цього припущення максимальний ефект від використання раніше отриманої інформації можливий шлях побудови сценарію для цього випадку може досягати

$$E_{max} = 9n / (9n - 9(n - m))$$

Для оцінки ефективності побудови моделі онтології можуть бути використані методи, що побудовані на пошуку найкоротших шляхів по графу, який її описує.

У загальному випадку можна записати, що реалізація вибору по графу шляхів досягнення мети T відповідно до запиту:

$$T = \{x / A(x)\},$$

де x - всі можливі шляхи на графі.

$A(x)$ - характеристична властивість, яке відображає суть конкретного запиту і призведе до результатів.

Тоді

$$\forall a \exists x \forall c (c \in x \Leftrightarrow c \leq a),$$

де c - всі шляхи на графі призводять до досягнення мети,

a - мінімальний шлях на графі, відповідний до мети.

Таким чином, завжди є мінімальний шлях на графі, що відповідає цілі – запиту $T_{min} = \min(a)$

Постановка задачі пошуку найкоротших шляхів по графу (Single Source Shortest Path — SSSP) [5] формується виходячи із загального опису характеристик графу моделі онтології.

Нехай заданий граф з вагами дуг і виділеної вершиною-джерелом – u . Послідовність дуг називається шляхом, що йде від вершини u до вершини v . Потрібно для кожної вершини v , досяжною з вершини-джерела u , вказати шлях, який має найменшу можливу сумарну вагу: $P^*(u, v)$

$$f^*(v) = fP^*(u, v) = \min f(P(u, v))$$

Завдання може бути поставлене як для орієнтованого, так і для неорієнтованого графа. Така постановка задачі може бути застосована в обох випадках.

Коректне вирішення завдання має задовольнити принципу оптимальності: якщо шлях є частиною найкоротшого шляху, то є найкоротшим шляхом від джерела u до вершини v .

Принцип оптимальності означає, що для кожної вершини v замість всього найкоротшого шляху досить зберігати його останні дуги.

Розглянемо зважений граф, який складається з n вершин та m дуг з вагами. Для зваженого графу необхідно побудувати найкоротший шлях від вершини (початкова) u до вершини v (заклучна).

Найкоротший шлях від вершини u до вершини v може бути встановлений зворотним шляхом якщо, починаючи з вершини v , проходити ребра в зворотному напрямку до тих пір, поки не буде переглянута вершина u . Ребра можуть бути обрані методом перебору та оцінки ваги дуг для кожної вершини:

$$e_v^* \in \{(u, v) \in E \mid f^*(v) = f^*(u) + f(u, v)\}$$

Для пошуку найкоротших відстаней (v) по набору дуг, необхідно побудувати дерево, на якому виконати пошук з можливістю паралелізації. Найбільш доцільним для задач обробки потокової інформації на розгалуженій мережі є алгоритми пошуку в ширину [6,7]. Цей алгоритм може використовуватися і для вирішення завдання пошуку найкоротшого шляху для усіх пар вершин графа. Для цього алгоритм необхідно запустити для кожної вершини графу.

Для задач оптимізація сценаріїв аналізу бюджетного процесу на розгалуженій мережі, з урахуванням структури опису онтології зберігання даних у вигляді ієрархічного графу (Рисунок 6.3) алгоритм пошуку в ширину дозволяє обчислити найкоротші відстані від виділеної вершини орієнтованого графа до всіх інших вершин. Крім того, пошук в ширину дозволяє вирішувати задачу перевірки досяжності, тобто чи існують шляхи між вершиною джерелом і іншими вершинами графа. Алгоритм заснований на обході вершин графа по рівнях ієрархії. Ідея алгоритму пошуку у ширину дозволяє розробляти різні модифікації алгоритму відповідно до особливостей задач, що вирішується із заданими властивостями та діями, що до нього мають входити.

Для рішення задачі оптимізації сценаріїв аналізу бюджетного процесу на розгалуженій мережі, з урахуванням онтології описують структуру зберігання даних у вигляді ієрархічного графу, пропонується наступний модифікований алгоритм пошуку в ширину, що значно зменшує час пошуку шляхів для побудови сценаріїв збору потокової інформації на розгалуженій мережі.

Нехай заданий граф без ваг з виділеною вершиною-джерелом u . При цьому шляхом $P(u, v)$ між вершинами u і v є множина дуг

$$(U, v_1), (v_1, v_2), \dots, (v_{n-1}, v)$$

Довжиною шляху $d(u, v)$ позначимо число дуг в даному шляху між вершинами u і v . Пошук в ширину знаходить найкоротші шляхи від вершини u до всіх інших вершин.

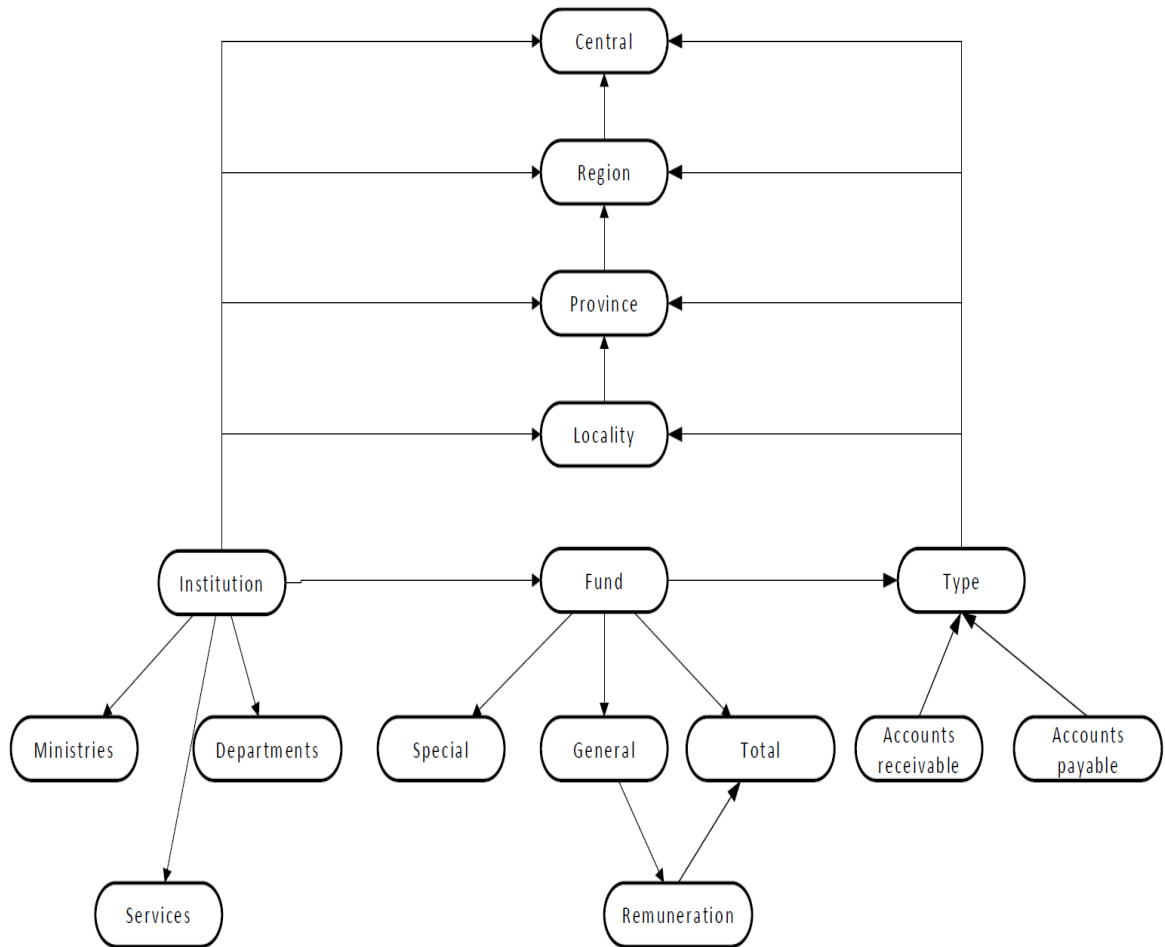


Рисунок 6.3 – Структура розгалуженої мережі аналізу бюджетного процесу регіонів.

На початку роботи алгоритму відстань до вершини-джерела

$$d(u) = 0 ,$$

до інших вершин

$$d(v) = \infty, \forall v \neq u$$

Обчислювальним ядром алгоритму є обхід вершин, суміжних з обраної вершиною v з подальшим додаванням ще не відвіданих вершин у множині P .

Ця операція виконується на кожному кроці для кожної вершини, такої що $v \in V$

Алгоритм послідовно уточнює значення функції $d(v)$. Для врахування особливостей задач побудови сценарію на основі ієрархічної онтології введемо додаткову функцію якості процесу формування сценарію із заданими властивостями та діями:

$F(v_j, v_i) = 1$ якщо v_j відповідає виконанню хоча б одної з дій, які є у складі дій сценарію.

$F(v_j, v_i) = 0$ якщо v_j не відповідає виконанню не одної з дій, які є у складі дій сценарію.

Наприклад, для аналізу бюджетного процесу такою функцією якості може бути фінансова стійкість бюджету є отримання оцінки стійкості для кожного суб'єкту аналізу за певний період часу та висновку про стан стабільності бюджету для цього суб'єкту

Кількість рівнів у графі дорівнює r , а кількість елементів на кожному рівні e .

Матриця $A = \{a_{ji}\}$ має розмірність $r \cdot e$.

Послідовність дій для оцінки якості моделювання сценарію аналізу інформації на мережі може бути наступна:

Крок 1. Вибір на нижньому шарі вершин графа відповідних функції якості.

Крок 2. Аналіз зв'язків виділених вершин з вершинами наступного рівня, який розташований вище по ієрархії моделі.

Крок 3. Збереження ребер графа, які відповідають знайденим зв'язкам, для подальшого аналізу і використання в пошукових операціях.

Крок 4. Повторення ітерацій до виходу на джерело пошуку на верхньому рівні.

Крок 5. Консолідація усіх зібраних ребер і вершин графа, який побудований на підставі функції якості.

Крок 6. Упорядкування всіх шляхів по постійному графу відповідно до функції якості.

Крок 7. Аналіз моделі сценарію на основі вибору найкоротших з мінімальною кількістю ребер на графі.

При цьому структура спрощеного алгоритму аналізу якості сценарію, побудованому на онтології ПрО, набуває такий вигляд (Рисунок 6.4):

1. Встановимо $k=r, i=0, j=0$.

2. $v_i = a_{ki} \quad i=1, \dots, e$
 $v_j = a_{k-1j} \quad j=1, \dots, e$
3. $i=i+1$.
4. $j=j+1$.
5. Визначаємо $F(v_j, v_i)$.
6. Якщо $F(v_j, v_i) = 1$, то перехід на 10, інакше на 7.
7. Якщо $j \neq e$ то на 4, інакше 8.
8. Якщо $i \neq e$ то на 5, інакше 9.
9. Якщо $k \neq 2$ то на 11, інакше 12.
10. $E(v_j, v_i)$ знайдено.
11. $k=k-1$ на 2
12. Процес побудови завершено

В ході роботи алгоритм обходить граф по рівням знизу у верх від останнього рівня до першого.

Отриманий на попередній ітерації результат визначає одну з дуг $e(v_j, v_i)$, яка є основою для пошуку нової дуги на наступному рівні від вершини v_j . Якщо на якомусь шарі, окрім нижнього, не вдається знайти $e(v_j, v_i)$, що відповідає умовам функції якості $F(v_j, v_i) = 1$, То процес повторюється з переходом на один рівень назад і пошук шляху повторюється. При цьому виключається з розгляду дуга, що вибрана невдало. Якщо на якомусь рівні умова взагалі не може бути виконана, то це означає, що сценарій не може бути побудований на цьому графі. Тоді необхідно зробити зміни у онтологію і перебудувати граф, що її описує, а потім провести повторну побудову сценарію на новій онтології.

У результаті виконання дій описаного алгоритму ми отримуємо сценарій, що відповідає заданим умовам, у відповідності до базової онтології, що описана графом. Представлений алгоритм відображає послідовний процес виявлення послідовності дуг, що з'єднують вершини графу від нижнього рівня до верхнього з урахуванням параметрів придатності.

При повторному проходженні графу за близьким за формою запитом ми використовуємо вже наявну модель сценарію, що значно скорочує час побудови

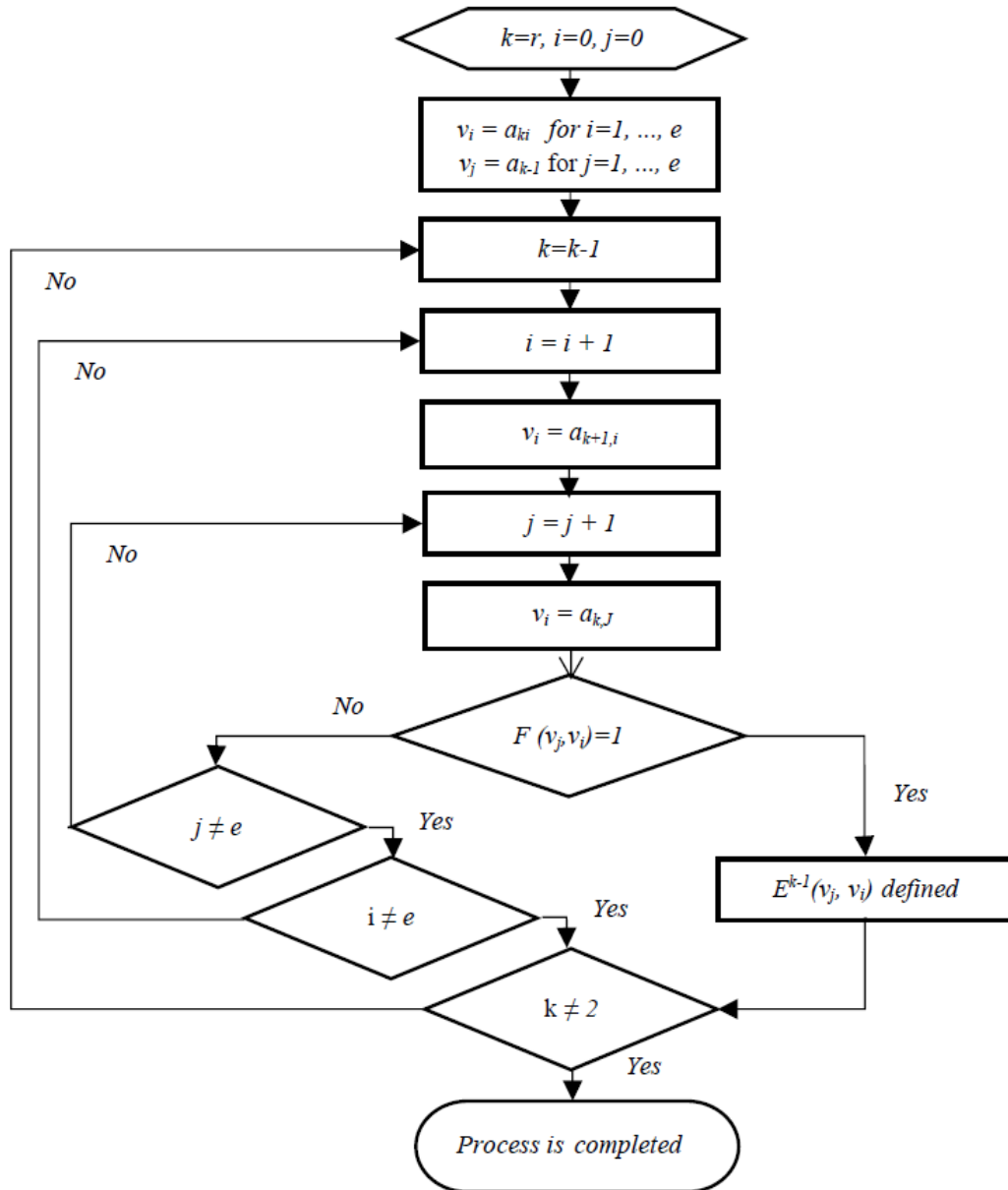


Рисунок 6.4 – Структура алгоритму аналізу якості сценарію

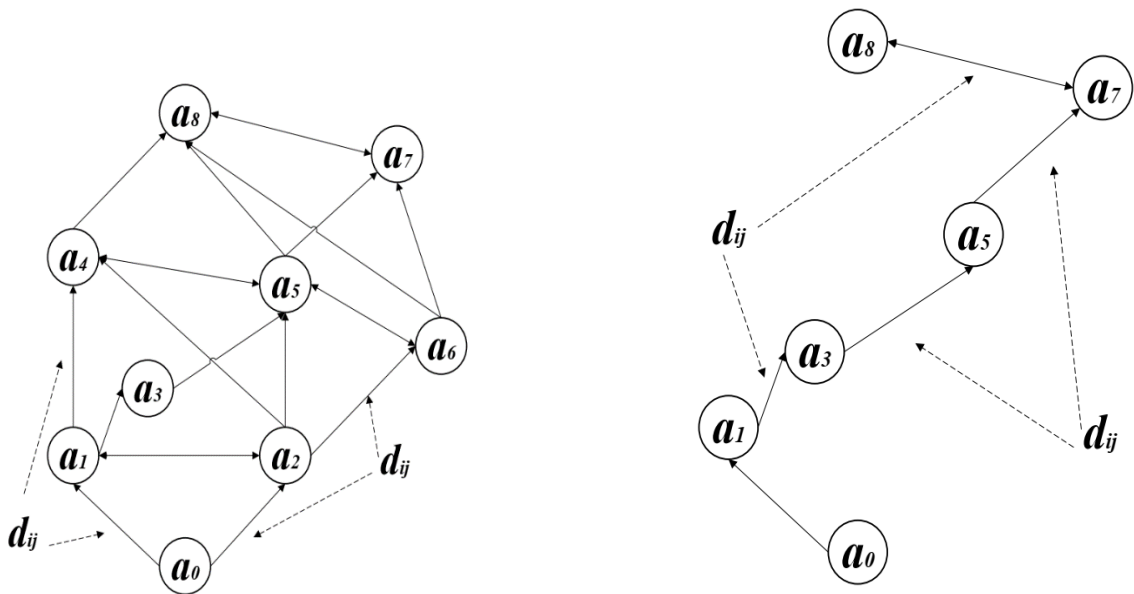
за рахунок зменшення кількості шарів, що розглядаються, але не виключає можливості необхідності перегляду онтології, та її перебудови.

Оцінка ефективності процесу моделювання сценарію аналітичної діяльності як ефективності пошуку оптимального шляху на графі може бути визначена як сукупна оцінка затрат на дії по кожному з вузлів та оцінки затрат

на перехід до наступного вузлу у процесі переходу від начального вузлу до кінцевого з визначенням можливих шляхів переходу (Рисунок 6.5):

$$P_a = \sum_k (t_k + r_k),$$

де t_k – затрати часу на кожному з k вузлів, що відповідають обраному шляху на графі; r_k – затрати ресурсів на кожному з k вузлів, що відповідають обраному шляху на графі; $k \in K$ – номери вузлів обраного шляху на графі.



а) Можливі шляхи

б) Обраний оптимальний шлях

Рисунок 6.5 – Приклад моделі для побудови оптимального шляху від початково a_0 до кінцевого a_8 вузлу по можливим шляхам (дугам) d_{ij}

Відповідно можна визначити критерій оцінки як суму затрат на обробку в ребрах переходу між вузлами як

$$Pd = \sum_{i,j} (T_{ij} + R_{ij}),$$

де T_{ij} – затрати часу на кожному з ij -тому ребрі графу, що відповідають обраному шляху на графі; R_{ij} – затрати ресурсів на кожному з ij -тому ребрі графу, що відповідає обраному шляху на графі; $i, j \in K$ – номери вузлів обраного шляху на графі.

Тоді загальний критерій може бути визначений як сума P_a та P_d .

Сумарний критерій

$$P = P_a + P_d.$$

чи

$$P = \sum_k (t_k + r_k) + \sum_{i,j} (T_{ij} + R_{ij}), \quad k, i, j \in K,$$

K – сукупність індексів, що відповідають обраному шляху.

При цьому як критерій ефективності так і оцінка отриманих результатів в конкретному вузлу, що визначається відповідно до конкретної задачі наявністю необхідних результатів за оцінками аналітиків-експертів:

$$P = \sum_k (c_t t_k + c_r r_k) + \sum_{i,j} (c_T T_{ij} + c_R R_{ij}), \quad k, i, j \in K,$$

де c_t – коефіцієнт впливу затрат часу кожному з k вузлів,

c_r – коефіцієнт впливу затрат ресурсів на кожному з k вузлів,

c_T – коефіцієнт впливу затрат часу на кожному з ij -тому ребрі графу,

c_R – коефіцієнт впливу затрат ресурсів на кожному з ij -тому ребрі графу.

$$c_t, c_r, c_T, c_R \geq 0$$

В залежності від відзначень цих коефіцієнтів аналітиком визначається той чи інший тип задачі пошуку оптимального шляху: за критерієм часу, за мінімальністю затрат ресурсів чи з використанням комбінованого критерію. Обрання загального сумарного напрямку шляху може визначатися наявністю найбільш ефективного напрямку з точки зору значення критерій V на кожному послідовному кроці вибору напрямку на графі.

Тригер оцінки результатів на кожному кроці визначає: чи є покращення результатів, чи ні.

$Q_i > 0$ – вузол розглядається, як можливий для порівняння та вибору.

$Q_i = 0$ – вузол a_i не розглядається, як можливий для порівняння та вибору.

Цю оцінку має виконувати аналітик на основі отриманих результатів.

Для кожного рівня у порівнянні розглядаються ті напрямки(ребра) та вузли, що відповідають $Q_i > 0$. Інші напрямки (ребра) та вузли з подальшого розгляду вилучаються.

Значення $Q_i > 0$ може визначатися за певними обраними мірами аналітиком, як оцінка збільшення результатів рішення аналітичної задачі, що вирішується.

При відсутності наявних та обґрунтованих методів оцінки $Q_i = 1$ – вузол a_i розглядається, як такий, що покращує рішення задач аналітиком. $Q_i = 0$ – вузол a_i не розглядається, як можливий для порівняння та вибору, як такий, що не надає нових знань чи нових результатів для аналітика.

Без зменшення ступеню загальності, модель, де розглядаються затрати, пов'язані як з сукупною оцінкою затрат на роботи по кожному з вузлів, так і оцінки затрат на перехід до наступного вузлу у процесі переходу від начального вузлу до кінцевого, може розглядатися, як модель з урахування тільки затрат на ребрах графу.

При цьому затрати в вузлах будуть включені до затрат ребер графу:

$$P_{ij} = t_j + r_j + T_{ij} + R_{ij}, \quad i, j \in K.$$

Коли є більш якісний за обраним критерієм (наприклад за часом отримання і обробки інформації) у порівнянні з попереднім (типовим) сценарієм ланцюжок дій, то він буде визначений для заміни типового сценарію на новий. У протилежному випадку можна стверджувати, що раніше отриманий сценарій і є більш ефективним. Таким чином проводиться перевірка побудованого сценарію на ефективність. У результаті виконання дій описаного алгоритму ми отримуємо сценарій, що відповідає заданим умовам, у відповідності до базової онтології, що описана графом.

В 3.4 вже визначалися необхідна і достатня умови стійкості аналітичного процесу, який відображує послідовності дій аналітика щодо досягненню мети дослідження. Області стійкого аналітичного процесу і нестійкого можна також представити в декартовому просторі в координатах $\lambda(G)$ та $\chi(G)$ (Рисунок 6.6). Залежно від параметрів $\lambda(G)$ та $\chi(G)$ графа структури, буде визначатися точка на площині, що характеризуватиме стан аналітичного процесу.

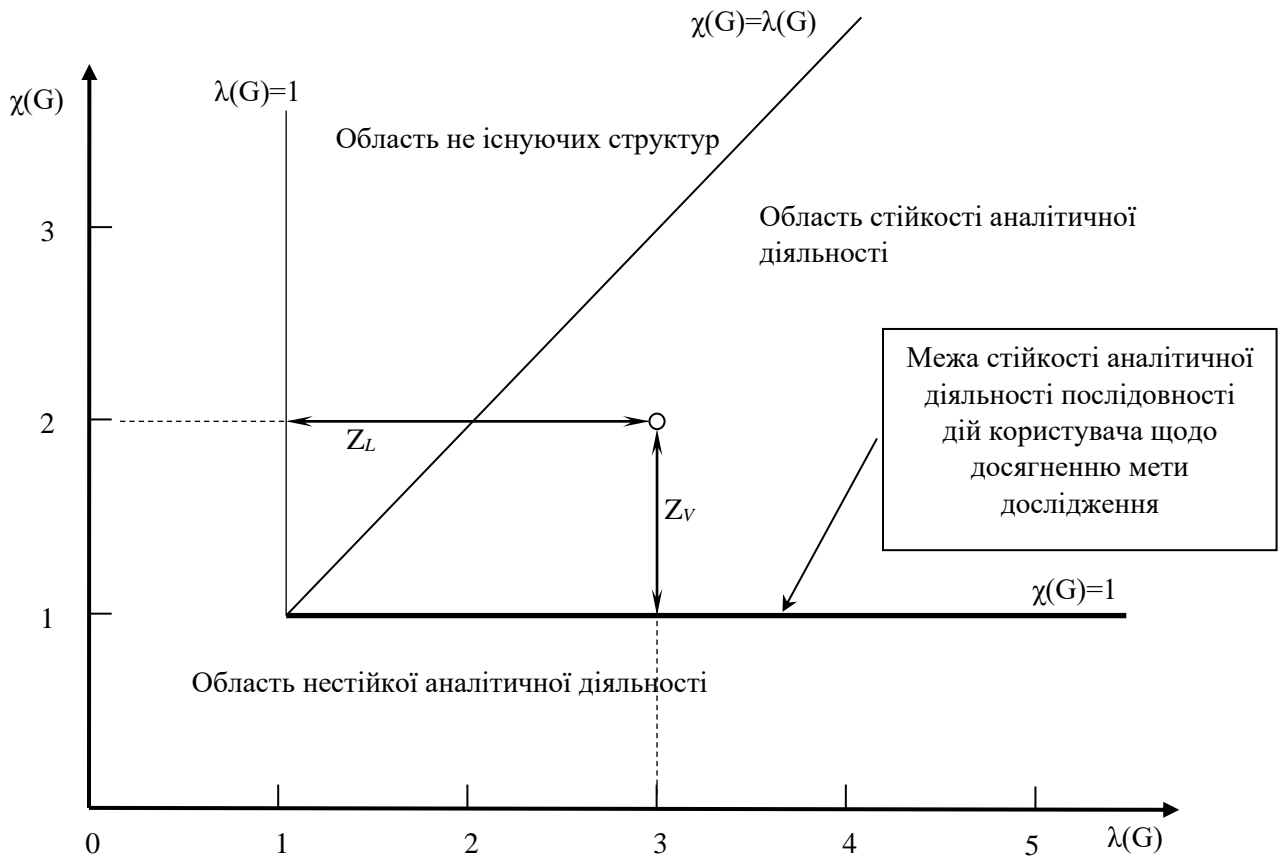


Рисунок 6.6 – Герметична інтерпретація областей стійкості аналітичного процесу

У графічному уявленні межею стійкості аналітичного процесу буде множина точок, що лежать на двох прямих $\lambda(G)=1$ та $\chi(G)=1$. Важливим є визначення наскільки далеко аналізована структура послідовності дій користувача для досягнення мети дослідження знаходиться від межі стійкості аналітичного процесу або, з іншого сторони, який запас вказаної стійкості аналітичного процесу. При цьому запас стійкості визначається відповідно до зв'язності структури. У цьому випадку запас характеризується числом неактуальних дій користувача щодо досягнення мети дослідження або відмови вузлів, які можуть привести аналізовану структуру до межі стійкого аналітичного процесу.

Запас стійкості аналітичного процесу можна визначити наступним чином:

– реберний запас стійкості аналітичного процесу – число Z_L , дорівнює відстані від точки з координатами $(\chi(G), \lambda(G))$ в просторі областей стійкості

аналітичного процесу до прямої $\lambda(G)=1$, що описує уявну частину межі стійкості аналітичної діяльності:

$$Z_L = \lambda(G) - 1.$$

– вершинний запас стійкості – число Z_V , що дорівнює відстані від точки з координатами $(\chi(G), \lambda(G))$ в просторі областей стійкості аналітичної діяльності до прямої $\chi(G)=1$, що описує межу:

$$Z_V = \chi(G) - 1.$$

Суть запасу стійкості аналітичного процесу полягає в наступному:

Z_V – характеризує число вершин графа, після видалення яких разом з інцидентними їм ребрами структура переходить з стійкого стану на межу стійкості. Ця величина характеризує граничне число відмов вузлів, що містять певні знання, які можна компенсувати за рахунок використання альтернативних вузлів;

Z_L – характеризує число ребер графа, після видалення яких, структура переходить із стійкого стану на межу стійкості аналітичного процесу. Ця величина характеризує граничне число видалених ребер графа, що відображає послідовності дій користувача по досягненню мети дослідження, які можна парировати за рахунок використання альтернативних маршрутів.

Таким чином, на підставі ознак стійкості аналітичного процесу після визначення введених в розгляд параметрів можна визначити стан аналітичного процесу, а саме знаходження системи у стійкому або нестійкому стані. Міра стійкості аналітичного процесу визначається запасом стійкості, який можна визначити аналітично використовуючи запропоновані залежності.

Для підтвердження вище описаного підходу до визначення оцінки якості та ефективності сценаріїв АнД було проведено ряд тестувань на згенерованих тестових прикладах за використання розробленої тривірневої комп'ютерної моделюючої системи (інтегрованого програмного інструментального середовища) для тестування і оцінки запропонованого теоретичного підходу моделювання сценаріїв АнД, яка ґрунтується на запропонованих моделях,

методах та алгоритмах моделювання. Було розглянуто більш ніж двохсот тестових прикладів для оцінки кількості обчислень та порівнянь при виконанні пошуку оптимального шляху на графічних структурах, тобто на графах, які є відображенням первинної онтології ПрО.

Проведення такого тестування та аналізу результатів цього тестування необхідно було для підтвердження як результативності самого підходу, так виявлення залежності ефективності запропонованого алгоритму послідовного уточнення сценарію у порівнянні з повним аналізом графу онтології.

При підготовці прикладів для тестування та аналізу розглядалися два групи прикладів, які побудовані на різних за структурою, але аналогічних за побудовою підходах.

У якості кількісної оцінки ефективності роботи алгоритму було використано кількість порівнянь, які необхідно провести для обрання оптимального шляху на графах, яке визначається кількістю вузлів на кожному рівні ієрархічної структури тестового прикладу, у яких необхідно зробити порівняння для вибору найбільш відповідного напрямку розвитку сценарію. При цьому кожне з порівнянь оцінювалось випадковим чином від 1 до 10 балів. Таким чином моделювався можливий час виконання робіт у кожному з вузлів.

При цьому розглядалися моделі генерації прикладів у двох напрямках. По перше, для аналізу по кількості вузлів на кожному рівні ієрархічної структури, а по друге, для аналізу залежності кількості порівнянь від кількості рівнів в ієрархічній структурі у згенерованих прикладах графів.

По своїй структурі усі згенеровані графи мали як мінімум один можливий оптимальний шлях та ряд шляхів, що не приводили до кінцевого вирішення поставленої задачі побудови сценарію.

На рисунку 6.7 представлені результати побудови оптимальних сценаріїв на тестових прикладах з усередненням результатів на двохстах тестових прикладах. Були розглянуті приклади з кількістю рівнів 4 та 10 та кількістю вузлів на кожному рівні 5,10,15,20,25 та 30.

Аналіз результатів показує значне скорочення затрат при використанні типових сценаріїв (позначено на графіку ТС). Крім того, як видно з графіку, перевага використання алгоритму на основі типового сценарію значно зростає при збільшенні кількості вузлів на кожному з рівнів. Таким чином чим більша за складністю структура графу онтології, який є основою для побудови сценарію, тим більш ефективно використання послідовного уточнення сценарію. Так при максимальній кількості вузлів скорочення часу моделювання в умовних одиницях складає у середньому більше як в п'ять разів.

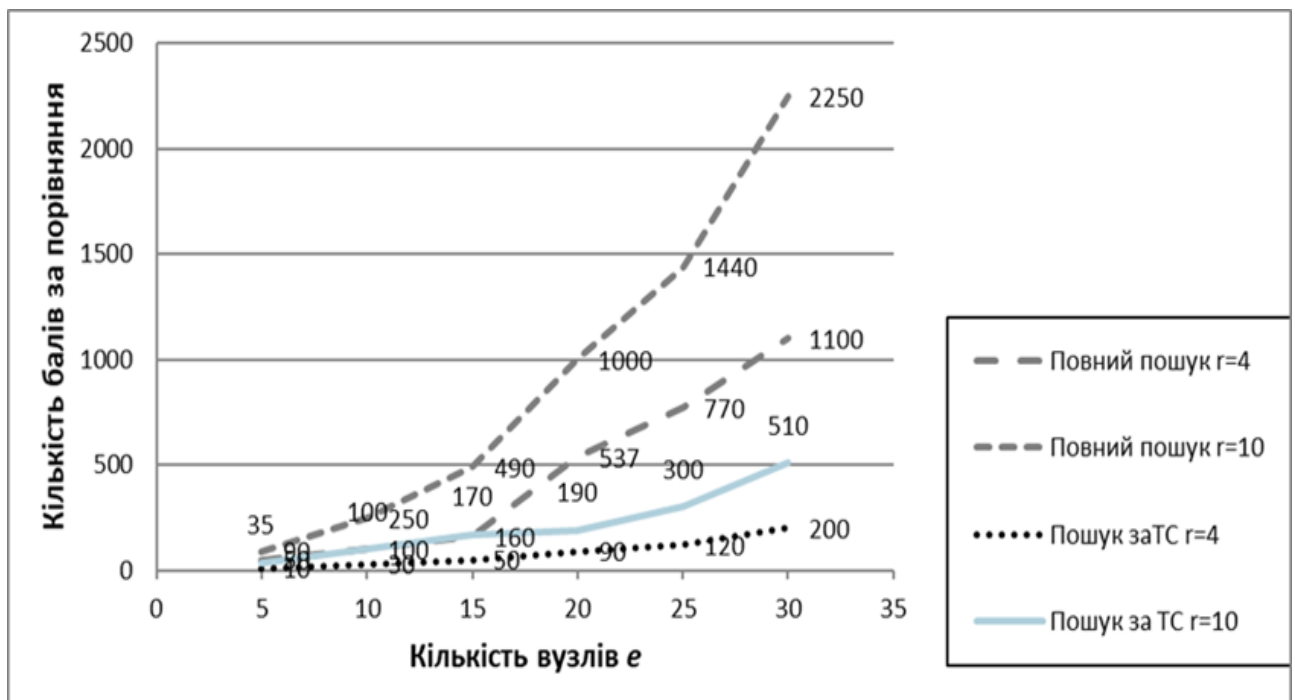


Рисунок 6.7 – Порівняння результатів використання алгоритму послідовного уточнення сценарію з повним аналізом графу за кількістю вузлів.

За аналогічною схемою було проведено друга частина тестування на групі з теж двохстах тестових прикладів, які були побудовані для виявлення та підтвердження залежності ефективності між кількістю рівнів та оцінкою кількості балів для алгоритму послідовного уточнення сценарію у порівнянні з повним аналізом графу.

При цьому були розглянуті (Рисунок 6.8) приклади з кількістю рівнів 10 та 20 та кількістю вузлів на кожному рівні 4,8,12,16,20 та 24. Кількість балів у згенерованих прикладах теж була визначена випадковим чином від 1 до 10. Були

розглянути тільки ті приклади, що мали не менш одного шляху, що дає можливість побудувати необхідний сценарій.

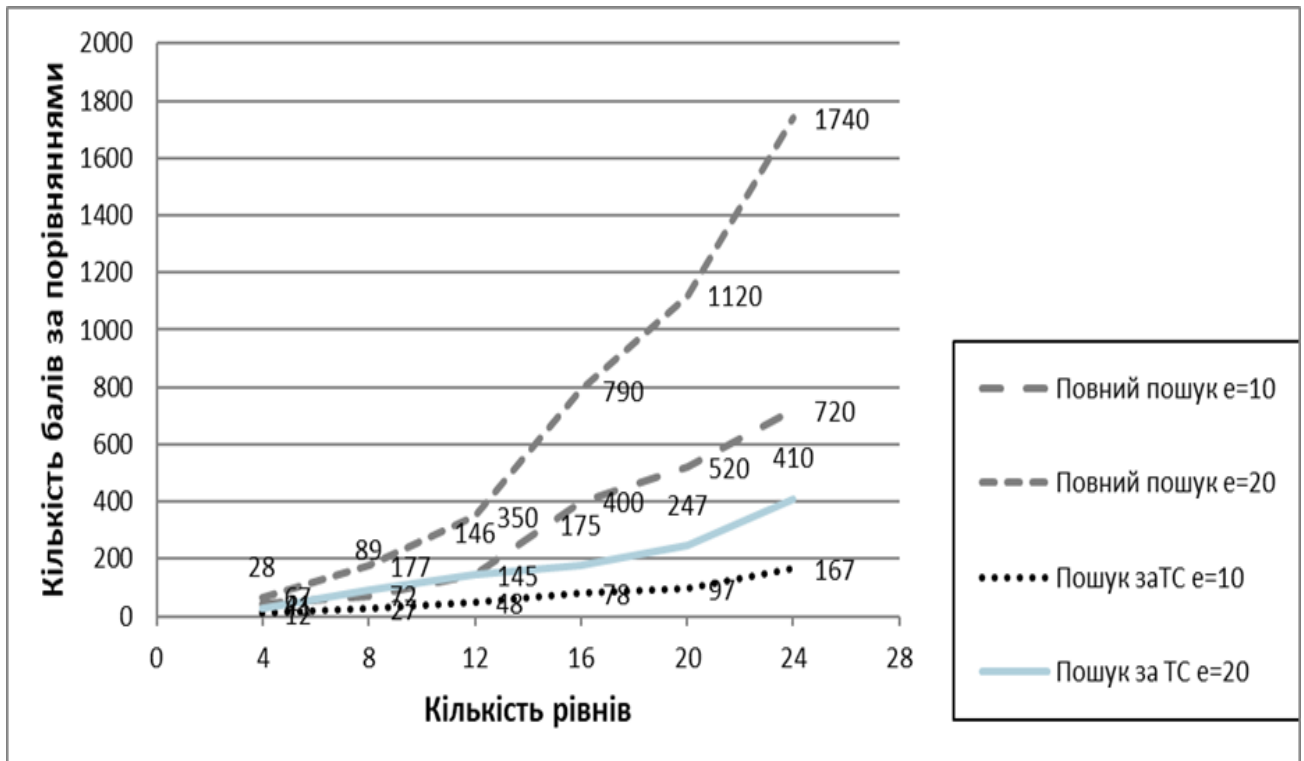


Рисунок 6.8 – Порівняння результатів використання алгоритму послідовного уточнення сценарію з повним аналізом графу за кількістю рівнів.

Аналіз результатів показує значне скорочення затрат при використанні типових сценаріїв (позначено на графіку ТС), що особливо дає, як видно з графіку, перевагу використання алгоритму на основі типового сценарію при збільшенні кількості рівнів. Так при максимальній кількості вузлів скорочення часу моделювання в умовних одиницях складає більше як в чотири рази. Чим більш складна структура графу онтології ПрО, який є основою для побудови сценарію, тим ефективність використання послідовного уточнення сценарію при його моделюванні є значною.

6.3 Дослідження впливу виразних можливостей мов SPARQL та DL – Query на ергономічність програмних засобів, які забезпечують моделювання сценарію АнД

Ефективність здійснення аналітичної діяльності користувачем (аналітиком) залежить і від того, наскільки зручно йому працювати з

програмними засобами, які йому надаються, тобто наскільки вони є ергономічними - зменшують фізичну та психологічну втому, не потребують проходження спеціального навчання. Будь-якій користувач повинен мати можливість інтуїтивно пов'язувати дії, які потрібно виконати, з об'єктами, що він бачить. Наприклад, надзвичайно важлива ергономічність мов, які використовують для формування запитів до побудованої онтології сценарію, оскільки вона використовується як інструмент, за допомогою якого користувачі можуть взаємодіяти з онтологією та даними.

Порівняльний аналіз виразних можливостей **SPARQL** та **DL-Query** виконувався на прикладі пошуку знань в онтології «Моделюючого комплексу для відпрацювання алгоритмів і програмного забезпечення виявлення, класифікації та визначення параметрів руху морських об'єктів» показав, що на запит типу:

Які Hydroacoustic_System встановлені в Yellow_Sea?

DL -Query дозволив виявити два об'єкти, які встановлені в Yellow_Sea (Рисунок 6.9а).

Аналогічний SPARQL-запит теж виявив об'єкти, встановлені в Yellow_Sea (Рисунок 6.9в).

Мова SPARQL – це класична мова запитів до онтологічної моделі знань, яка представлена у вигляді RDF-графу. З деякими припущеннями можна вважати, що мова SPARQL це аналог мови SQL для реляційних баз даних.

SPARQL-запит являє собою набір графових шаблонів, з якими порівнюються RDF-дані. Результати SPARQL запитів можуть бути повернуті або відтворені у різних форматах:

- XML SPARQL визначає XML-словник для повернення таблиць результатів.
- JSON. JSON "порт" XML словник, особливо корисний для Web-додатків.

– RDF. Деякі пункти результатів SPARQL викликають відповіді RDF, які, в свою чергу, можуть бути серіалізовані різними способами (RDF / XML, N-Triples, Turtle тощо).

DL query:

Query (class expression)

Hydroacoustic_System and SetTo some {Yellow_Sea}

Execute Add to ontology

Query results

Subclasses (1 of 1)

- owl:Nothing

Instances (2 of 2)

- Device_41
- Device_42

Query for

- Direct superclasses
- Superclasses
- Equivalent classes
- Direct subclasses
- Subclasses
- Instances

а) DL-запит

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX h: <http://www.semanticweb.org/администратор/ontologies/2016/09/HidroChina#>
SELECT ?pr ?Hydroacoustic_System
WHERE
{
  ?pr h:SetTo ?Hydroacoustic_System
  Filter regex( str(?Hydroacoustic_System), "Yellow_Sea")
}

```

	pr	Hydroacoustic_System
Device_41		Yellow_Sea
Device_42		Yellow_Sea

в) SPARQL-запит

Рисунок 6.9 – Порівняння результатів логічного висновку мов запитів DL та SPARQL при зверненні до бази знань онтології моделюючого комплексу

– HTML. При використанні інтерактивної форми для роботи з запитом SPARQL. Часто результати запитів реалізується шляхом застосування перетворення форми подання XSL в форму XML.

Однією з суттєвих переваг мови запитів SPARQL є наявність розвинутих механізмів фільтрації для формування складних запитів. Це дозволяє суттєво зменшити кількість надлишкової інформації в відповіді. Наприклад, потрібно знайти Hydroacoustic_object рівень Noise_emission перевищує 60 db:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?Hydroacoustic_object ?Noise_emission

```

WHERE {
    ? Hydroacoustic_object: Helicopter ;
        rdfs:label ? Hydroacoustic_object;
        prop: Noise_emissionEstimate ? Noise_emission.
    FILTER (?Noise_emission > 60 db) .
}

```

Наважливою перевагою мови запитів SPARQL є принципова можливість знаходження аналогів в багатодоменному веб-просторі за рахунок їх пошуку різними мовами.

Недоліком мови SPARQL, с точки зору кінцевого користувача, є необхідність знання синтаксису SPARQL та ретельного написання назв сутностей і предикатів. Якщо індивідуальні знання мови SPARQL користувача не належному рівні, це дуже ускладнює коректне формування запитів до онтології. Оскільки синтаксичні помилки призводять до не виконання запиту і, фактично, до повної неможливості використання онтології.

Мова запитів DL-Query в інструментальному середовищі Protégé 5.2 мова запитів реалізована в рамках програми reasoner HermiT. DL-Query є повною версією мови OWL 2 Direct Semantics як стандарту World Wide Web Consortium (W3C) [HermiT: An OWL 2 Reasoner]. HermiT також підтримує правила DL-безпечної SWRL та графі опису [24], що дозволяє точно описувати довільно пов'язані структури онтології (багато доменні онтології).

Програма Reasoner HermiT сприймає онтологію як набір TBox(SD) – опис ієрархії класів <Class hierarchy> (понять предметної області та їх екземплярів), а також ABox(SD) – множин властивостей, якими описуються класи та їх екземпляри (за допомогою словників <owl:topDataProperty> та <owl:topDataProperty >). Така реалізація програми логічного виводу суттєво спрощує сам процес формуванні DL-запиту, оскільки у користувача з'являється можливість скористатися інтелектуальним інтерфейсом, який дозволяє обирати потрібні об'єкти, властивості або логічні функції із відповідних словників: класів

<Class hierarchy>, <owl:topDataProperty> та <owl:topDataProperty > (Рисунок 6.10).

Наявність інтелектуального інтерфейсу при написанні запиту (у вигляді триплету) суттєво скорочує кількість помилок користувача та позитивно впливає на психологічне сприйняття результатів логічного виводу.

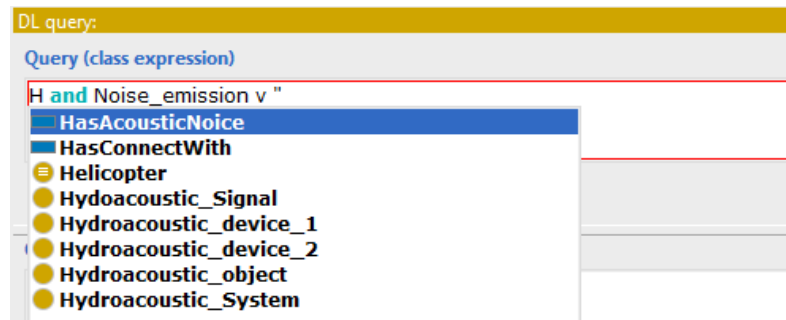


Рисунок 6.10 – Приклад застосування інтелектуального меню при формуванні object DL- запиту

Крім того, при виборі DL функції над триплетами теж є можливість скористатися меню вбудованих DL функції (Рисунок 6.11).

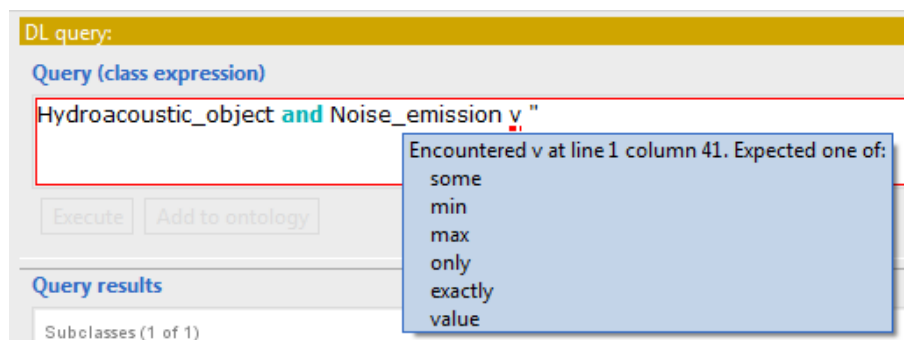


Рисунок 6.11 – Приклад інтелектуального меню вбудованих DL-функції при формуванні DL- запиту

Більш того, програма Reasoner HermiT підтримує алгоритм HyperTableau-based, реалізоване на Java, що дає можливість використовувати результати в різноманітних застосунках.

Розглянуті мови запитів DL query, SWRL та SPARQL query реалізують функціональність, яка декларується мовою OWL 2.

Кожна з розглянутих мов дозволяє генерувати n-мірні триплети при формуванні запиту, спираючись на застосування логічних функцій. Це надає можливість формування доволі складних запитів та знаходження неявних

зав'язків між фактами та подіями, що описуються онтологією. Крім того, розглянуті мови використовують різні методи фільтрації, що зменшує кількість надмірної інформації у відповіді та спрощує її аналіз користувачем.

Однак, як показують дослідження мова DL query характеризується більшим рівнем виразності, оскільки вона, окрім традиційних можливостей запитів типу SELECT (як у мові SPARQL) також застосовує апарат логічних функції синтаксису Manchester OWL при формуванні запиту. Крім того, реліз мови DL query програми Reasoner HermiT пропонує інтелектуальний інтерфейс, який вичитує онтологію ПрО та дозволяє обирати потрібні об'єкти, властивості або логічні функції із відповідних словників: класів <Class hierarchy>, <owl:topDataProperty> та <owl:topDataProperty >. Це додає значні переваги у порівнянні із SPARQL query та SWRL.

6.4 Висновки до шостого розділу

1. Проаналізовано фактори, що впливають на якість та ефективність застосування запропонованих моделей та методів моделювання сценаріїв АнД. По-перше, досліджені наявні метрики оцінки онтологічної моделі, яка застосовується для моделювання сценарію, з урахуванням методологічних підходів щодо оцінки якості програмного продукту Стандарту ISO/IEC 9126 (Software engineering — Product quality), включаючи й якість моделі (Quality model) та визначені критерії оцінки якості моделі онтології. Запропоновані вимоги, яким повинна відповідати онтологічна модель, що забезпечить логічність висновків, однакову відповідь на однакові, але по-різному сформульовані запити, насамперед вимоги щодо цілісності онтологічної моделі. Визначені умови, при яких оцінки онтологічної моделі свідчать про її відповідність запропонованим вимогам. В разі, якщо онтологічна модель не відповідає критеріям оцінки її якості, здійснюється її структурна перевірка та оптимізація змістової частини послідовною редукцією її графу до виконання вимог вибраних критеріїв, причому розв'язування цих задач розноситься у часі, щоб зберегти цілісність онтології.

По друге, визначені оцінки ефективності самого процесу моделювання сценарію, якій призначений багатofакторного аналізу. Побудований алгоритм, який надає можливість однозначного вибору ефективного сценарію за обчисленим значенням заданої метрики затрат на моделювання на основі оцінки скорочення витрат пошуку по графу, що описує онтологію, у порівнянні з неспрямованим пошуком шляху на графі.

2. На основі розроблених ознак і показників стійкості аналітичного процесу введені поняття межі та запасу стійкості аналітичного процесу. Запропоновано кількісні методи оцінки стійкості аналітичного процесу за чисельними показниками. В якості переваг даного методу можна вказати можливість кількісно оцінити стійкість аналітичного процесу поточної структури послідовності дій користувача для досягнення мети дослідження на основі простих зовнішніх ознак. Використовуючи дані оцінки можна давати рекомендації щодо вимог організації аналітичного процесу, що відображає послідовності дій користувача для досягнення мети дослідження.

2. Були проведені імітаційні експерименти моделювання сценаріїв за використанням запропонованих методів та розроблених програмних засобів на двохстах тестових прикладах вирішення задач аналітики з усередненням результатів тестових прикладах для різних ПрО (онтологічних моделей). Оцінка затрат визначена як оцінка скорочення часу (в умовних одиницях) виконання аналітичних процедур. Результати тестування показують, що використання запропонованих методів та програмних засобів, які створені на їх базі, дає можливість значно зменшити складність та затрати на побудову нових більш ефективних за визначеними критеріями сценаріїв, зокрема значно скоротити час здійснення АнД – у середньому більш чим у п'ять разів.

3. Показана на прикладі порівняльного аналізу виразних можливостей мов, які використовують для формування запитів до побудованої онтології сценарію, важливість їх ергономічності для забезпечення ефективності здійснення аналітичної діяльності користувачем (аналітиком). Будь-якій

користувач повинен мати можливість інтуїтивно пов'язувати дії, які потрібно виконати, з об'єктами, що він бачить, зазначені мови використовуються як інструмент, за допомогою якого користувачі можуть взаємодіяти з онтологією та даними.

ВИСНОВКИ

У дисертаційній роботі вирішено важливу науково-прикладну проблему – побудови інтелектуальних аналітичних систем підтримки прийняття рішень, за рахунок порівняння та оцінки найбільш ймовірних сценаріїв розвитку процесів та подій. Запропоновано методологію формування сценаріїв АнД та підвищення ефективності процесу формування сценаріїв завдяки застосуванню розробленого математичного та програмного забезпечення, що ґрунтується на використанні онтології, адаптації онтології до специфіки задач ПрО та інтелектуалізації формування сценаріїв. Під час виконання роботи одержано такі наукові результати.

Вперше запропоновано:

- теоретичний підхід до комп’ютерного моделювання багаторівневих сценаріїв АнД, якій відрізняється від відомих тим, що оснований на поєднанні переваг візуального моделювання – графічної нотації BPMN (Business Process Model and Notation) з керованою моделлю і семантичного аналізу на базі онтологічної моделі предметної області задач АнД, яка конвертується з BPMN-модель, та подальшого автоматичного перетворення метаданих BPMN-моделі в виконавче програмне середовище (серіалізації), що зменшує кількість логічних розривів в комп’ютерній моделі побудови сценарію АнД, яка будується, та суттєво зменшує час моделювання в цілому;

- метод перетворення BPMN-моделі в модель уявлення про предметну область сценарію АнД мовою OWL (Web Ontology Language) - OWL-модель, якій відрізняється від відомих тим, що за рахунок забезпечення співвідношення категорії основних графічних примітивів нотації BPMN до відповідних класів онтології (сутностей) та зв’язування їх з властивостями та відношеннями між елементами реальної моделі вирішення задачі АнД в нотації BPMN дозволяє проводити семантичний аналіз коректності логічно-зв’язаних фрагментів сценарію вирішення задачі АнД та його верифікацію (особливо багаторівневих сценаріїв);

– інформаційна технологія конвертації BPMN-моделі в OWL-модель сценарію АнД, яка відрізняється від відомих тим, що дозволяє перетворювати метадані опису сценарію та його предметної області в базі знань, накопичуючи нові знання, що дає можливість побудови нових сценаріїв АнД на базі вже існуючих в базі знань. Для спрощення процесу комп'ютерного моделювання сценаріїв залучається графічна нотація BPMN 2, яка дозволяє не тільки моделювати сценарії АнД у вигляді BPMN-діаграм, але й серіалізувати його опис в XML-файл для реалізації сценарію в комп'ютерному середовищі із залученням стандартного програмного забезпечення BPM-систем.

Удосконалено:

– математична модель обчислення розподілу значень можливих подій сценарію та інтелектуальний метод визначення наступних подій сценарію АнД, який відрізняється від відомих тим, що на основі застосування методів машинного навчання, а саме розвитку методу навчання деревами класифікації і регресії (Classification and Regression Trees) з використанням комбінації метрик оцінок ефективності та накопичених знань про семантику предметної області для визначення кореневого вузлу графу, дозволяє виконувати формування дерева рішень для класифікації можливих наступних кроків сценарію. Це дає можливість значно підвищити точність розбиття графу складного сценарію на відповідні гілки при обчисленні ймовірності наступного кроку сценарію;

– метод реалізації сценаріїв АнД на основі Web-сервісів, який відрізняється від відомих тим, що вибір сервісів засновано на механізмі порівняння параметрів їх метаописів із параметрами функціональних завдань сценарію, які задаються його метаописом, що забезпечує можливість виконання завдань кінцевого користувача шляхом динамічного формування послідовності Web-сервісів.

Набули подальшого розвитку:

– процес автоматизованої побудови сценаріїв АнД на основі бібліотеки сценаріїв (прецедентів) для різних предметних областей. Це дало можливість в

автоматизованому режимі будувати сценарії на основі експертного опису проблемних ситуацій, здійснювати деталізацію сценаріїв та розгалуження їх. У свою чергу, це суттєво зменшує часові та фінансові витрати на моделювання та реалізацію сценаріїв АНД.

– інтелектуальний програмний засіб, що навчається на діях користувача та на основі отриманих в процесі навчання знань представляє дії користувача у вигляді дерева рішень ймовірного сценарію аналітичного процесу, пропонуючи користувачу найбільш доцільні наступні кроки вирішення завдання АНД;

– трирівнева комп'ютерна моделююча система (інтегроване програмне інструментальне середовище) для тестування і оцінки запропонованого теоретичного підходу (методології) моделювання сценаріїв АНД, яка ґрунтується на запропонованих моделях, методах та алгоритмах моделювання, що дало можливість реалізувати окремі програмні інтелектуальні засоби вирішення задач аналітики для різних предметних областей аналітичної діяльності, ядром баз знань яких є знання та накопичений досвід аналітика (експерта, проектувальника). Програмні інтелектуальні засоби були використані зокрема для вирішення задачі побудови та реалізації сценаріїв моделювання процесів функціонування вимірювальної гідроакустичної системи з метою удосконалення алгоритмів обробки гідроакустичних сигналів, та сценаріїв розподілу енергопостачання з метою оптимізації енерговитрат.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Amazon Web Services. URL: https://en.wikipedia.org/wiki/Amazon_Web_Services
2. Thacker B.H. et al. Concepts of model verification and validation. – Los Alamos National Lab., Los Alamos, NM (US), 2004. No. LA-14167.
3. France R., Rumpe B. Model-driven development of complex software: A research roadmap // 2007 Future of Software Engineering. – IEEE Computer Society. 2007. С. 37–54.
4. J.L.G. Dietz. Architecture – Building strategy into design. Academic Service, The Hague, The Netherlands, 2008.
5. Thomas Erl. Principles of Service Design. The Prentice Hall Service-Oriented Computing Series from Thomas Erl. Prentice Hall, Upper Saddle River, NJ, USA, 2007.
6. Stein S. et al. Evaluation of OrViA framework for model-driven SOA implementations: An industrial case study // Business Process Management. – Springer Berlin Heidelberg, 2008. С. 310–325. URL: <http://www.theenterpriseearchitect.eu/blog/2009/06/03/a-framework-for-model-driven-soa/>
7. Aamodt A. Case-based reasoning: Foundational issues, methodological variations, and system approaches / A. Aamodt, E. Plaza // AI Communications, Vol. 7, Issue 1, 1994. P. 39–59.
8. Alex Guazzelli, Michael Zeller, Wen-Ching Lin and Graham Williams PMML: An Open Standard for Sharing Models: available at: https://journal.r-project.org/archive/2009-1/RJournal_2009-1_Guazzelli+et+al.pdf
9. Juha-Pekka Tolvanen and Steven Kelly, Model-Driven Development Challenges and Solutions: Experiences with Domain-Specific Modelling in Industry: available at: https://www.metacase.com/papers/Tolvanen_Kelly_MODELSWARD_2016.pdf
10. Alexander Koval, Larisa Globa, Rina Novogrudska The approach to web services composition, Hard and Soft Computing for Artificial Intelligence, Multimedia

and Security, Volume 534 of the series *Advances in Intelligent Systems and Computing*, Springer international publication AG 2017. DOI 10.1007/978-3-319-48429-7. P. 293–304.

11. Alexis Gabadinho, Gilbert Ritschard, Matthias Studer: Mining sequence data in R with the TraMineR package: available at: <http://mephisto.unige.ch/pub/TraMineR/doc/TraMineR-Users-Guide.pdf>

12. AllegroGraph: available at <http://franz.com/agraph/allegrograph/>

13. An integrated approach to automated semantic web service composition through planning. *Services Computing, IEEE Transactions on*, 5(3), 319–332.

14. An introduction to the owl web ontology language, available at: <http://www.cse.lehigh.edu/~heflin/IntroToOWL.pdf>

15. An Overview of Data Mining Techniques. available at: <http://www.thearling.com/text/dmtechniques/dmtechniques.htm>

16. Analyzing ontology editing tools for effective semantic information retrieval INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY L, Namrata Rastogi, Dr. Parul Verma, Dr. Pankaj Kumar. URL: <http://www.ijesrt.com/issues%20pdf%20file/Archive-2017/May-2017/6.pdf>

17. Analytic Solver Platform. XLMiner Platform. Data Mining User Guide. Version 2016-R3. Frontline Systems, Inc., 2016. 187 p.

18. Announcing the 2015 Winner for Best Free CMS. URL: <https://www.cmscritic.com/announcing-the-2015-winner-for-best-free-cms/>

19. ARIS Cloud is a full-scale Business Process Analysis-as-a-Service product. available at: <http://www.ariscommunity.com/ariscloud>

20. ASP.NET MVC: available at http://www.w3schools.com/aspnet/mvc_intro.asp

21. Astah: Java UML Modeller: available at: <http://arakhne.org/astah/index.html>

22. Basu 3 A., Blanning R. *Metagraphs and their applications*. Springer Science+Business Media, LLC, 2008, 173 p.

23. Big data: available at: https://www.webopedia.com/TERM/B/big_data.html
24. Camel case notation: available at: https://en.wikipedia.org/wiki/Camel_case
25. Carol E. Brown ra Daniel E. O’Leary, Introduction to artificial intelligence and expert systems: available at: https://msbfile03.usc.edu/digitalmeasures/doleary/intellcont/Brown-Oleary-es_tutor-1.htm
26. Case Based Reasoning: available at: <http://www.aiai.ed.ac.uk/links/cbr.html>
27. Chris Snijders, Uwe Matzat, Ulf-Dietrich Reips «Big Data»: Big Gaps of Knowledge in the Field of Internet Science *International Journal of Internet Science* 2012, 7 (1): available at http://www.ijis.net/ijis7_1/ijis7_1_editorial.pdf
28. Cloud AutoML: available at: <https://cloud.google.com/automl>
29. Cluster analysis: available at: https://en.wikipedia.org/wiki/Cluster_analysis
30. Codd E.F.; Codd S.B. & Salley C.T. (1993). «Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate» (PDF). Codd & Date, Inc. Retrieved 2008-03-05.
31. Computer-Aided System Engineering. URL: <https://www.techopedia.com/definition/3973/computer-aided-software-engineering-case>
32. Data Catalog Vocabulary (DCAT): available at: <http://www.w3.org/TR/2014/REC-vocab-dcat-20140116/>
33. Data Flow Diagrams. URL: <http://www.cems.uwe.ac.uk/~kg-doyle/tdrewry/dfds.htm>
34. Data – Information – Knowledge: available at: http://www.cyberartsweb.org/cpace/ht/thonglipfei/data_info.html.
35. Data Mining Concepts, Oracle Database Online Documentation 11g : available at: https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/algo_svm.htm#DMCON025
36. David J. DeWitt, Ph.D.; Samuel Madden, Ph.D.; Michael Stonebraker, Ph.D How to Build a High-Performance Data Warehouse: available at http://db.csail.mit.edu/madden/high_perf.pdf
37. Data mapping. URL: https://en.wikipedia.org/wiki/Data_mapping

38. Description Logics as Ontology Languages for the Semantic Web Franz Baader, Ian Horrocks, Ulrike Sattler. URL: <http://www.cs.ox.ac.uk/ian.horrocks/Publications/download/2003/BaHS03.pdf>
39. Disruptive technologies: Advances that will transform life, business, and the global economy: available at: http://www.mckinsey.com/insights/business_technology/
40. Dustdar S., and Schreiner W. A survey on web services composition. In Int. J. Web and Grid Services. 2005. Vol. 1, No. 1. P.1–30.
41. Edraw Max Pro: available at: <https://www.edrawsoft.com/EDrawMax.php>
42. Eriksson-Penker Process Diagram. URL: <http://astah.net/features/useful-functions>
43. Eugene, K and Carman, W.C. (2006). Analysis of means-end chain data in marketing research. Cambridge, Mass: MIT Press.
44. Extensible Markup Language (XML). available at: <https://www.w3.org/XML/>
45. Коваль А.В., Зайцева Е.А., Бойко Ю.Д. Formation of Analytical Activity Scenarios. *Системні дослідження та інформаційні технології*. 2014. № 1. С. 20–25.
46. Frame language. URL: https://en.wikipedia.org/wiki/Frame_language
47. Front and back ends. URL: http://en.wikipedia.org/wiki/Front_and_back_ends
48. General Descriptions of the CmapTools Toolkit: available at: <http://cmap.ihmc.us/docs/documentation-cmaps>
49. Graham Williams: Data Mining with Rattle and R. The Art of Excavating Data for Knowledge Discovery: ISBN 978-1-4419-9889-7 e-ISBN 978-1-4419-9890-3 DOI 10.1007/978-1-4419-9890-3. Springer. New York Dordrecht Heidelberg London: available at: <https://mineriaddatos.wikispaces.com/file/view/>
50. Graph Analysis and Visualization: Discovering Business Opportunity in Linked Data Published by John Wiley & Sons, Inc. 10475 Crosspoint Boulevard Indianapolis, IN 46256

51. Graph Data Model. URL: <http://infolab.stanford.edu/~ullman/focs/ch09.pdf>
52. Graph database: available at: https://en.wikipedia.org/wiki/Graph_database
53. Hadoop: available at: https://uk.wikipedia.org/wiki/Apache_Hadoop
54. HermiT: An OWL 2 Reasoner, Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, Zhe Wang: Journal of Automated Reasoning manuscript. URL: <https://www.cs.ox.ac.uk/boris.motik/pubs/ghmsw14HermiT.pdf>
55. How does a reasoner work. URL: <http://ontogenesis.knowledgeblog.org/1486>
56. IBM Watson Studio. URL: <https://www.ibm.com/cloud/watson-studio>
57. IDEF1X. URL: <http://www.essentialstrategies.com/publications/modeling/idef1x.htm>
58. IDEF3. URL: <http://www.conceptdraw.com/examples/idef3>
59. IDEF5. URL: <http://www.idef5.com/Download/Idef5.pdf>
60. Introduction to Multidimensional Databases. URL: <https://www.topcoder.com/community/data-science/data-science-tutorials/anintroduction-to-multidimensional-databases/>
61. Introduction to Oracle Real Application Clusters: available at: http://docs.oracle.com/cd/B28359_01/rac.111/b28254/admcon.htm#i1058057
62. ISO/IEC 9126-1:2001(E). Information technology – Software engineering – Product quality –Part 1: Quality model 25 p.
63. ISO/IEC TR 9126-2: Software engineering – Product quality – Part 2: External metrics, 106 p.
64. ISO/IEC 15288:2002 «System engineering –System life cycle processes»
65. Explore IBM Cloud products. URL: <https://www.ibm.com/cloud>
66. IaaS, PaaS and SaaS – IBM Cloud service models. URL: <https://www.ibm.com/cloud/learn/iaas-paas-saas>
67. ISO/IEC 7498-1:1994. *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.*
68. jCOLIBRI. URL: <http://gaia.fdi.ucm.es/research/colibri>

69. jCOLIBRI 1.0 in a nutshell. A software tool for designing CBR systems.
70. John Mylopoulos (2004). Conceptual Modelling III. Structured Analysis and Design Technique (SADT). Retrieved 21 September 2008.
71. JUDE/Professional Overview. URL: <http://jude.change-vision.com/jude-web/product/features.html>
72. Joomla – Content management system. URL: <https://www.joomla.org/about-joomla.html>
73. KAON Ontology Framework <https://sourceforge.net/projects/kaon/>
74. KNIME Analytics Platform: available at: <http://tech.knime.org/knime-text-processing>
75. Know-how: available at: <https://en.wikipedia.org/wiki/Know-how>
76. Knowledge Interchange Format: available at: <http://logic.stanford.edu/kif/dpans.html>
77. Knowledge representation languages: available at: <http://www.cs.man.ac.uk/~stevensr/onto/node14.html>
78. Koval A., Senchenko V., Gagarin A., Gaydargu V. Gathering and processing of information on marine moving facilities IEEE BlackSeaCom 2017 – International Black Sea Conference on Communications and Networking 5-8 JUNE 2017 // ISTANBUL, TURKEY
79. Koval O.V., Kuzminykh V.A., Khaustov D.V. Using stochastic automaton for data consolidation. *Наукoви вiснoвкoвi «KIII»*. 2017. № 2. С.29–36.
80. Kontchakov Roman, Mariano Rodr'iguez-Muro, Zakharyashev Michael Ontology-Based Data Access with Databases: A Short Course. URL: <http://www.dcs.bbk.ac.uk/~roman/papers/RW-Chapter.pdf>
81. Leemans Maikel, Wil M.P. van der Aalst, Process mining in software systems: Discovering real-life business transactions and process models from distributed systems. In *International Conference on Model Driven Engineering Languages and Systems 2015*, Ottawa, Canada. URL: <http://www.processmining.org/>

_media/blogs/pub2015/models2015_leemans_process_mining_in_software_systems.pdf

82. Leslie F. Sikos: Description Logics in Multimedia Reasoning. URL: <https://books.google.com.ua/books?>

83. List of countries by GDP (nominal). URL: [https://en.wikipedia.org/wiki/List_of_countries_by_GDP_\(nominal\)](https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal))

84. Magic Quadrant for Data Science Platforms: available at: <https://www.gartner.com/doc/reprints?id=1-3TINH7V&ct=170214&st=sb>

85. MapReduce. URL: <https://en.wikipedia.org/wiki/MapReduce>

86. Markus Hofmann, Ralf Klinkenberg, RapidMiner: Data Mining Use Cases and Business Analytics Applications, Butler Analytics, 2013. URL: <http://rapidminerbook.com/yN6sl08>

87. Martin D., Paolucci M., McIlraith S., Burstein M., McDermott, D., McGuinness D., & Srinivasan, N. (2004). Bringing semantics to web services: The OWL-S approach. In *Semantic Web Services and Web Process Composition* (P. 26–42). Springer Berlin Heidelberg.

88. Marvin Minsky A Framework for Representing Knowledge, available at: <http://web.media.mit.edu/~minsky/papers/Frames/frames.html>

89. Max Planck. URL: <http://cityat.socratify.net/maks-plank>

90. McElroy, M. W. (2003a). Understanding ‘The New Knowledge Management: available at: www.macroinnovation.com

91. Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, KDD’96. P. 226–231

92. McKinsey Global Institute: James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers – Rejrt – Big data: The next frontier for innovation, competition, and productivity, may 2011, 156 p.

93. Medjahed, B., & Bouguettaya, A. (2011). *Service composition for the Semantic Web*. Springer Science & Business Media.

94. Mika, P., Akkermans H. Towards a New Synthesis of Ontology Technology and Knowledge Management Knowledge. Engineering Review, Cambridge University Press. 2004. Vol. 19, No. 4.

95. Moghaddam, M., & Davis, J.G. (2014). Service selection in web service composition: A comparative review of existing approaches. In: Web Services Foundations (P. 321–346). Springer New York

96. Murray E. Jennex, Knowledge Management Systems: available at: http://www.setlab.net/?view=KMS_Jennex

97. Novograduska R., Globa L., Koval O. The Method of User's Tasks Scenario Formation. The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 21–23 September, 2017, Bucharest, Romania/

98. Nyce, Charles (2007), *Predictive Analytics White Paper* (PDF), American Institute for Chartered Property Casualty Underwriters/Insurance Institute of America. P. 1.

99. Novograduska R.L., Globa L.S., Koval O.V., Senchenko V.R., Ontology model of intelligent modeling system for marine facilities identification, Proceedings of International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), 2017, DOI:10.1109/UkrMiCo.2017.8095426, IEEE Digital Library, 8095426 (Scopus).

100. Novograduska R.L. Ontology for Applications Development / R.L. Novograduska, L. S. Globa, O. V. Koval, V. R. Senchenko // chapter 2 Ontology in Information Science / book edited by Ciza Thomas, Print ISBN 978-953-51-3887-7, Published: March 8, 2018. P. 29–53. <http://dx.doi.org/10.5772/intechopen.74042>

101. Novograduska R.L., Globa L. S., Koval O. V., Senchenko V. R. Examples of Ontology Model Usage in Engineering Fields / R. L. Novograduska, L. S. Globa, O. V. Koval, V. R. Senchenko // chapter 3 Ontology in Information Science / book edited by Ciza Thomas, Print ISBN 978-953-51-3887-7, Published: March 8, 2018. P. 254–81. <http://dx.doi.org/10.5772/intechopen.74369>

102. OASIS SOA reference model. URL: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
103. Object Management Group / Modeling Specifications. URL: <http://www.omg.org/spec/#M&M>
104. Object-data-model. available at: <http://www.gartner.com/it-glossary/object-data-model/>
105. Object-Oriented Data Model available at: <http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter8/node3.html>
106. Odase Platform: available at: <http://www.odaseontologies.com/odase-platform/>
107. OLAP Council White Paper (PDF). OLAP Council. 1997. Retrieved 2008-03-18. URL: http://www.symcorp.com/downloads/OLAP_CouncilWhitePaper.pdf
108. On Data, Information, and Knowledge. URL: <http://beaststwo.org/inforant/index.shtml>
109. Ontop: Answering SPARQL queries over relational databases, Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebria, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, Guohui Xiao: available at: <http://eprints.bbk.ac.uk/15625/1/main.pdf>
110. OntoStudio: available at: <http://www.semafora-systems.com/en/products/ontostudio/>
111. Open Knowledge Base Connectivity: available at: <http://www.ai.sri.com/~okbc/>
112. Oracle Advanced Analytics Data Mining Algorithms and Functions SQL API: available at: <http://www.oracle.com/technetwork/database/options/advanced-analytics/odm/odm-techniques-algorithms-097163.html>
113. Oracle Data Mining Concepts, 11Release 1 (11.1): available at: https://docs.oracle.com/cd/B28359_01/datamine.111/b28129.pdf
114. Orange – офіційний сайт: <https://orange.biolab.si/download/#windows>

115. OWL 2 Web Ontology Language Document Overview (Second Edition) W3C Recommendation 11 December 2012: available at: <http://www.w3.org/TR/owl2-overview/>

116. OWLGrEd Ontology Visualizer, Institute of Mathematics and Computer Science, University of Latvia, Raina bulvaris 29, Riga, LV-1459, Latvia: available at <http://ceur-ws.org/Vol-1268/paper7.pdf>

117. PMML 4.3 – General Structure: available at: <http://dmg.org/pmml/v4-3/GeneralStructure.html>

118. Practical Reasoning with OWL and DL-Safe Rules, P. Haase, P. Hitzler, M. Krötzsch, J. Angele: available at: https://km.aifb.kit.edu/ws/prowl2006/prowl06_4on1.pdf

119. Protégé – Free, open-source ontology editor and framework for building intelligent systems: available at: <http://protege.stanford.edu>

120. RapidMiner Data Mining Use Cases and Business Analytics Applications: Markus Hofmann Institute of Technology Blanchardstown, Dublin, Ireland, Ralf Klinkenberg Rapid-I / RapidMiner Dortmund, Germany. International Standard Book Number-13: 978-1-4822-0550-3 (eBook – PDF).

121. RapidMiner Studio: Visual Workflow Designer for Data Scientists: available at: <https://rapidminer.com/products/studio/>

122. Rational Unified Process. URL: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

123. RDF Model Theory: available at: <https://www.w3.org/TR/2002/WD-rdf-mt-20020429/>

124. Relational model: available at: https://en.wikipedia.org/wiki/Relational_model

125. Resource Description Framework (RDF): available at: <http://www.w3.org/RDF/>

126. Resource Description Framework Schema, RDF Schema 1.1: available at: http://www.w3.org/TR/rdf-schema/#ch_introduction

127. Reviews for Data Science Platforms: available at: <https://www.gartner.com/reviews/market/advanced-analytics-platforms>
128. Richard J. Mayer et al. (1995). IDEF4 Object-Oriented Design Method Repor.
129. Robert Kosara (2007). Visual Analytics. ITCS 4122/5122, Fall 2007.
130. Salton, G.; Allan, J.; Buckley, C.; Singhal, A. (1994). Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts. *Science* 264 (5164): 1421–1426.
131. TopQuadrant Technology Briefing Semantic Technology: available at : [//lists.oasis-open.org/archives/regrep-semantic/200402/pdf00000.pdf](https://lists.oasis-open.org/archives/regrep-semantic/200402/pdf00000.pdf)
132. Semantic network: available at: https://en.wikipedia.org/wiki/Balance_theory
133. Semantic Web Technologies (2013): available at: <https://open.hpi.de/course/semanticweb>
134. Senchenko V., Koval A., Gerayimchuk M., Khazanovych Y. Ontological design of an intelligent system for studying hydroacoustical processes IEEE ICPCSI-2017 – International Conference on Power, Control, Signals and Instrumentation Engineering 21–22 September 2017 INDIA.
135. Senchenko V.R. Intelligent modeling system based on Cloud-technology / Senchenko V.R., Koval A.V., Globa L.S., Novogrudska R.L. 2016 IEEE Міжнародна конференція з інформаційно-телекомунікаційних технологій та радіоелектроніки (УкрМКО'2016/UkrMiCo'2016). 11–15 вересня 2016 р. м. Київ, Україна.
136. Senchenko V.R., Koval A.V., The technology of semantic modeling for knowledge management system in environment Protégé Информационные технологии и безопасность, материалы XVII международной научно-практической конференции, выпуск 17, Киев, 2017. С. 211–234.
137. Solve more with Google Cloud. URL: <https://cloud.google.com/>

138. Shehu U., Epiphaniou G., & Safdar G. A. (2014). A survey of QoS-aware web service composition techniques. *International Journal of Computer Applications*.
139. Six of the Best Open Source Data Mining Tools, available at: <https://thenewstack.io/six-of-the-best-open-source-data-mining-tools/>
140. Sniedovich, M. (2006). Dijkstra's algorithm revisited: the dynamic programming connexion. *Journal of Control and Cybernetics* 35: available at: <http://matwbn.icm.edu.pl/ksiazki/cc/cc35/cc3536.pdf>
141. SPARQL Query Language for RDF, Update, 26 March 2013: available at: <https://www.w3.org/TR/rdf-sparql-query/#introduction>
142. Stahl A. Rapid Prototyping of CBR Applications with the Open Source Tool myCBR / A. Stahl, T. Roth – Berghofer // Proc. of the 9th European Conference on CBR (ECCBR 2008). Trier (Germany): Springer, 2008. P. 615–629.
143. SharePoint: available at: <https://en.wikipedia.org/wiki/SharePoint>
144. Statistical Data and Metadata eXchange: available at: https://sdmx.org/?page_id=5008
145. Stumme, G. Semantic Web Mining. State of the art and future directions / G. Stumme, A. Hotho, B. Berendt // *Web Semantics: Science, Services and Agents on the World Wide Web*. 2006. No. 4. P. 124–143.
146. SWOOP: available at: <http://www.softpedia.com/get/Internet/Other-Internet-Related/MIND-lab-SWOOP.shtml>
147. SWRL: A Semantic Web Rule Language Combining OWL and RuleML: available at: <https://www.w3.org/Submission/SWRL/>
148. Team Data Science Process Documentation. URL: <https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/>
149. The Dublin Core metadata element set ISO 15836:2009: available at: <https://www.iso.org/standard/52142.html>
150. TensorFlow Model Analysis <https://github.com/tensorflow/model-analysis/>

151. The Manchester OWL Syntax: available at: http://ceur-ws.org/Vol-216/submission_9.pdf
152. The Neo4j Graph Database [open-source]: available at: <http://neo4j.com/docs/stable/graphdb-neo4j.html>
153. The Open Graph Viz Platform: available at: <https://gephi.org/>
154. TODE: A Dot Net Based Tool for Ontology Development and Editing. In proceedings of the 2nd International Conference on Computer Engineering and Technology, Chengdu, China, 14–16 April 2010.
155. Tim O'Reilly. What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. URL: <http://oreilly.com/web2/archive/what-is-web-20.html>.
156. Tutorial: How to Use Ontobee SPARQL to Query RDF triple store: available at: <http://www.ontobee.org/tutorial/sparql>
157. Ulrich's Periodicals Directory: available at <http://www.proquest.com/products-services/Ulrichsweb.html>
158. Unified Modeling Language™ (UML®) Version 2.5: available at: <http://www.omg.org/spec/UML/2.5/PDF/>
159. Web Services Choreography Description Language (WS-CDL): available at: <http://www.goland.org/wscdl/>
160. WordPress Content management system: available at: <https://wordpress.org/about/>
161. Vipin Kumar, COMPUTATIONAL BUSINESS ANALYTICS - Data Mining and Knowledge Discovery Series: University of Minnesota Department of Computer Science and Engineering Minneapolis, Minnesota, U.S.A. P. 505.
162. Web Ontology Language: available at: <http://www.w3.org/TR/owl-features/>
163. Web Services Business Process Execution Language: available at: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
164. Web Services Choreography Description Language: available at: <https://www.w3.org/TR/ws-cdl-10/>

165. WebProtégé User Guide: available at: –
<http://protegewiki.stanford.edu/wiki/WebProtegeUsersGuide>
166. Weka 3 – Data Mining with Open Source Machine Learning. URL:
<https://www.cs.waikato.ac.nz/ml/weka/>
167. William S. Davis (1992). Tools and Techniques for Structured Systems Analysis and Design. Addison-Wesley. ISBN 0-201-10274-9.
168. World Wide Web Consortium (W3C). URL: <http://www.w3.org>
169. Wollenberg, E. Anticipationg change: scenarios as a tool for adaptive forest management [Text]: a guide / E. Wollenberg, D. Edmunds, L. Buck. – Bogor, Indonesia: Center for International Forestry Research, 2000. 36 p.
170. Xlminer data mining add-in for excel:available at:
<https://www.solver.com/xlminer-data-mining>
171. Multi-agent system. URL: https://en.wikipedia.org/wiki/Multi-agent_system
172. Алекс Дж. Шампандар: Искусственный интеллект в компьютерных играх: как обучить виртуальные персонажи реагировать на внешние воздействия: пер. с англ. Москва: Издательский дом «Вильямс», 2007. 768 с.
173. Андон Ф.И., Гришанова И.Ю., Резниченко В.А. Semantic Web как новая модель информационного пространства Интернет. URL:
<http://shcherbak.net/semantic-web-kak-novaya-model-informacionnogo-prostranstva-internet/>
174. Басергян А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. 2-е изд. Санкт-Петербург: БХВ-Петербург, 2007. 375 с.
175. Барсегян А.А., Куприянов М.С, Холод И.И., Тесс М.Д., Елизаров С.И. Анализ данных и процессов: учеб. пособ. 3-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2009. 512 с.
176. Божко С.Г., Шевченко І.А., Печериця О.В., Сингаївський А.М. Управління знаннями, робочими процесами та електронним документообігом у системі Держатомрегулювання. Основні елементи передпроектної підготовки

створення Порталу знань. URL: http://sstc.kiev.ua/documents/journal/2016/2/1_2_2016_uk.pdf

177. Букович У., Уильямс Р. Управление знаниями: руководство к действию; пер. с англ. Москва: ИНФРА-М, 2002.

178. Бурков В.Н., Коргин Н.А., Новиков Д.А. Введение в теорию управления организационными системами / под ред. чл.-корр. РАН Д.А. Новикова. Москва: Либроком, 2009. 264 с.

179. Бідюк П.І., Загірська І.О. Методика побудови сценарного аналізу із використанням байєсівських методів. *Електротехнічні та комп'ютерні системи*. 2012. № 08(84). С. 137–142.

180. Кононов Д.А., Кульба В.В. Формирование сценариев развития макроэкономических процессов на базе использования языка знаковых графов.

181. Василевич Л.Ф., Михайлюк А.Ю., Тарасенко В.П., Тесленко О.К. Функціонально-орієнтований підхід до проектування інтелектуальних інформаційно-аналітичних систем. *Ресстрація, зберігання і обробка даних*. 2010. Т. 12, № 2. С. 128–142.

182. Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений. *Искусственный интеллект и принятие решений*. 1/2009.

183. Воелтер М. Разработка DSL: проектирование, внедрение и использование предметно-ориентированных языков: пер. с англ. Маркус Воелтер. Москва: ООО «И.Д. Вильямс», 2013. 560 с.

184. Гладун А.Я., Рогушина Ю.В. Онтологии в корпоративных системах: <http://www.management.com.ua/ims/ims115.html>

185. Глоба Л.С., Коваль О.В., Новогрудська Р.Л., Сенченко В.Р. Створення сценаріїв обробки даних на основі онтологій / Системний аналіз та інформаційні технології: матеріали 18-1 Міжнародної науково-технічної конференції SAIT 2016. (30 травня – 2 червня 2016 р. м. Київ). Київ: ННК «ІПСА» НТУУ «КПІ», 2016. С. 262–265.

186. ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. 106 с.

187. Грейді Буч, Джеймс Рамбо, Айвар Джекобсон. Мова *UML*. Керівництво користувача = *The Unified Modeling Language user guide*. 2-е вид. – Москва, Санкт-Петербург: ДМК Пресс, Пітер, 2004. 432 с.

188. Грищенко М.А., Николайчук О.А., Павлов А.И., Юрин А.Ю. Инструментальное средство создания продукционных экспертных систем на основе MDA. *Образовательные ресурсы и технологии*. 2016'2 (14). С. 144–151.

189. Девятков В.В. Системы искусственного интеллекта. Москва: Изд. МГТУ им. Баумана, 2001.

190. Дейт, К. Дж. Введение в системы баз данных, 8-е издание: пер. с англ. Москва: Издательский дом «Вильямс», 2005. 1328 с.: ил. Парал. тит. Англ.

191. Джанетто К., Уилер Э. Управление знаниями. Руководство по разработке и внедрению корпоративной стратегии управления знаниями. Москва: Хорошая книга, 2005. 192 с.

192. Джарратано Д., Райли Г. Экспертные системы: принципы разработки и программирование, 4-е издание: пер. с англ. Москва: Издательский дом «Вильямс», 2007. 1152 с.

193. Джексон П. Введение в экспертные системы. Москва: Издательский дом «Вильямс», 2001. ISBN 5-8459-0150-2 ISBN 0-201-87686-8.

194. Дзяна С.Р., Дзяний Р.Б. Теоретичні засади управління змінами в сучасних умовах. *Ефективність державного управління*. 2013. Вип. 34. С. 31–40. URL: http://nbuv.gov.ua/j-pdf/efdu_2013_34_5.pdf

195. Додонов А.Г., Бойченко А.В. Разработка сценариев аналитической деятельности. *Регістрація, зберігання і обробка даних*, 2010, Т. 12, № 4. С. 71–82.

196. Додонов А.Г., Сенченко В.Р., Коваль А.В. Аналитика и знания в компьютерных системах. Киев: ИПРИ НАН Украины, «КПИ имени Игоря Сикорского», 2020. 315 с.

197. Додонов О.Г., Коваль О.В, Глоба Л.С, Бойко Ю.Д. Комп'ютерне моделювання інформаційно-аналітичних систем. Монографія. Київ: ІПРІ НАН України, 2017. 239 с.

198. ДСТУ 2481-94 «ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»

199. Заде Л. Понятие лингвистической переменной и ее применение к принятию приближенных решений. Москва: Мир, 1976. 166 с.

200. Згуровский М.З., Панкратова Н.Д. Системный анализ: проблемы, методология, приложения. Київ: Наук. думка. 2011. 727 с. ISBN 978-966-99-1124-3.

201. Згуровский М.З., Павлов А.А. Принятие решений в сетевых системах с ограниченными ресурсами. Монография. Киев: Наук. думка. 2010. 573 с.

202. ИСО/МЭК 15288:2002 «Системная инженерия. Процессы жизненного цикла систем» 57 с.

203. Коваль А.В., Бойко Ю.Д., Зайцева Е.А. Модель сценарно-целевого подхода при построении информационно-аналитической системы // Сборник тезисов 16-й международной научно-технической конференции «Системный анализ и информационные технологии» SAIT 2014 (26–30 мая 2014 г. Киев). Киев: ННК «ІПСА» НТУУ «КПІ», 2014. С. 105–106.

204. Коваль А.В., Зайцева Е.А., Бойко Ю.Д. Верифікація комп'ютерної моделі системи інформаційного управління. Збірник «Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка». 2014. № 2. С.26.

205. Круглов В.В., Дли М.И., Голунов Р.Ю. Нечеткая логика и искусственные нейронные сети. Москва: Физматлит, 2000. 224 с.

206. Курносков Ю.В., Конотопов П.Ю. АНАЛИТИКА: методология, технология и организация информационно-аналитической работы. Москва: Издательство «Русакі», 2004 г. 550 с

207. Курносков Ю.В. Аналитика как интеллектуальное оружие. Москва: РУСАКИ, 2012. ISBN 978-5-93347-433-3. URL: http://www.litres.ru/pages/biblio_book/?art=24156898

208. Ланде Д.В., Додонов В.А. Модель розповсюдження інформації з урахуванням поняття сприйняття і пам'яті. Авіаційна та екстремальна психологія у контексті технологічних досягнень: зб. наук. праць. Київ: Аграр Медіа Груп, 2017. С. 148–153.

209. Ларман К.: Применение UML 2.0 и шаблонов проектирования. Москва; Санкт-Петербург; Киев: ИД «Вильямс», 2007. 728 с.

210. Литвин В.В., Гопяк М.Я, Демчук А.Б. Метод автоматизованої розбудови та оцінювання якості онтологій баз знань. УДК 004.89. С. 61–68.

211. Люгер Дж.Ф. Искусственный интеллект. Москва: Вильямс, 2005. 864 с.

212. Луканин А.В.: Нейронные сети как механизм представления лексико-семантической информации. URL: artyom.ice-ic.com/science/lexsem2004.doc

213. Мариничева М.К. Управление знаниями на 100%: Путеводитель для практиков. Москва: Альпина Бизнес Букс, 2008. 320 с .

214. Марченко А.В. Вступ. Моделі даних. URL: http://elearning.sumdu.edu.ua/free_content/lectured-20151030212503.pdf

215. Методы построения деревьев решений в задачах классификации в Data Mining. URL: https://ami.nstu.ru/~vms/lecture/data_mining/trees.htm

216. Мильнер Б.З., Румянцева З.П., Смирнова В.Г., Блинникова А.В. Управление знаниями в корпорациях; под редакцией Мильнера Б.З. Москва: «ДЕЛО», 2006.

217. Мильнер Б.З. Управление знаниями: Эволюция и революция в организации. Москва: ИНФРА-М, 2003. 177 с.

218. Миркин Б.Г. Методы кластер-анализа для поддержки принятия решений: обзор : препринт WP7/2011/03 [Текст] / Б. Г. Миркин; Национальный исследовательский университет «Высшая школа экономики». Москва: Изд. дом Национального исследовательского университета «Высшая школа экономики», 2011. 88 с.

219. Павленко Т.В., Ключнікова М.В. Особливості управління знаннями в наукоємних підприємствах. URL: <http://economy.kpi.ua/uk/node/724>

220. Палагин А.В., Крытый С.Л., Петренко Н.Г. Онтологические методы и средства обработки предметных знаний: монография. Луганск: изд-во ВНУ им. В. Даля, 2012. 324 с. [

221. Палагин А.В., Петренко Н.Г., Крытый С.Л. К вопросу о построении знание-ориентированных компьютерных систем для научных исследований. УСиМ. 2015. № 2. С. 64–73.

222. Пасічник В.В. та ін. Глобальні інформаційні системи та технології: моделі ефективного аналізу, опрацювання та захисту даних. Монографія / В.В. Пасічник, П. І. Жежнич, Р. Б. Кравець, А. М. Пелешишин, Д. О. Тарасов. Львів: Видавництво Львівської політехніки, 2006. 348 с. ISBN: 966-553-578-1.

223. Переверза К.В. Методологія побудови сценаріїв розвитку складних соціальних систем із використанням морфологічного та swot-аналізу: ISSN 1681-6048 system research & information technologies, 2012. № 4.

224. Проект Semantic Web. URL: <http://www.w3c.org/sw>

225. Решения Azure. URL: <https://azure.microsoft.com/ru-ru/solutions/>

226. Ракитянська Г.Б.: Побудова класифікаційної нечіткої бази знань на основі трендових правил і оберненого виведення. *Восточно-Европейский журнал передовых технологий*. 1/3 (73) 2015/С. 25–32. ISSN 1729-3774.

227. Резниченко В.А., Чистякова И.С. Отображение дескриптивной логики алс в бинарную реляционную структуру данных. *Проблемы программирования*. 2015. № 4. С. 13–30. ISSN 1727-4907.

228. Саак А.Э., Тушняков В.Н. Разработка управленческого решения: учебн. для вузов. Санкт-Петербург: Изд-во «Питер», 2007. 272 с.

229. Саати Т.Л. Принятие решений. Метод анализа иерархий. Москва: Радио и связь, 1993.

230. Садовенко В.С., Понєжа Г.В. Бази даних і бази знань у державному управлінні: навч. посіб. для слухачів до курсу «Інформаційні технології в державному управлінні». – Київ: Вид-во НАДУ, 2007. 104 с.

231. Сайт редактора онтологий Стэнфордского университета. URL: <http://protege.stanford.edu/>
232. Самойлов В.Д. Модельное конструирование компьютерных приложений. Киев. Наук. думка, 2007. 198 с.
233. Серіков А.В. Управління організаційними змінами: навч. посіб. – Харків: Фірма «БУРУН і К», 2013. 264 с.
234. Смирнов С.В.: Онтологическое моделирование в ситуационном управлении. URL: <http://cyberleninka.ru/article/n/ontologicheskoe-modelirovanie-v-situatsionnom-upravlenii>
235. Сценарии для расширенной аналитики в Машинном обучении Azure. URL: <https://docs.microsoft.com/ru-ru/azure/machine-learning/team-data-science-process/plan-sample-scenarios>
236. Субботин С.А. Неитеративный синтез нейро-нечетких диагностических экспертных систем. URL: <http://www.researchgate.net/publication/247158372>
237. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: навч. посіб. – Запоріжжя: ЗНТУ, 2008. 341 с.
238. Трофимова Л.А., Трофимов В.В.: Управленческие решения (методы принятия и реализации): учеб. пособ. – Санкт-Петербург: Изд-во СПбГУЭФ, 2011. 190 с.
239. Тузовский А.Ф., Чириков С.В., Ямпольский В.З.: Системы управления знаниями (методы и технологии) / под общ. ред. В.З. Ямпольского. Томск: Изд-во НТЛ, 2005. 260 с.
240. Тузовский А.Ф: Формирование семантических метаданных для объектов системы управления знаниями Институт «Кибернетический центр» ТПУ. Томский научный центр СО РАН.
241. Фаулер М. Предметно-ориентированные языки программирования. пер. с англ. / Мартин Фаулер. Москва: ООО «И.Д. Вильямс», 2011. 576 с.

242. Форман Д. Data Smart: Using Data Science to Transform Information into Insight / пер. с англ. Москва: Альпина Паблишер, 2016. 461 с.

243. Цуканова О.А. Методология и инструментарий моделирования бизнес-процессов Санкт-Петербург: Университет ИТМО, 2015. 100 с. URL: <https://books.ifmo.ru/file/pdf/1720.pdf>

244. Хайкин С. Нейронные сети. Полный курс 2-е изд. / пер. с англ. Москва: Издательский дом «Вильямс», 2006. 1104 с.: ил. Парал. тит. англ.

245. Шульц В.Л., Кульба В.В., Шелков А.Б., Чернов И.В. Сценарный анализ в управлении региональной безопасностью. *Вопросы безопасности*. 2016. № 3. С. 41–79. DOI:10.7256/2409-7543.2016.3.19157. URL: https://nbpublish.com/library_read_article.php?id=19157

246. Черняк Леонид. Большие Данные – новая теория и практика. Открытые системы. СУБД. Москва: Открытые системы, 2011. № 10. ISSN 1028-7493.

247. Malcolm Barrett, An Introduction to Directed Acyclic Graphs. URL: <https://cran.r-project.org/web/packages/ggdag/vignettes/intro-to-dags.html>

248. The Single Source Shortest Path algorithm. URL: <https://neo4j.com/docs/graph-algorithms/current/algorithms/single-source-shortest-path/>

249. Graph Algorithms in Neo4j: Single Source Shortest Path. URL: <https://dzone.com/articles/graph-algorithms-in-neo4j-single-source-shortest-p>

250. Decision Tree Classification in Python. URL: <https://www.datacamp.com/community/tutorials/decision-tree-classification-python>

251. Python 3.6.4. URL: <https://www.python.org/downloads/release/python-364/>

252. Python 3.7.3 documentation. URL: <https://docs.python.org/3/>

253. Top Cloud Computing Service Provider Companies. URL: <https://www.softwaretestinghelp.com/cloud-computing-service-providers/>

254. Cloud products / Analytics. URL: <https://aws.amazon.com/products/>

255. Big-data. URL: <https://cloud.google.com/products/big-data/>

256. DelleMC Digital universe of opportunities. URL: <https://blog.dellemc.com/en-us/digital-universe-2014/>

257. Laney D. 3D Data Management: Controlling Data Volume, Velocity and Variety. URL: <https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data>

258. Шульц В.Л., Кульба В.В., Шелков А.Б., Чернов И.В. Управление региональной безопасностью: анализ эффективности мониторинга социальной стабильности.

259. The Basics of Model Driven Architecture, Cephias Consulting Corp – January 2006. URL: https://www.omg.org/mda/mda_files/Cephias_MDA_Fast_Guide.pdf

260. Business Process Management Software. URL: <https://www.capterra.com/business-process-management-software/>

261. ISO/IEC 19510:2013(E) Business Process Model and Notation, Information technology – Object Management Group. URL: <https://www.omg.org/spec/BPMN/ISO/19510/PDF>

262. Robert Singer, An Ontological Analysis of Business Process Modeling and Execution. URL: https://www.researchgate.net/publication/332833270_An_Ontological_Analysis_of_Business_Process_Modeling_and_Execution

263. Marco ROSPOCHER, Chiara GHIDINI and Luciano SERAFI, An ontology for the Business Process Modelling Notation. URL: <https://dkm-static.fbk.eu/resources/ontologies/BPMN/BPMNontology.pdf>

264. Christine Natschlager Towards a BPMN 2.0 Ontology. URL: https://www.researchgate.net/publication/221448529_Towards_a_BPMN_20_ontology

265. BPMN 2.0 OWL від Institute for Public Policy Research. URL: <https://github.com/reiterma13/IPPR>

266. Ресурс PM2ONTO: [Online]. URL: <https://github.com/lukasriehl/pm2onto>

267. Ресурс BPMN-to-S-BPM-Ontology: [Online]. Available: <https://github.com/reiterma13/>

268. Копп А.М., Орловський Д.Л. Підход к анализу и оптимизации моделей бизнес-процессов в нотации BPMN. *Радіоелектроніка, інформатика, управління*. 2018. № 2. URL: http://www.kb-ametist.com/rus/18_publish/10.pdf. 23.

269. *Amina Annane, Nathalie Aussenac-Gilles, Mouna Kamel: BBO BPMN 2 Based Ontology for Business Process Representation*. URL: https://oatao.univ-toulouse.fr/25024/1/annane_25024.pdf

270. Додонов О.Г., Сенченко В.Р., Коваль О.В., Бойченко А.В. Моделювання сценаріїв аналітичної діяльності на основі нотації BPMN OWL. *Реєстрація, зберігання і обробка даних*. 2020. Т. 22, № 1. С. 31–48.

271. Business Process Model and Notation v2.0, URL <https://www.omg.org/spec/BPMN/2.0/PDF>

272. Bizagi Modeler, User Guide. URL: <http://help.bizagi.com/process-modeler/en/>

273. Офіційний сайт: Bizagi. URL: <https://www.bizagi.com/en/>

ДОДАТОК А. Побудова та реалізація сценаріїв моделювання процесів функціонування вимірювальної гідроакустичної системи

Аналіз динамічних технічних систем (тобто систем, що включають взаємопов'язані і взаємодіючі елементи), характерною особливістю яких є відсутність жорсткого алгоритму аналізу, передбачає застосування аналітичних інструментів в різній послідовності з варіаціями параметрів кожного конкретного кроку, що доцільно виконувати за використання сценарно-цільового підходу, тобто як за використанням типових сценаріїв аналізу, так і за використанням методів та програмних інструментів побудови та виконання динамічних сценаріїв. До таких систем відноситься вимірювальна система прийому і обробки гідроакустичних сигналів в морському середовищі (ВС). Проектування такої системи є складним процесом і вимагає попереднього моделювання функціонування ІС і аналізу результатів моделювання з метою визначення найбільш ефективних конструктивних рішень і алгоритмів обробки сигналів.

Експериментальним дослідженням гідроакустичних, електричних та магнітних процесів, пов'язаних з вивченням особливостей випромінювання морських об'єктів, кінцевою метою яких є класифікації та ідентифікації цих об'єктів, на сьогоднішній час приділяється значна увага.

Процедури класифікації та ідентифікації об'єктів відбуваються на основі використання ряду відповідних алгоритмів, таких як: генерації сигналів, виділення окремих сигналів (акустичного, електричного, магнітного) з поточного стану полів, зафіксованих у акваторії, та аналізу особливостей відокремлених сигналів. По результатам аналізу проводиться класифікація та ідентифікація об'єктів.

Зазвичай такі процедури виконуються на основі натурних експериментів або з застосуванням засобів комп'ютерного моделювання.

Детальні описи натурального експерименту можна знайти у роботах присвячених експериментальній перевірці та уточненню структури шумів

слоїстого океану. Зокрема, під час проведення натурних експериментів застосовувалися випромінювачі низькочастотних сигналів, які випромінюють сигнали з приведеним тиском 30-100 Па, та максимальні глибини занурення досягали 80-100 метрів. Під час проведення експериментів проводилося вимірювання гідрологічних характеристик водних акваторій. Прийняті сигнали реєструвалися на магнітних реєстраторах, а їх обробка проводилася в лабораторних умовах на спеціальних стендах і обчислювальних комплексах. При виконанні робіт аналізувалися морські шуми, визначалися локальні джерела, визначалися дальності, пеленги, швидкості і за цими даними будувалися траси руху морських об'єктів. Опрацювання результатів натурних експериментів, які проводилися в різних акваторіях з різними об'єктами підтверджують високу ефективність дослідження спектрально-енергетичних характеристик виділених сигналів випромінюючих морських об'єктів для вирішення задач їх ідентифікації та класифікації. При цьому виникає проблема оцінки ефективності процедур ідентифікації та класифікації для достовірного дослідження об'єктів. Однак роботи, пов'язані з проведенням кожного окремого натурального експерименту є вкрай ресурсовитратними та вимагають наявності засобів проведення відповідних технологічних процедур: фіксації умов експерименту, кроків експерименту і ретельного їх опису для зберігання і подальшого поглибленого аналізу і обробки. Крім цього натурні експерименти передбачають використання різноманітних засобів фіксації сигналів та потребують врахування різноманітності їх спектрально-енергетичних характеристик

Таким чином, проведення натурального експерименту потребує чіткого визначення місця, умов та задіяних об'єктів, а також ретельного опису послідовності проведення окремих кроків експерименту, що можна умовно звести до процедури організації сцени та сценарію проведення експерименту.

Зазначені складності проведення натурних експериментів приводять до визнання необхідності застосування засобів комп'ютерного моделювання гідроакустичних експериментів, в основу якого покладаються інформаційні

моделі сцен і сценаріїв та комп'ютерні моделі процесів розповсюдження, виділення та аналізу гідроакустичних сигналів.

При цьому, слід зауважити, що сценарії комп'ютерних експериментів дослідження гідроакустичних процесів (ГАП) мають яскраво виражену залежність від цілі експерименту та виду використаних моделей. Подібні системи вимагають створення різнопланових імітаційних моделей ГАП та побудови адаптивних до цілій проведення експериментів методик моделювання. Основною привабливою властивістю таких систем є здатність накопичення, покрокове вироблення та повторне використання знань (формалізованих посилок, висновків, закономірностей, що відносяться до досліджуваного об'єкта) в режимі реального процесу моделювання поведінки СДО.

У програмному комплексі, якій моделює функціонування ВС в морському середовищі (МК), предметом аналізу є сама вимірювальна система, морське середовище, в якій поширюються гідроакустичних сигнали, і об'єкти, що випромінюють ці сигнали. На основі програмних інструментів генерації, розпізнавання, аналізу гідроакустичного сигналу формуються висновки щодо класифікації та ідентифікації випромінюючих об'єктів для вжиття заходів, що виключають або обмежують можливі негативні наслідки від поведінки ідентифікованих об'єктів.

A1. Створення онтології предметної області програмного комплексу моделювання функціонування вимірювальної системи в морському середовищі

Базою для побудови та виконання сценаріїв аналізу функціонування ВС в морському середовищі є онтологія МК, яку доцільно розпочати розробляти з визначення її обсягу та масштабу. Тобто знайти відповідь на кілька основних питань:

- Яка ПрО охоплюється онтологією?
- Якого типу питання має відповідати та інформація, яка є в онтології?
- Хто буде використовувати та підтримувати онтологію?

Звичайно, відповіді на ці питання змінюються під час процесу розробки онтології, але в будь-який час вони дозволяють обмежити масштаб онтології, якщо вона стане занадто складною.

Основною якістю таких систем як МК є накопичення двох типів інформації:

- реальні дані - гідроакустичні в результаті сканування морської території;
- дані моделювання - отримується в результаті математичного та комп'ютерного моделювання поведінки об'єкта.

Моделювання гідроакустичних процесів потребує розробки складових моделей, включаючи інформаційну модель акваторії, базу даних параметрів морського середовища (морський шум, ґрунт, берегова лінія, температура води, глибина, солоність) та параметри гідроакустики пристроїв та їх взаємодія з носієм моделювання.

Гідроакустичні процеси дозволяють враховувати як проблеми прямого моделювання (з метою отримання об'єктивних даних - знань про досліджувані морські об'єкти), так і поєднувати отримані дані з експертними знаннями (представленими у вигляді характерного набору параметрів реальних об'єктів і їх оцінки на основі досвіду експерта та результатів попереднього попиту використання подібних ВС систем).

Узагальнена структура системи моделювання на основі знань для ідентифікації, класифікації та визначення параметрів руху морських споруд, наведена на рисунку А.1. Концептуальна модель складається з таких структурних компонентів:

- Імітація морського середовища - виконує функції створення імітаційної сцени (параметри глибини, температури та солоності, тип дна, берегової лінії тощо) та розташування та специфікацію параметрів морських об'єктів (тип, розмір, напрямок, швидкість) тощо);



Рисунок А1 - Структурні компоненти МК

- Обслуговування бази знань - містить набір інструментів для перевірки моделей розпізнавання об'єктів, їх ідентифікації, класифікації та визначення параметрів руху об'єктів, включаючи методи нечіткої логіки;
- Підтримка бази знань - призначений для організації засобів формування бази знань для вирішення проблем ідентифікації та класифікації морських об'єктів;
- Оцінювання результатів моделювання - призначений для організації логічного висновку на основі накопичених знань, включаючи засоби складання правил продукту, самонавчання та адаптацію;

– Зберігання інформації - є ядром системи і надає інформацію всім структурним компонентам системи моделювання, а також містить інформацію з експертними оцінками проведених експериментів;

– Адміністрування та управління - надають налаштування для послуг та програм, керують правами доступу користувачів до інформаційних ресурсів, керують безпекою та продуктивністю системи моделювання.

МК покликаний вирішувати задачі комп'ютерного моделювання наступних процесів:

1. розповсюдження акустичних сигналів у водному середовищі;
2. виділення сигналів, що надійшли від різних шумовипромінюючих об'єктів;
3. аналіз виділених сигналів за амплітудно - частотними характеристикам шумовипромінювання об'єкта(джерела звуку) та зафіксованими ГАС (гідроакустичною системою-приймачем) параметрами згенерованої звукової хвилі, а саме тиску та складових коливальної швидкості;
4. ідентифікація сигналу тобто визначення траси та параметрів руху джерела;
5. класифікації шумовипромінюючих об'єктів за класами поверхневий або підводний;
5. ідентифікації об'єктів на основі гідроакустичних портретів (або сигнатур) з застосуванням технології штучного інтелекту та використанням відповідної бази знань

Запропонована архітектура МК надає можливості для наскрізного документування процесів гідроакустичних експериментів, що дає додаткові переваги у формуванні та накопиченні знань про досліджувані процеси, включаючи формування сцен та моделювання сценаріїв. З метою можливості здійснення моделювання сценаріїв аналізу функціонування МК було забезпечено вирішення наступних задач:

1. Створення та обслуговування бібліотеки гідроакустичних моделей, включаючи: моделі генерації сигналів, моделі вилучення сигналу, моделі аналізу сигналів, реалізація алгоритмів класифікації та ідентифікації об'єктів.

2. Створення та технічне обслуговування бази знань, введення даних результатів моделювання та реальних даних (сканування акваторій).

3. Розробка онтології ПрО.

4. Розробка алгоритмів та програмного забезпечення розпізнавання та ідентифікації типів морських об'єктів.

5. Організація логічного висновку за використанням онтологічної моделі ПрО.

6. Створення засобів навчання МК (сценарії та алгоритми навчання з урахуванням ПрО).

7. Формування правил на основі нечіткого логічного висновку.

8. Розробка алгоритмів ідентифікації розвитку на основі логічного висновку за онтологією та правилами виробництва.

9. Розробка алгоритмів класифікації та кластеризації знань.

10. Формування класифікаторів системи (об'єкти, що випромінюють шум, гідроакустичні системи, акваторії).

11. Формування сцени та моделювання сценарію експерименту.

12. Закріплення результатів експерименту в БД.

13. Оцінка результатів моделювання.

Вирішення задачі формування сцени та моделювання сценарію експерименту повинне передбачати наявність засобів визначення:

– сцени проведення експерименту, на який фіксуються умови проведення експерименту (район проведення, параметри акваторії), сукупність об'єктів, які беруть участь у експерименті, набір приладів фіксації сигналів (їх розташування, характеристики);

– сценарію виконання експерименту, який визначає: параметри переміщення об'єктів (напрями, швидкості), параметри випромінювання

(частота, амплітуда), умови розповсюдження гідроакустичних сигналів (профіль та характеристики дна акваторії), параметри стану поверхневих вод (ступінь хвилювання, сила вітру), параметри процесу моделювання гідроакустичних сигналів (частота дискретизації, формати збереження результатів).

Проектування МК базувалось на онтологічному підході з визначенням концептів предметної області та відношень між ними на основі схем проведення експериментів. Онтологічна модель Ont (МК) містить основні поняття (сутності - базові поняття предметного домену), їх атрибути та описує відносини між ними і може бути представлена у вигляді:

$$Ont(SD)_{IS} = \langle C^{(Ax)}, Ex^{(C)}, Rel^{(H)}, T^{(Q)}, Ax^{(s)}, Rul^{(S)} \rangle$$

де: $C^{(Ax)}$ - класи (Classes) - скінченний набір основних понять гідроакустичних процесів;

$Ex^{(C)}$ - скінченний набір класів онтології набору Exemplars;

$Rel^{(H)}$ - кінцевий набір відносин між класами та їх типами;

$T^{(Q)}$ - кінцевий набір атрибутів кожного класу, їх типи даних та поля значень;

$Ax^{(s)}$ - кінцевий набір аксіом - визначає основні поняття ПрО, які для неї завжди вірні;

$Rul^{(S)}$ - кінцевий набір правил логічного завершення.

Детальний розгляд концепцій ПрО та функціональних проблем призводить до виділення таких основних класів:

1. Experiment - клас $C^{(Ax)}$, що визначає характеристики проведеного експерименту - ідентифікаційний номер, дата проведення, ідентифікатор дослідника, дані служби.

2. Models_Hydroacoustic_processes - клас $C^{(Mod)}$, що описує різні моделі, методи та алгоритми генерації, відбору та аналізу сигналів, а також методи та алгоритми класифікації та ідентифікації морських об'єктів.

3. 3. Hydroacoustic_objects - клас $C^{(Obj)}$, складаються з деяких підкласів (морські об'єкти, підводні об'єкти та повітряні об'єкти) та описують об'єкти властивостей/

4. Aquatorium - це клас $C^{(Sea)}$ для опису морської зони, обраної для експерименту: тип акваторію, назву та параметри уточнення для алгоритмів моделювання типу (координати, глибини, температури, солоність).

5. Experiment_scene - клас $C^{(Sn)}$ для опису сцени моделювання з урахуванням характеристик морського регіону, а також характеристик об'єктів, які беруть участь в експерименті.

6. Hydroacoustic_System - клас $C^{(Div)}$, що описує координати, склад та типи гідроакустичних пристроїв.

7. Modeling_Scenario - клас $C^{(Sig)}$ для визначення послідовності окремих етапів моделювання. Кожен крок-це набір процедур "старт-біг-виправ результат".

8. Model_estimation - клас $C^{(Val)}$ для фіксації результатів моделювання та оцінки правильності моделей.

9. Acoustic signal - клас $C^{(Sig)}$, що визначає характеристики сигналу, включаючи дату виявлення сигналу та параметри блокувального пристрою.

10. Waveguide - клас $C^{(Noise)}$ для опису гідроакустичних перешкод, що впливають на поширення та спотворення гідроакустичного сигналу, а також засоби нейтралізації перешкод.

Визначення класів та створення їх ієрархії (таксономії) є ключовим у розвитку онтології. Таксономія класів - це дерево описових термінів, що мають ієрархічну структуру.

У редакторі Protege 5 створення класів $C^{(Ax)}$ відбувається у закладці <Classes>. У мові OWL класи інтерпретуються як підмножина індивідів, які є частиною визначеного класу. Особливістю проектування в середовищі Protege є те, що класи розглядаються як підкласи загальної онтології THING. Відповідно до позначення CamelCase для OWL - усі назви класів повинні починатися з великої

літери і не повинні містити пробілів. Для забезпечення класифікації просто потрібно натиснути кнопку <Додати підклас>. У вікні, що відкріється, потрібно ввести назву класу. (Рисунок А2).

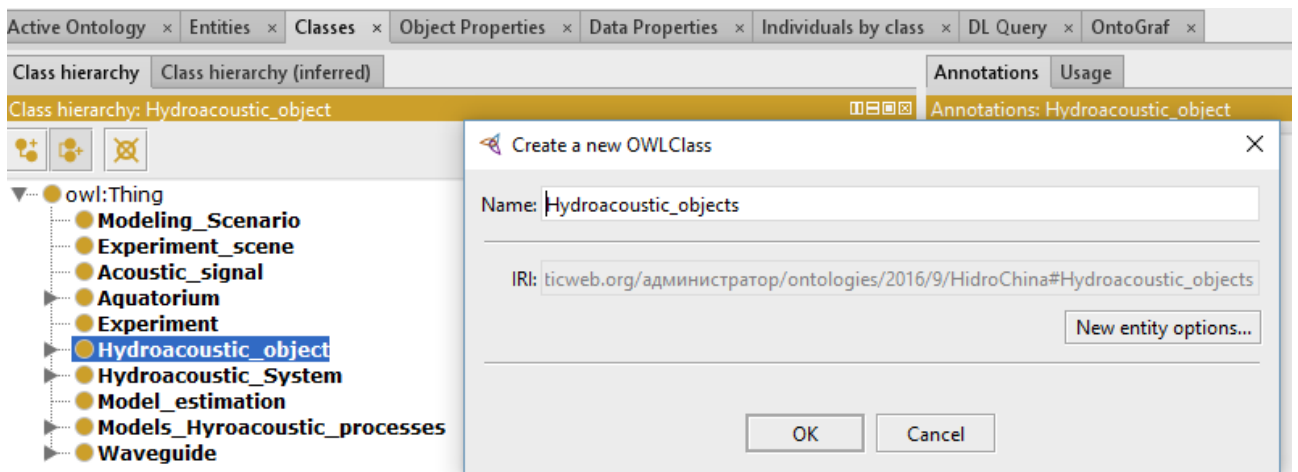


Рисунок А2 - Приклад створення класів

За замовчуванням класи в OWL можуть перетинатися. Щоб розділити класи, їх потрібно роз'єднати. Це гарантує, що особа не може бути екземпляром кількох класів. Для цього на вкладці <Classes> потрібно визначити клас, який не повинен перетинатися, потім у полі <Description> потрібно натиснути на + стороні функції Disjoint With та у вікні <Class hierarchy> відкрийте клас, який не повинен перекриватись із зазначеним класом (Рисунок А3).

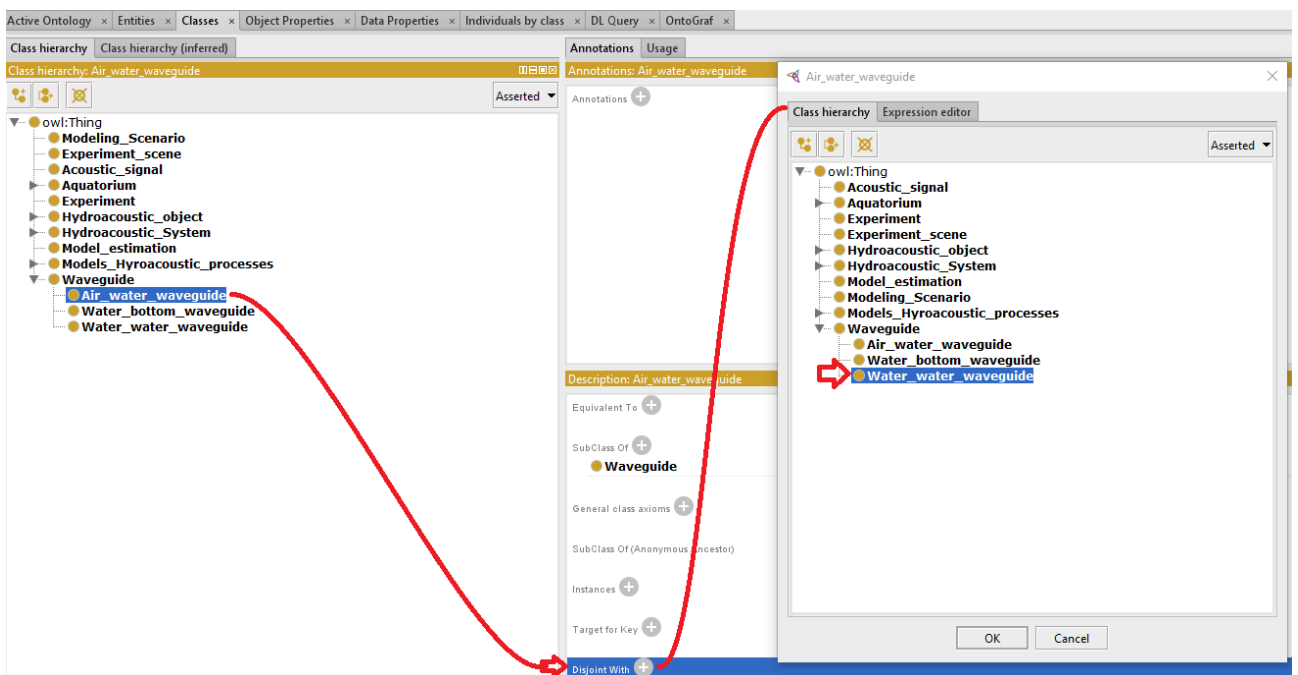


Рисунок А3 - Поділ класів на неподільні

У редакторі Protégé кожен клас визначається властивостями, що описують відносини між класами, і поділяється на два типи:

1. $\langle \text{ObjectProperty} \rangle$ - описує відношення $Rel^{(H)}$ (типи відносин), які встановлюються між окремими класами онтології.
2. $\langle \text{DatatypeProperty} \rangle$ - описує конкретні атрибути $T^{(Q)}$ (характеристики), що визначають клас. Наприклад, швидкість руху морського об'єкта, його розміри, технічні параметри тощо.

A1.1 Формування підмножини атрибутів ПрО

Підмножина атрибутів $T^{(A)}$ описує властивості класів $C^{(Ax)}$ і використовується для введення конкретних значень класів екземплярів - $Ex^{(C)}$.

Створення підмножини атрибутів редактором Protege було здійснено за допомогою вкладки $\langle \text{Datatype Properties} \rangle$ на панелі інструментів редактора Protégé 5. Далі відображається вікно, що містить вкладку $\langle \text{OWL: DataProperty} \rangle$. Коли натискається панель інструментів кнопки, відкривається вікно $\langle \text{Create a new OWLDataProperty} \rangle$, можна внести властивість імені, наприклад $\langle \text{Cruising Speed} \rangle$, і натиснути ОК (Рисунок A4). Після виконання відповідних дій, передбачених процесом, у лівій рамці з'являється новий тип властивості.

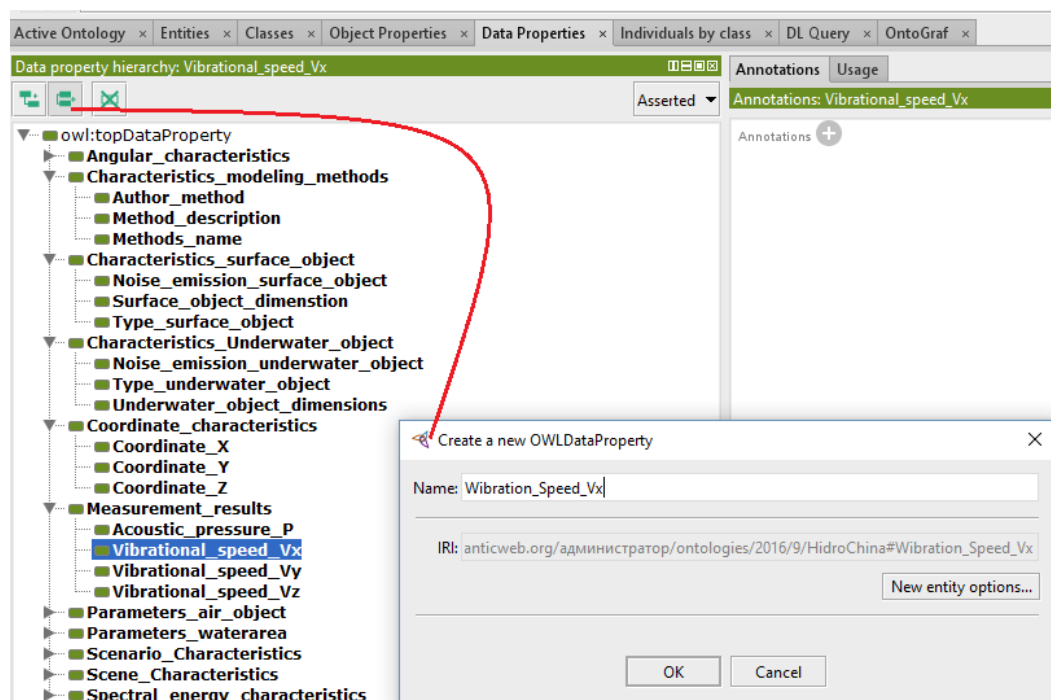


Рисунок A4 – Створення нового атрибуту

Попередньо визначені атрибути вказані у словнику формату XML і можуть бути представлені у різних форматах даних: цілі числа, плаваюча крапка, рядки, логічні значення тощо. На приклад, тег `owl: DatatypeProperty <URI>` у XML, який описує клас `<Underwater_object>`, прдставляється наступним чином:

```
<owl: DatatypeProperty rdf: about =
"http://www.semanticweb.org/svr/ontologies/ -9# ="# Underwater_object_dimensions
"/>.
```

A1.2 Формування екземплярів класів для наповнення бази знань

Приклади класів $Ex^{(C)}$ - це конкретні об'єкти ПрО (екземпляр), що належать певному класу $C_i^{(Ax)}$. Кожен клас $C_i^{(Ax)}$ описує підмножину атрибутів $T_i^{(A)}$.

Процес створення екземпляра виконується через вкладку <Особи за класом>, яка записується тегом типу

```
<owl: NamedIndividual rdf: about = "...">
```

Процес створення нового екземпляра складається з таких кроків:

1. Вибір класу за допомогою вкладки <Classes> або <Entities>.
2. Вибір вкладки <Особи за класом> на панелі інструментів редактора.

Активується нове вікно. У цьому вікні можна розробити екземпляр онтології.

3. Щоб створити новий екземпляр обраного класу (Southern seas), слід натиснути на кнопку, як показано на рисунку А5. Відкриється нове вікно <Creat a new OWLNamedIndividual>, призначене для введення імені екземпляра. Після виконання відповідних дій, передбачених процесом, у лівій рамці з'являється новий екземпляр вибраного класу, наприклад, <Taiwan Strait>.

Процедуру <Create a new OWL Named Individual> також можна використати для заповнення бази знань тестовими зразками. Для цього потрібно вибрати потрібний екземпляр зазначеного класу (Southern_Seas), як показано на рисунку А6, і в полі <Property assertions> з назвою екземпляра (Babco Bay) натиснути на значок <+> поруч із функцією

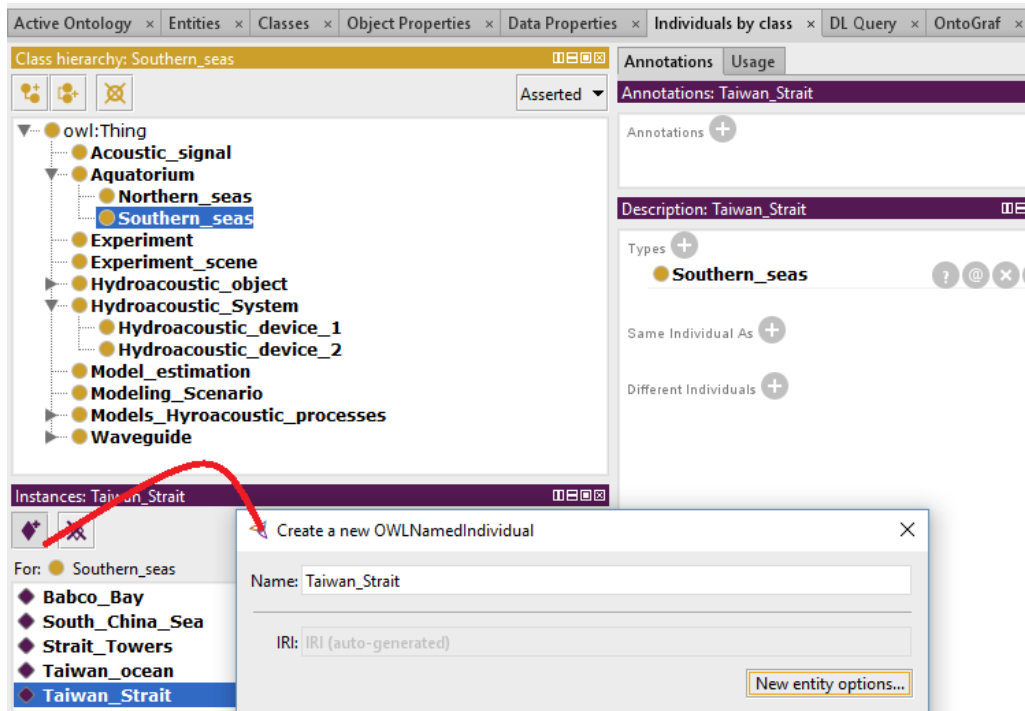


Рисунок А5 – Приклад додавання нового екземпляру класу

<Data Property assertions>. При натисканні на піктограму з'являється нове вікно <Data Property>, де потрібно в сутності <Owl: TopDataProperty>, яке вибирає відповідний атрибут, наприклад <Maximum_depth>, і вводить його значення у поле <Value> правого вікна (Рисунок А6). Після натискання кнопки ОК в базі онтології записується її значення. Оскільки екземпляр класу описується деякою підмножиною атрибутів, цю процедуру додавання вартості потрібно повторювати для кожного атрибута. Набір властивостей для різних екземплярів класів може бути абсолютно індивідуальним.

А1.3 Визначення та формування підмножини відносин та зв'язування класів

У редакторі Protégé кожен клас визначається властивостями, які описують відносини між класами. Властивості Object на вкладці <ObjectProperty> - це відносини між двома класами або окремими особами.

За допомогою вкладки <ObjectProperty> виконуються такі дії:

1. Формування підмножини відношення $Rel^{(H)}$ між класами $C^{(Ax)}$ у визначеній ПрО, додавання типів відносин до списку <owl: topObjectProperty>.

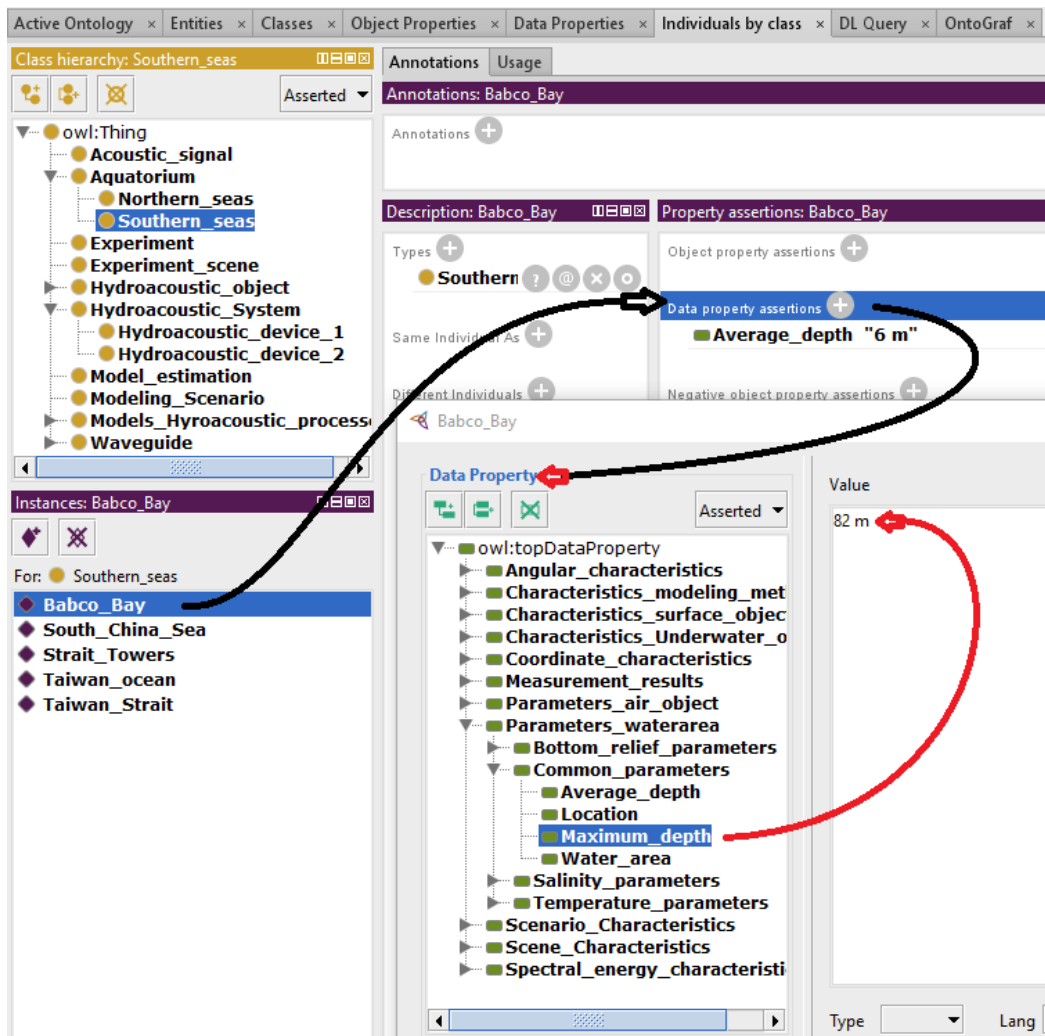



Рисунок А6 – Приклад введення значення певного атрибута

2. Формування підмножини Аксіоми $Ax^{(s)}$ у визначеній ПрО шляхом зв'язування певних онтологічних відносин класу за допомогою предикатів з $\langle owl: topObjectProperty \rangle$.

3. Формування відносин між суб'єктами за допомогою предикатів, що містяться в $\langle owl: topObjectProperty \rangle$.

А1.4 Формування підмножини відносин у визначеній предметній області

Щоб створити підмножину відносин та додати її до $owl: topObjectProperty$, необхідно вибрати вкладку $\langle ObjectProperty \rangle$ і перейти до рядка $owl: topObjectProperty$. Панель інструментів відкриває інструменти для опису відносин. Щоб додати новий тип відносин, потрібно натиснути кнопку . У

вікні з назвою <Create a new OWLObjectProperty> необхідно ввести ім'я типу відношення, наприклад, <has_communication_> (Рисунок А7) і натиснути ОК.

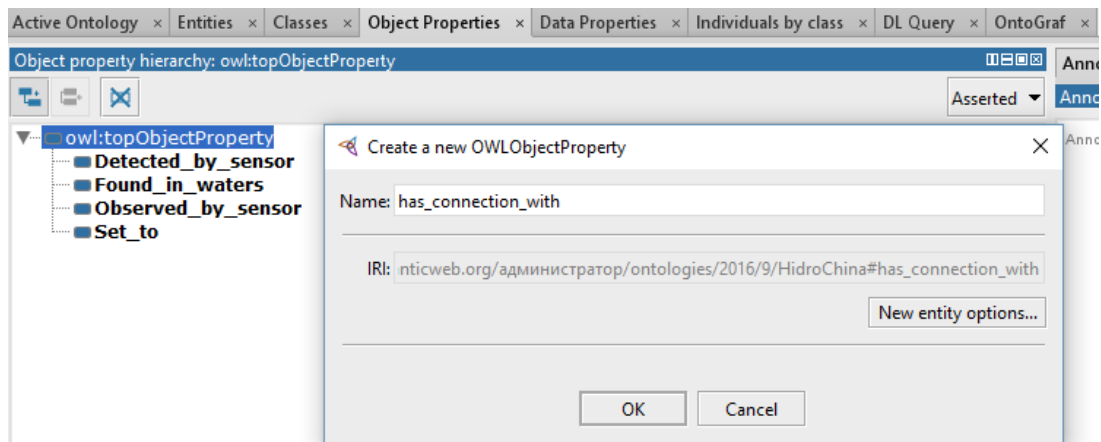


Рисунок А7 – Приклад формування нового типу відносин

Після виконання відповідних дій у підмножині <owl: topObjectProperty> з'являється новий тип з'єднання. Повторюючи цей процес, можна сформувати всю підмножину типів відносин для вибраної ПрО.

А1.5 Утворення підмножини Axioms

Під аксіомою розуміються твердження, які введені в онтологію в готовому вигляді, з яких можна вивести інші твердження. Аксіоми пов'язують два класи (поняття) з певними співвідношеннями:

- зв'язок між класами описується графом RDF онтології (Resource Description Framework);
- створені відносини між класами (аксіоми) описуються RDF-графом <subject-predicate-object>

$Subject C_{si}^{(Ax)}$ (Class _i)	$Rel_{ij}^{(H)}$ Predicate	$Object C_{oj}^{(Ax)}$ (Class _j)
--	-------------------------------	---

У редакторі Protégé 5 дві функції використовуються для визначення аксіом домену ПрО:

- Домени (перетин) - стверджує, що суб'єкти таких висловлювань про властивості мають належати до розширення класу зазначеного опису класу.

- Діапазони (перетин) - стверджує, що значення цієї властивості повинні належати до розширення класу опису класу або до значень даних у зазначеному діапазоні даних.

Для створення підмножини Аxioms необхідно вибрати вкладку <ObjectProperty> на панелях інструментів Protégé 5 і перейти до рядка owl:topObjectProperty. У підмножині <owl:topObjectProperty> вибрати потрібний тип відносин, який пов'язує два класи, наприклад клас <Experiment_scene>, пов'язаний з класом <Aquatorium> відносини <has_connection_with>.

Для цього на вкладці <ObjectProperty> у вікні <Description> потрібно натиснути на «+» поруч із функцією <<Domains (intersection)>>. Після натискання кнопки "+" відкриється окреме вікно, де потрібно вибрати вкладку <Class hierarchy> (Рисунок А8) та визначити клас (<Experiment_scene>), якому надано вибрану властивість.

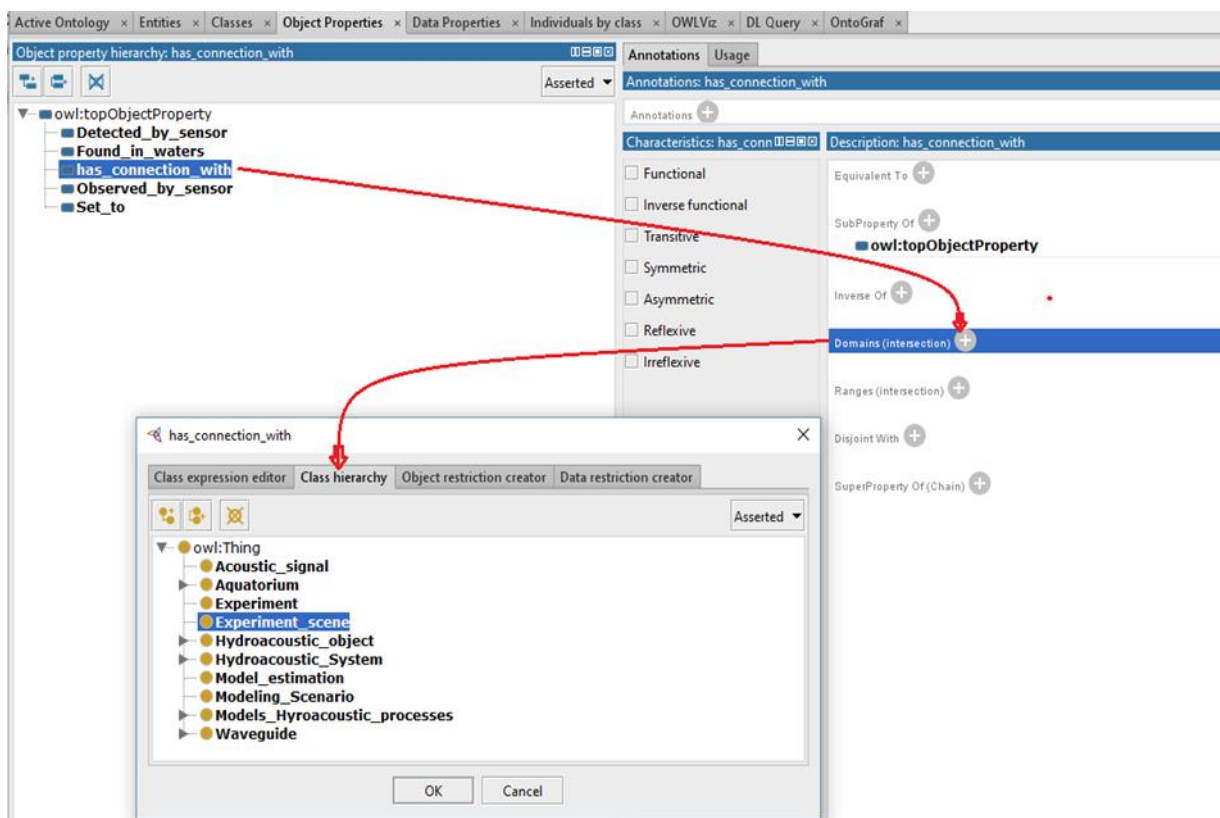


Рисунок А8 – Приклад надання вибраної властивості класу

Для вибору другого класу, потрібно натиснути на «+» поруч із функцією <Ranges (intersection)> у вікні <Description>. Крім того, у вікні, що відкриється, необхідно вибрати клас (<Акваторій>) та натиснути кнопку ОК (Рисунок А9).

Після виконання відповідних дій, новий зв'язок між класами буде створеним та збереженим в моделі онтології Subject Domain.

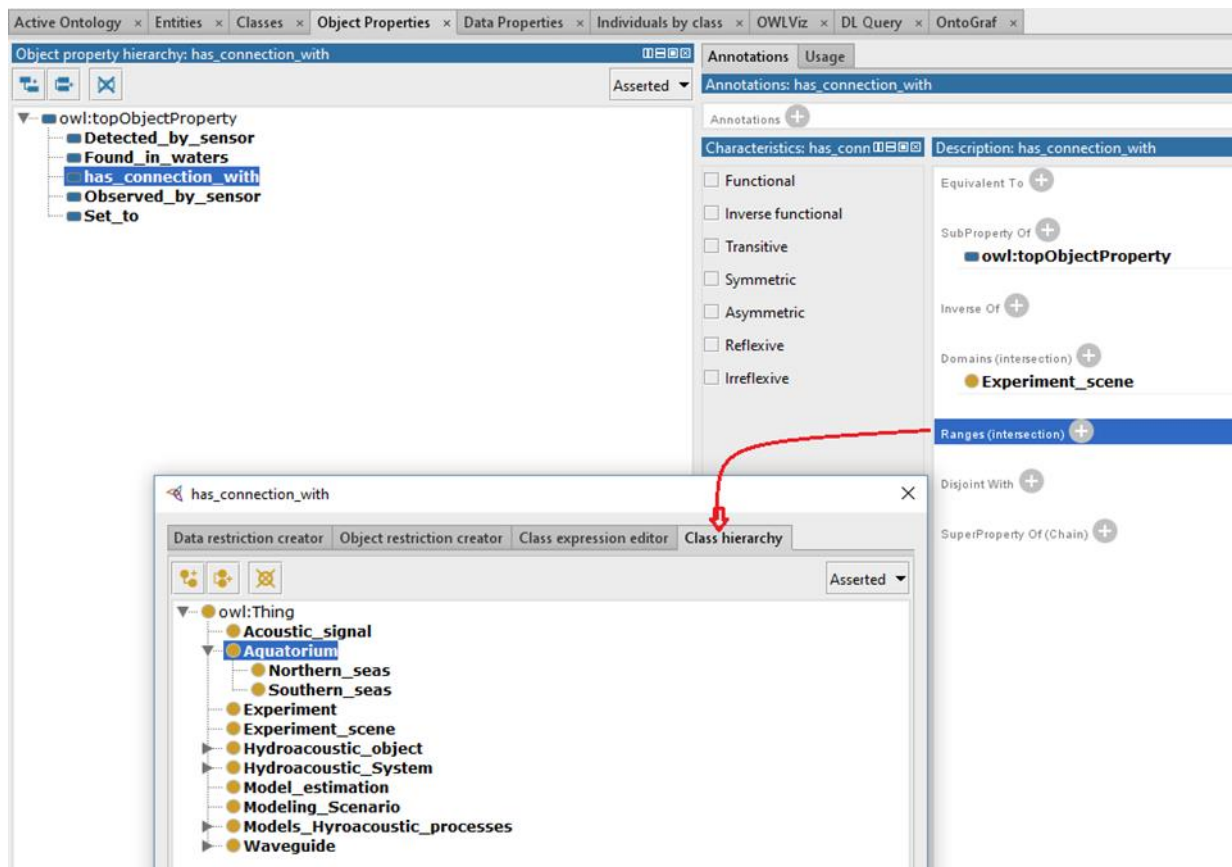


Рисунок А9 – Приклад створення відносин між класами

А1.6 Формування відносин між окремими суб'єктами

Щоб налагодити стосунки між двома суб'єктами у Protégé 5, потрібно перейти на вкладку <Entities> на панелі інструментів. У відкритому вікні <Individuals by type> є ієрархія класів, яка містить екземпляри. Далі в ієрархії класів (наприклад, <Helicopter>)) потрібно вибрати екземпляр (тему) і відкрити його (<BoeingCH-47>). Після цього відкриваються два вікна <Description> та <Property Assertions>. Для формування зв'язків між обраним суб'єктом потрібно натиснути на "+" поруч із функцією <Object Property Assertions >. Після цього відкриється нова форма для введення двох полів <Object property name> та <Individual name> (Рисунок А10). Наприклад, окремий <BoeingCH-47> характеризується високим рівнем акустичного шуму, граничні значення якого записуються в окремому <Big_acoustic_noise>. У поле <Object property name>

потрібно ввести відношення <Has_acoustic_noise>, а в полі <Individual name> зареєструвати суб'єкта <Big_acoustic_noise> та натиснути ОК.

Після виконання відповідних дій нові відносини між суб'єктами будуть створені та збережені у моделі онтології суб'єктного домену ПрО.

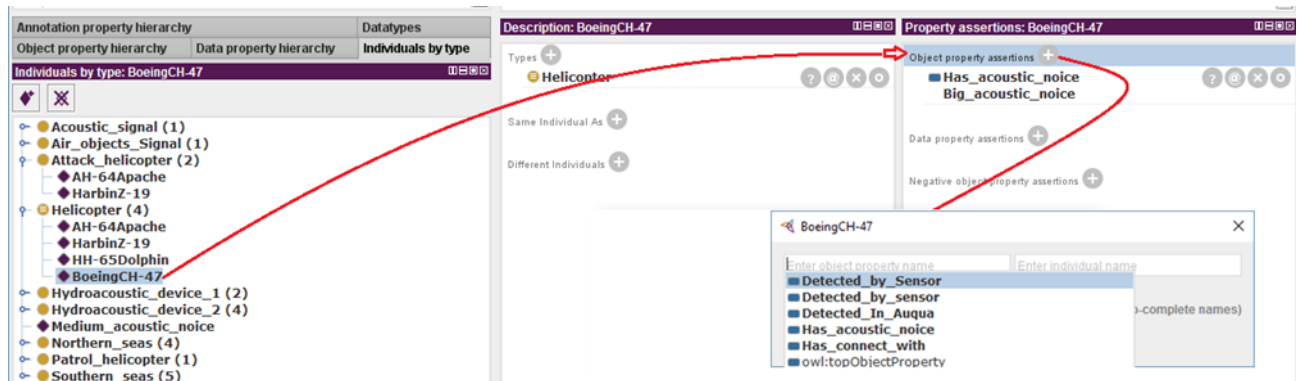


Рисунок А10 – Приклад створення відносин між окремими суб'єктами

Ієрархія класів, побудована за допомогою редактора Protégé, показана на рисунку А11.

Онтологія ПрО включає 34 класи, кожен з яких має від 5 до 15 атрибутів; описано майже 30 аксіом: 23 асоціативні відносини, 15 відносин «є-є», майже 70 відносин спадковості та 50 відносин «клас-дані», крім того, виділяються стандартні типи значень атрибутів та межі значень атрибутів.

Наприклад, акустичний сигнал - клас $C^{(Sig)}$ має такі атрибути:

- Frequency of the signal
- Phase
- Amplitude
- Interference level
- Spectral-energy characteristics
- Vector-phase characteristics to identify the object
- Coefficient of total attenuation (pressure signals, Speed along the axes - V_x , V_y , V_z)
- Location coordinates
- Distance to the object,
- Speed of propagation,

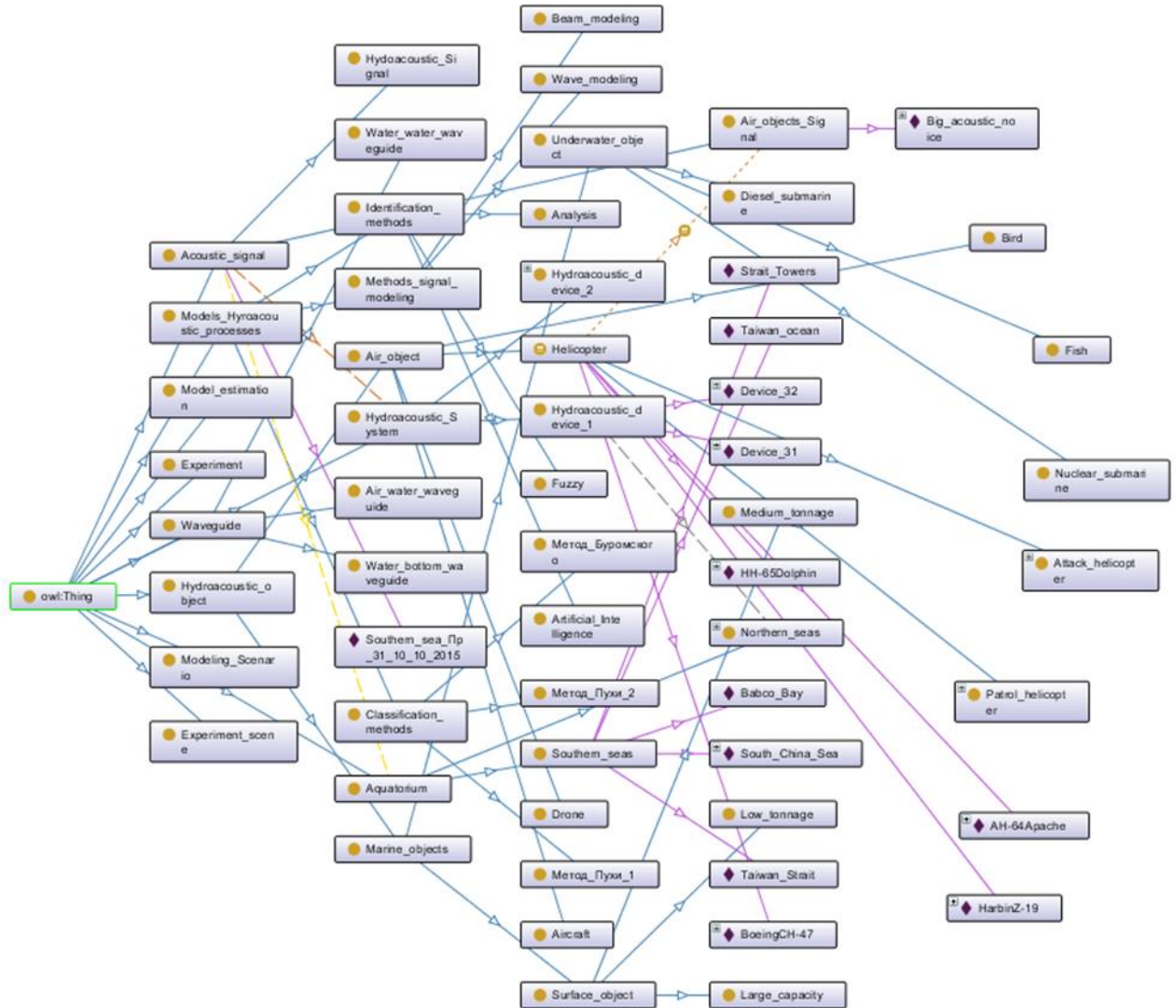


Рисунок А10 – Приклад графу моделі онтології ідентифікації сигналу

- Position of the object relative to the receiver,
- Pressure at the depth of the receiver,
- The results of modulation of the noise spectra of objects
- Coefficient of noise emission.

Після опису всіх класів та їх окремих осіб та встановлення всіх відношень до них можна використовувати алгоритми описової логіки для обробки онтологічної інформації, а також побудувати граф онтологічної моделі.

Як приклад, на рисунку А11 представлена екранна форма, яка відображає граф побудованої онтологічної моделі формування середовища моделювання гідроакустичних процесів.

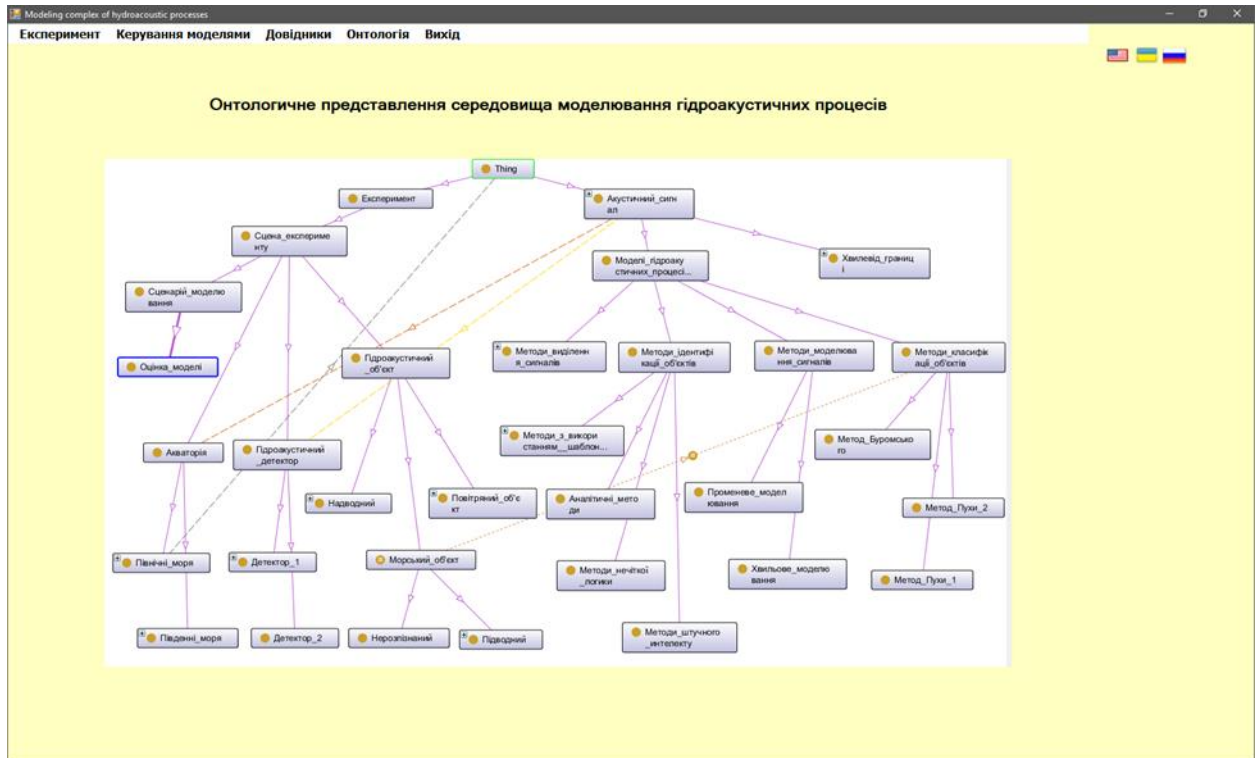


Рисунок А11 – Екранна форма відображення онтологічної моделі формування середовища моделювання гідроакустичних процесів

А2. Методика моделювання функціонування ВС за сценарно-цільовим підходом.

Моделювання функціонування ВС являє собою певну систему проведення модельних гідроакустичних експериментів з використанням елементів сценарного підходу. Сутність сценарного підходу полягає у створенні сценаріїв аналізу процесів моделювання, тобто деяка формалізована фіксація послідовності застосування інструментів моделювання та процедур оцінювання якості обраних моделей із завданням параметрів, які визначають правила їх використання. МК повинен містити засоби, що дозволяють зафіксувати сценарії, організувати їх виконання, а також сформуванню поетапну оцінку як роздільної здатності методів моделювання, так і оцінку результатів, отриманих за їх допомогою.

На основі знань експертів або попереднього моделювання гідроакустичного поля формується логіка розрахунку гідроакустичного поля.

Сценарій інтерпретує і реалізує розроблену у вигляді логіки моделювання експерименту на трьох рівнях:

- на рівні вибору раціональної моделі розрахунку акустичного поля;
- на рівні покрокової оптимізації обчислювального процесу при проведенні обчислень по заданій трасі поширення акустичної енергії;
- на рівні оптимізації обчислень параметрів гідроакустичного поля трас, що примикають по азимуту до напрямку раніше розрахованої траси. Покрокова оптимізація обчислювального процесу полягає у виборі техніки розрахунку параметрів акустичного поля на поточному кроці за результатами обчислень на попередньому кроці і з урахуванням аналізу апроксимованих на поточному кроці вихідних даних середовища. За допомогою такого підходу з одного боку підвищується оперативність вирішення завдань через пропуск незначущих точок розрахунку, а з іншого боку - точність обчислень через відповідне використання математичних методів. Крім покрокової оптимізації по заданій трасі, забезпечується оптимізація розрахунку сусідніх трас або просторових каналів.

Загальна схема проведення модельних експериментів з використанням сценарного підходу наведена на рисунку А12.

Перед безпосередньо аналізом Функціонування ВС слід виділити якесь операційне поле, яке представляє собою обмежену частину області дослідження, яка аналізується, і визначити аналітичні інструменти, використовувані для аналізу в межах виділеного операційного поля. В МК операційним полем є виділена частина морської акваторії із зафіксованими на ній об'єктами двох видів - об'єкти, що випромінюють звукові сигнали, і об'єкти, здатні фіксувати їх.

Першим кроком моделювання є формування переліку експериментів з наданням облікових параметрів для кожного з них. Загальній вигляд форми реєстрації експерименту наведено на рисунку А13.

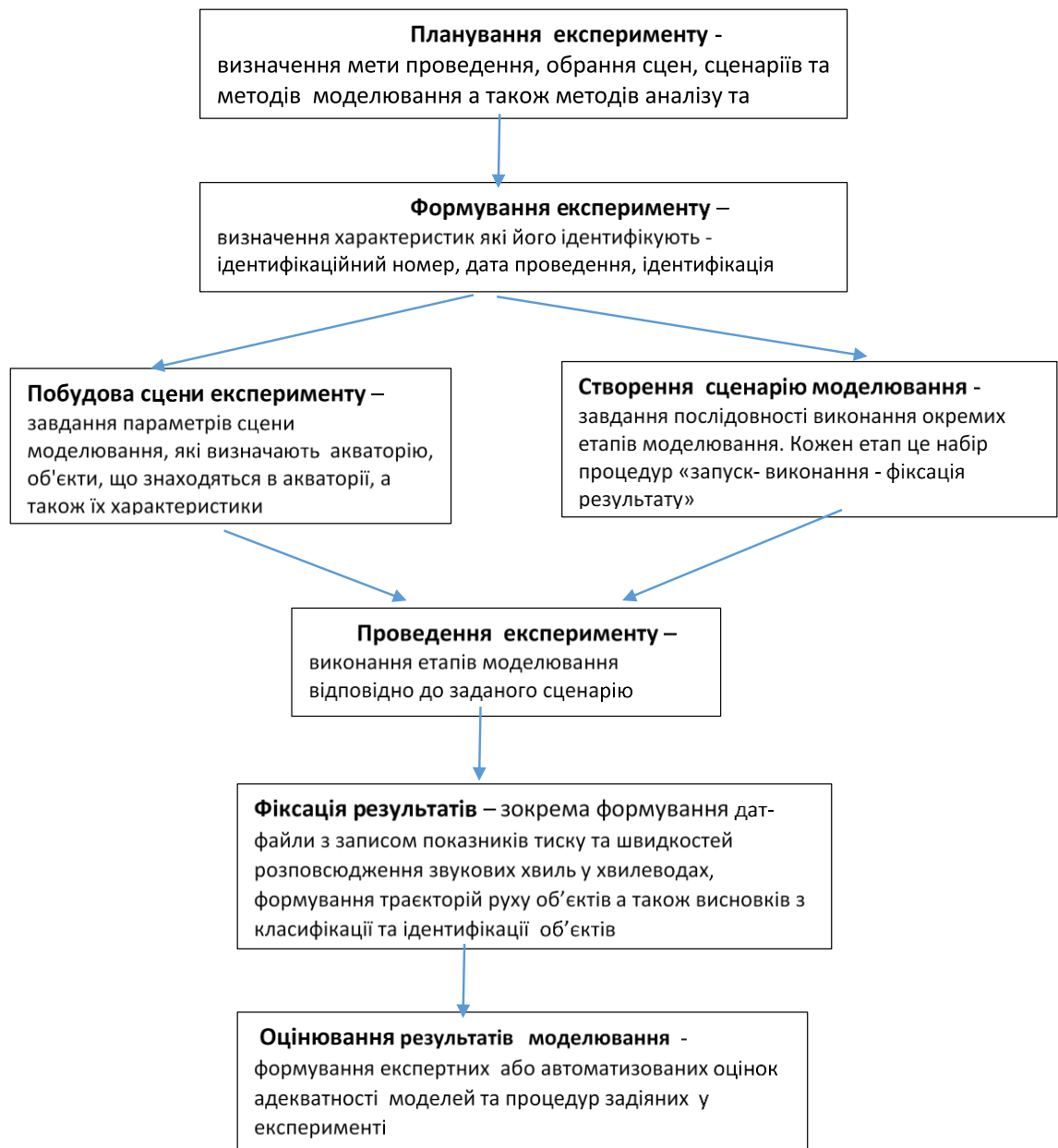


Рисунок А12 – Загальна схема проведення експериментів з використанням сценарного підходу

Другим кроком моделювання є формування переліку сцен окремо для кожного експерименту з наданням облікових параметрів для кожного з них. Загальний вигляд форми реєстрації сцени наведено на рисунку А14.

Третім кроком моделювання є побудова зареєстрованої сцени експерименту, яка полягає у визначенні місця розташування ГАС на сцени, а

також у визначенні розташування на сцені звуковипромінюючих об'єктів. експерименту з наданням облікових параметрів для кожного з них. Загальний вигляд екранної форми реєстрації сцени наведено на рисунку А15.

The screenshot shows a web application window titled 'Dir_Exp'. It is divided into two main sections: a table for viewing experiments and a form for editing experiment details.

Таблиця перегляду експериментів

Id_ex	Name_ex	Autor_ex	DatReg	Descr	Evaluation
1	Exp1	Гагарин Гайдар...	15.10.2018	Первый экспер...	Эксперимент 1 ...
2	Exp2	Сенченко	05.05.2016	Второй экспери...	Эксперимент 2 ...
3	Exp 3	Петров Коваль ...	23.11.2019	Exp 3 начальный	Эксперимент 3 ...
4	Exp 4	Варава	19.12.2018	Exp 4 проверка...	Эксперимент 4 ...
5	exp 5	Мельник	11.11.2019	exp 5 проверка ...	Эксперимент 5 ...

Облікова картка експерименту

Введіть або змініть дані про експеримент

Назва:

Автор:

Дата створення:

Опис:

Оценка:

Buttons: Фиксувати зміни у таблицю перегляду, Створити картку обліку експеримента, Вилучити картку обліку експеримента

Зберегти зміни у базу даних

Рисунок А13. - Екранна форма реєстрації експерименту

The screenshot shows a web application window titled 'Форма керування сценами експерименту'. It is divided into two main sections: a table for viewing scenes and a form for editing scene details.

Таблиця перегляду сцен

Id_sc	Id_exp	Id_activator	DataReg	Descr	Name
1	1	1	10.10.2016	Первая сцена п...	Сцена 1 ...
2	1	1	10.11.2016	Вторая сцена п...	Сцена 2 ...
7	1	1	20.11.2017	Контрольная сц...	Сцена 7 ...
1010	1	2	10.10.2016	Четвертая сц...	Сцена 8 ...
1012	1	3	10.01.2019	сцена 10 for exp 1	сцена 10 ...

Облікова картка сцени

Показати обрану сцену(або створити її, в разі додавання)

Введіть або змініть дані про сцени експерименту

Exp1

Назва сцени:

Акваторія:

Дата створення:

Опис:

Оценка:

Buttons: Фиксувати зміни у таблицю перегляду, Створити картку обліку сцени, Вилучити картку обліку сцени

Зберегти зміни у базу

Рисунок А14. – Екранна форма реєстрації сцени експерименту

Четвертим кроком є побудова сценарію експерименту який проводиться на обраній сцені. Побудова сценарію це найбільш складна частина аналітичного процесу, яка важко формалізується та яка передбачає вибір аналітичних інструментів, послідовність їх застосування і завдання параметрів, що визначають їх використання. Основним моментом при побудові сценарію

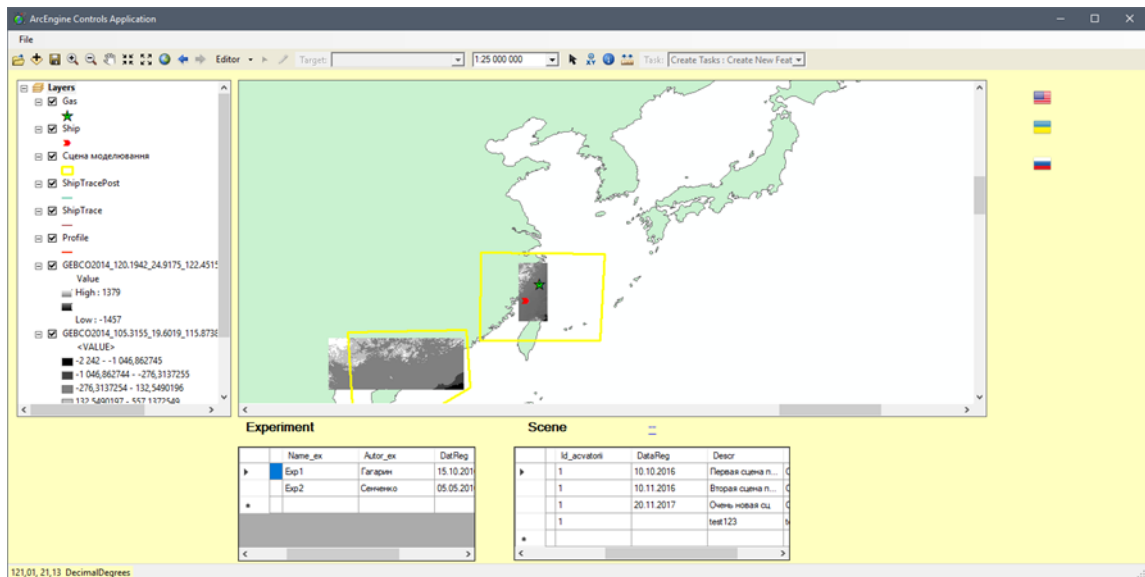


Рисунок А15. – Екранна форма побудови сцени експерименту є чітка фіксація мети використання сценарію і стану системи при його проведенні.

Сценарій моделювання складається по перше з визначення параметрів об'єктів розташованих на сцені на фоні координатної сітки з центром в місці розташування ВС (Рисунок А16).

У МК формування сценарію передбачає вибір певного набору методів (алгоритмів) обробки акустичних сигналів, а також набору методів (алгоритмів) їх класифікації та ідентифікації з метою розпізнавання об'єкта, що випромінює сигнал. Наявність різних алгоритмів обробки сигналів і об'єктів визначає безліч можливих сценаріїв. (Рисунок А17).

На цьому етапі за використання BPMN-діаграм здійснюється опис сценарію – побудова BPMN-моделі і її конвертація в OWL-модель сценарію та подальше автоматичне перетворення метаданих BPMN-моделі в виконавче програмне середовище моделювання сценарію.

У загальному випадку можливе використання різних інструментів для досягнення поставленої мети, що і є основною причиною використання сценарного підходу.

П'ятим кроком є виконання сценарію експерименту який проводиться на обраній сцені, з наданням користувачу форм з результатами виконання обраного

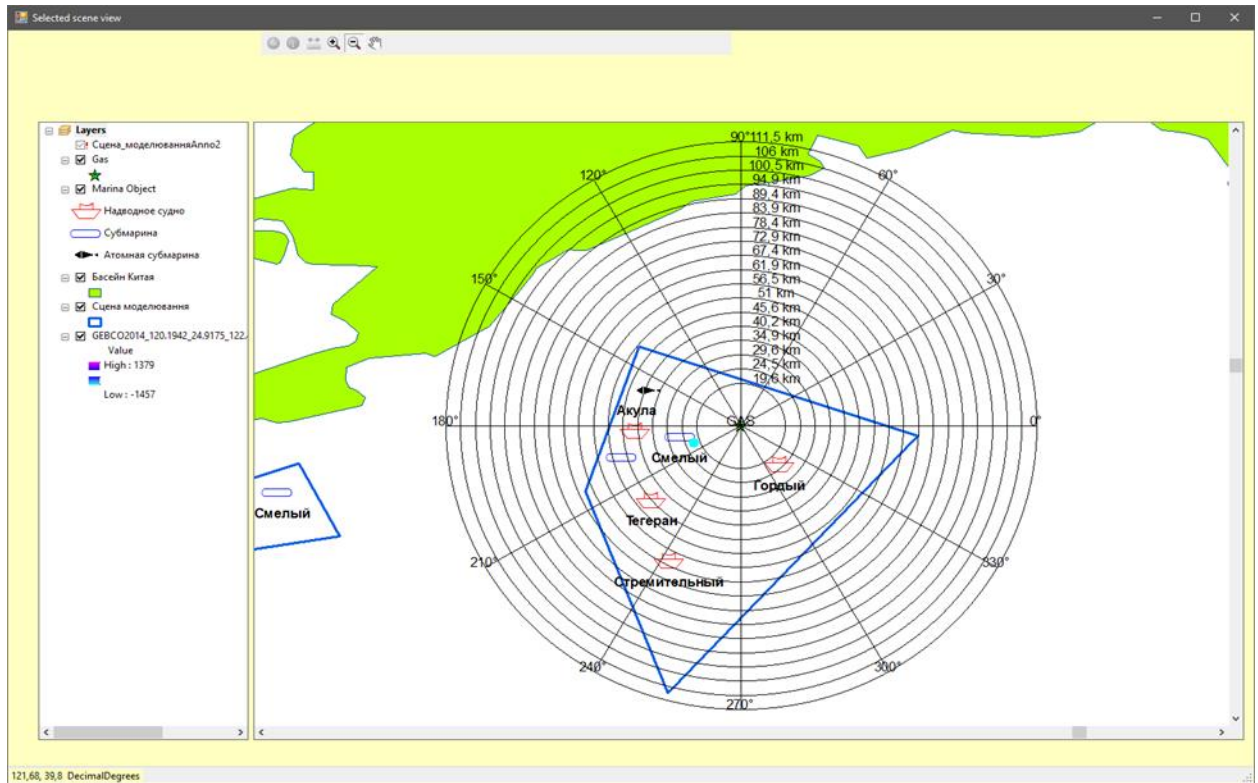


Рисунок А16. – Экранна форма визначенням розташування об'єктів на сцені

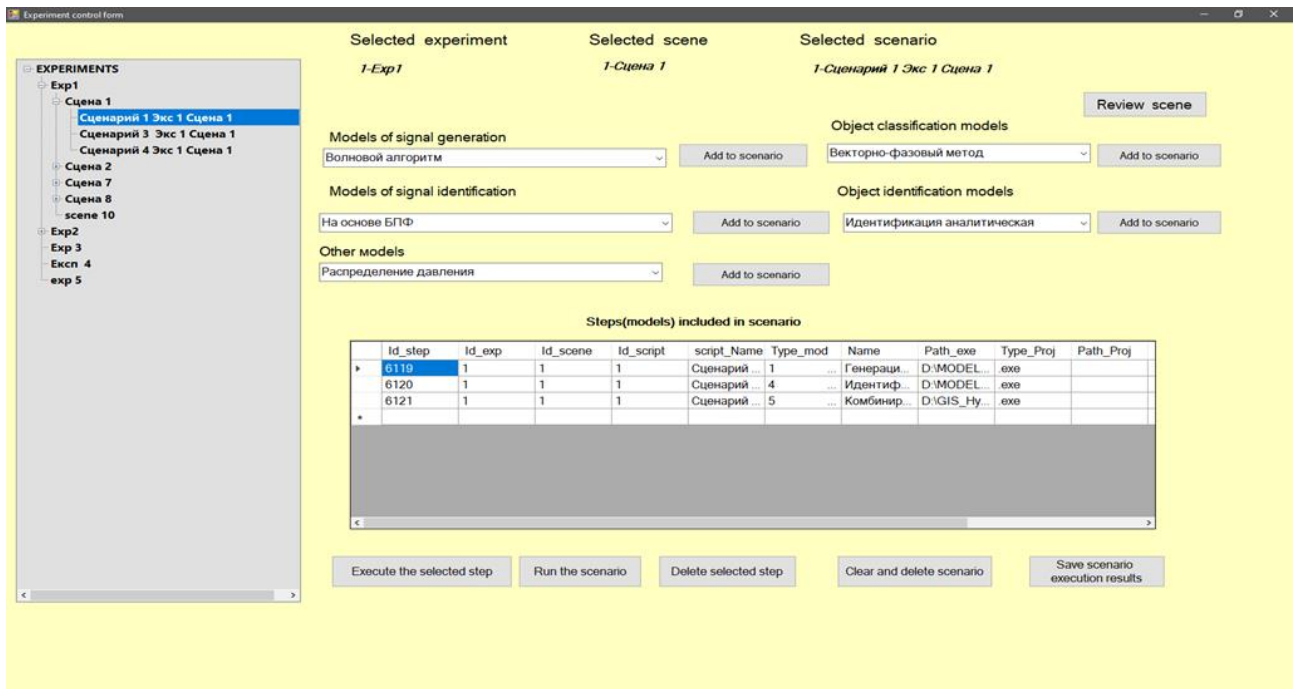


Рисунок А17. – Экранна форма визначення переліку обчислювальних моделей, задіяних у сценарії

кроку сценарію, або всього сценарію. На рисунку А18 наведено зразок форми з результатами моделі розповсюдження звукової хвилі за визначеними користувачем параметрами об'єкта та параметрами водного середовища.

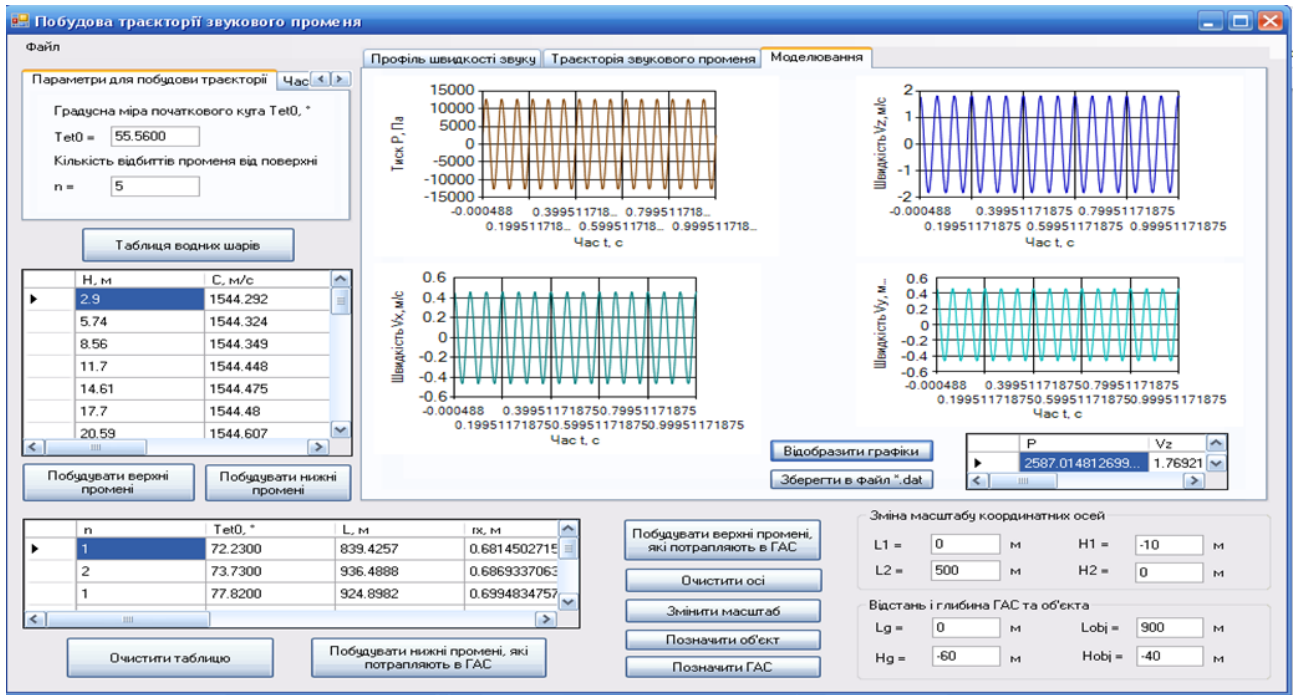


Рисунок А18. – Екранна форма результатів моделювання процесу розповсюдження звукової хвилі за променевим алгоритмом

Шостим кроком є оцінювання отриманих результатів окремого етапу експерименту. Оцінка може формуватися «вручну» дослідником, або відповідним програмним модулем в разі наявності формалізованого алгоритму оцінки і програми його реалізації (Рисунок А19).

Рисунок А19 – Екранна форма фіксації результатів окремих етапів експерименту

Сьомим кроком є формування підсумкової оцінки проведення експерименту, якій формується у вигляді протоколу оцінок по кроках з формуванням загального висновку (Рисунок А20):

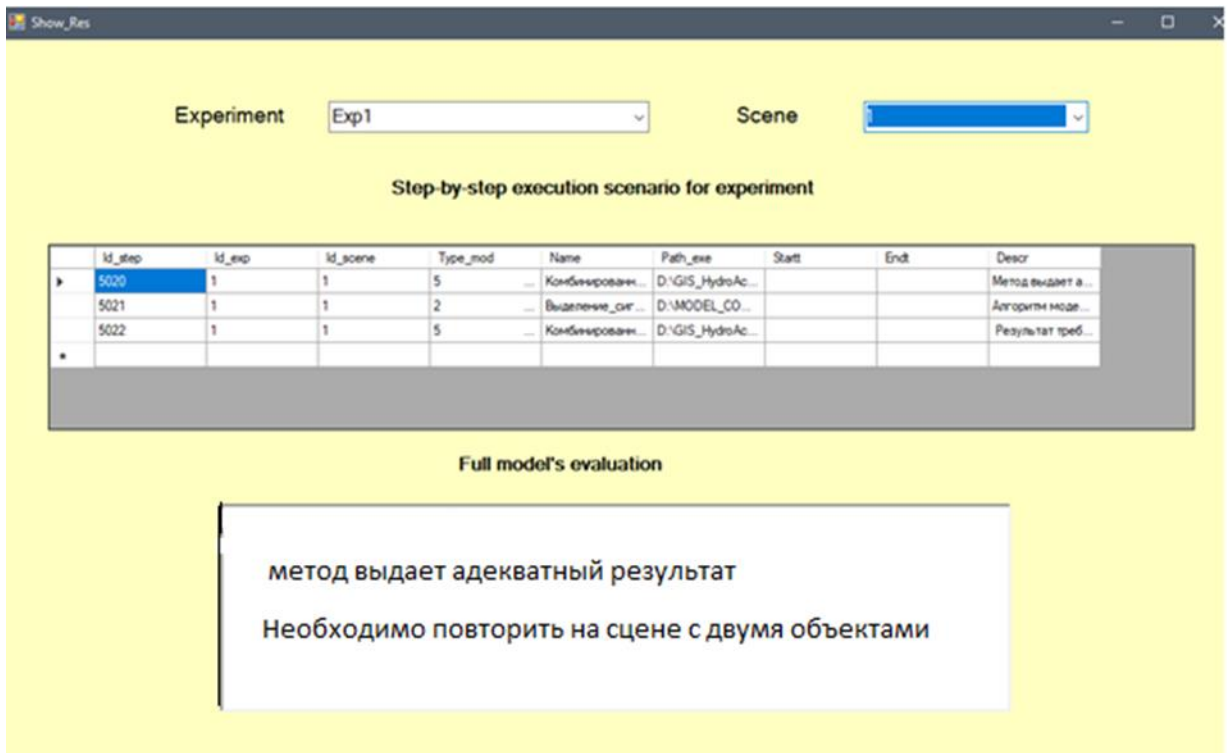


Рисунок А20 – Екранна форма загальної оцінки експерименту

А3. Фрагмент сценарію ідентифікації об'єктів на основі мови запитів SPARQL-Query

Крок 1. Формування запиту на виявлення невідомих сигналів

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?a ?b ?c

WHERE

{

 ?a ?b ?c

 Filter regex(str(?a),"UMO")

 Filter (regex(str(?b),"speed"))

```
# Filter (regex(str(?b), "rdf:type"))
```

```
}
```

Результат виконання запиту:

The screenshot shows a SPARQL query editor window with the following content:

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?a ?b ?c
  WHERE
{
  ?a ?b ?c
  Filter regex(str(?a), "UMO")
  Filter (regex(str(?b), "speed"))
  # Filter (regex(str(?b), "rdf:type"))
}
```

a	b	c
UMO1SIG03	Vibrational_speed_Vy	"19.3"^^<http://www.w3.org/2001
UMO1SIG03	Vibrational_speed_Vx	"18.3"^^<http://www.w3.org/2001
UMO1SIG01	Vibrational_speed_Vx	"18.0"^^<http://www.w3.org/2001
UMO1SIG02	Vibrational_speed_Vx	"18.2"^^<http://www.w3.org/2001
UMO1SIG02	Vibrational_speed_Vz	"22.2"^^<http://www.w3.org/2001
UMO1SIG01	Vibrational_speed_Vy	"19.0"^^<http://www.w3.org/2001
UMO1SIG03	Vibrational_speed_Vz	"22.2"^^<http://www.w3.org/2001
UMO1SIG01	Vibrational_speed_Vz	"22.0"^^<http://www.w3.org/2001
UMO1SIG02	Vibrational_speed_Vy	"19.2"^^<http://www.w3.org/2001

Execute

Synchronising

Крок 2. Вибираємо невпізнаний сигнал для подальшої ідентифікації по наступній умові:

18.0 => Vibrational_speed_Vx <= 20.0

Крок 3. Формуємо запит на пошук відповідного цій умові шаблону

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

```
PREFIX rdfs:
```

```
SELECT ?a ?b ?c
```

```
WHERE
```

```

{
    ?a ?b ?c
    Filter regex( str(?a), "MTTS")
    Filter (regex(str(?b),"Vibrational_speed_Vx" ) )
    Filter (?c<=20.0)
    Filter (?c>=18.0)
}

```

The screenshot shows a SPARQL query editor window with the following content:

```

SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?a ?b ?c
  WHERE
  {
    ?a ?b ?c
    Filter regex( str(?a), "MTTS")
    Filter (regex(str(?b),"Vibrational_speed_Vx" ) |
    Filter (?c<=20.0)
    Filter (?c>=18.0)
  }

```

a	b	c
MTTS03	Vibrational_speed_Vx	"18.3"^^<http://www.w3
MTTS01	Vibrational_speed_Vx	"18.0"^^<http://www.w3
MTTS02	Vibrational_speed_Vx	"18.2"^^<http://www.w3

Execute

Synchronising

Даному запиту відповідають три шаблону. Уточнюючи умови вибору $\text{Vibrational_speed_Vx} = 18.0$

визначаємо відповідний шаблон, а саме шаблон MTTS01.

Крок 4. Візуалізуємо характеристики (властивості) визначився шаблону MTTS01.

Загальний вигляд результату виконання сценарію:

The screenshot displays the Protege OWL editor interface. The main window shows the results of a SPARQL query. The query is as follows:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?a ?b ?c
WHERE
{
  ?a ?b ?c
  Filter regex(str(?a), "UMO")
  Filter (regex(str(?b), "speed"))
}

```

The results table shows the following data:

a	b	c
UMO1S1G03	Vibrational_speed_Vy	"19.3"^^<http://www.w3.c
UMO1S1G03	Vibrational_speed_Vx	"18.3"^^<http://www.w3.c
UMO1S1G01	Vibrational_speed_Vx	"18.0"^^<http://www.w3.c
UMO1S1G02	Vibrational_speed_Vx	"18.2"^^<http://www.w3.c
UMO1S1G02	Vibrational_speed_Vz	"22.2"^^<http://www.w3.c
UMO1S1G01	Vibrational_speed_Vy	"19.0"^^<http://www.w3.c
UMO1S1G03	Vibrational_speed_Vz	"22.2"^^<http://www.w3.c
UMO1S1G01	Vibrational_speed_Vz	"22.0"^^<http://www.w3.c
UMO1S1G02	Vibrational_speed_Vy	"19.2"^^<http://www.w3.c

A4. Фрагмент програмного застосунку побудови сценарію

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Data.Sql;
using System.Data.SqlTypes;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Diagnostics;
using System.Threading;

namespace HidroAcModComp
{
    public partial class Script_scene : Form
    {
        static public int IdExp, IdScen, IdScript, IdStep;
        static public string ExpName, ScenName, ScriptName;
        static public TreeNode selectedNode = new TreeNode();
        public Process currProc;
        public Form mainf;
        public string lang;
        public DataTable dt;
        public int currentId_GAS;
        public int Currtreelev = 1;
        public string priznakmenu;
        SqlConnection cn;
        SqlCommand cmd;
        public DataTable dt1, dt2, dt3, dt4, dt5, dt6, dt7, dtMod, dtModd, dt8;
    }
}

```

```

SqlDataReader dr;
int IdModGen, IdModVyd, IdModClass, IdModIdent, IdModComb;

public string selexp, selScene;

public Script_scene(Form fmain)
{
    mainf = fmain;
    InitializeComponent();
}

private void Script_scene_Load(object sender, EventArgs e)
{
    lang = Main_Form.language;
    // MessageBox.Show("lang in Script_scene = " + lang);
    Jastlabel.Text = "";
    switch (lang)
    {
        case "en-US":
            Thread.CurrentThread.CurrentUICulture = new
System.Globalization.CultureInfo("en-US");
            Controls.Clear();
            InitializeComponent();

            break;
        case "uk-UK":
            Thread.CurrentThread.CurrentUICulture = new
System.Globalization.CultureInfo("uk-UK");
            Controls.Clear();
            InitializeComponent();
            break;
        case "ru-RU":
            Thread.CurrentThread.CurrentUICulture = new
System.Globalization.CultureInfo("ru-RU");
            Controls.Clear();
            InitializeComponent();
            break;
    }
    cn = new SqlConnection();
    string cnstr = Properties.Settings.Default.ModCompDBConStr;
    cn.ConnectionString = cnstr;
    cn.Open();
    cmd = cn.CreateCommand();
    // // заполнение дерева экспериментами
    cmd.CommandText = "SELECT * FROM Experiment";
    dr = cmd.ExecuteReader();

    dt6 = new DataTable();
    dt6.Load(dr);

    this.treeView1.Nodes.Clear(); // Очищаем дерево
    switch (lang) // заносим корень дерева
    {
        case "en-US": this.treeView1.Nodes.Insert(0, "0", "EXPERIMENTS");
            break;
        case "ru-RU": this.treeView1.Nodes.Insert(0, "0", "ЭКСПЕРИМЕНТЫ");
            break;
        case "uk-UK": this.treeView1.Nodes.Insert(0, "0", "ЕКСПЕРИМЕНТИ");
            break;
    }
}

```

```

    Currtreelev = 1;
    TreeNode[] Snode = this.treeView1.Nodes.Find("0", false);
    // this.treeView1.Font.Size = 10;
    LoadSubNodes(Snode[0], dt6); // Вызываем функцию добавления узлов

// модели генерации
    cmd.CommandText = "SELECT * FROM model where TypeID = 1"; // =
    'Генерация_сигнала';
    dr = cmd.ExecuteReader();

    dt1 = new DataTable();
    dt1.Load(dr);
    // this.dataGridView1.DataSource = this.dt1;

    this.comboBox1.DataSource = this.dt1;
    this.comboBox1.DisplayMember = "Name";
    this.comboBox1.ValueMember = "Name";
// модели выделения
    cmd.CommandText = "SELECT * FROM model where TypeID =2 "; // =
    'Выделение_сигнала';
    dr = cmd.ExecuteReader();
    dt2 = new DataTable();
    dt2.Load(dr);

    this.comboBox2.DataSource = this.dt2;
    this.comboBox2.DisplayMember = "Name";
    this.comboBox2.ValueMember = "Name";
// модели классификации
    cmd.CommandText = "SELECT * FROM model where TypeID = 3 "; // =
    'Классификация_объекта';
    dr = cmd.ExecuteReader();
    dt3 = new DataTable();
    dt3.Load(dr);

    this.comboBox3.DataSource = this.dt3;
    this.comboBox3.DisplayMember = "Name";
    this.comboBox3.ValueMember = "Name";
// модели идентификации
    cmd.CommandText = "SELECT * FROM model where TypeId = 4 "; // =
    'Идентификация_объекта';
    dr = cmd.ExecuteReader();
    dt4 = new DataTable();
    dt4.Load(dr);

    this.comboBox4.DataSource = this.dt4;
    this.comboBox4.DisplayMember = "Name";
    this.comboBox4.ValueMember = "Name";
// комбинированные модели
    cmd.CommandText = "SELECT * FROM model where TypeId =5 "; // =
    'Комбинированная';
    dr = cmd.ExecuteReader();
    dt5 = new DataTable();
    dt5.Load(dr);

    this.comboBox5.DataSource = this.dt5;
    this.comboBox5.DisplayMember = "Name";
    this.comboBox5.ValueMember = "Name";
}

```

ДОДАТОК Б. Комп'ютерне моделювання сценаріїв розподілу енергопостачання

Сучасні енергетичні системи продовжують еволюціонувати. Децентралізація стає характерною рисою нової моделі енергопостачання. Це веде до подальшого збільшення частки тарифів із значною диференціацією ціни на електроенергію. Споживач вже не в змозі управляти енергетичною інфраструктурою самостійно без додаткових інструментів, оскільки йому буде необхідно враховувати декілька аспектів, серед яких мінімізація витрат, рівень виникаючих при зміні режиму електроспоживання незручностей, експлуатаційні проблеми, особливості електроприладів, баланс електронавантаження. Тому постає задача забезпечити споживача таким інструментом, завданням якого буде, знаючи прагнення або необхідність у використанні тих чи інших електроприладів, терміни навантаження, створити сценарій функціонування всіх елементів енергетичної інфраструктури з метою мінімізації фінансових витрат на енергію.

На сьогодні на енергетичному ринку пропонуються підходи до зміни структури споживання електроенергії споживачами з метою зменшення фінансових витрат, пікового попиту на електроенергію, зокрема за використанням програмних систем. Основними недоліками розроблених систем є проблема врахування рівня незручностей користувача. Розроблюваний інструмент повинен надавати користувачу можливість налаштування рівня виникаючих незручностей, аби користувач міг їх оцінити та обрати оптимальний режим енерговикористання.

Прийняття рішень про розвиток енергетичної інфраструктури починається з аналізу прогнозованого попиту на енергію. Враховуючи співпадіння у часі споживання та виробництва енергії, важливо знати не тільки обсяги попиту за певний проміжок часу, а й характер попиту. Для характеристики попиту на енергію використовують графіки навантаження.

Графік навантаження енергосистеми є сумою графіків навантаження окремих груп споживачів, кожна з яких має свій графік навантаження.

Для прикладу, розглянемо режими споживання електроенергії двома споживачами (Рисунок Б1). Очевидно, що при однаковому добовому споживанні електричної енергії витрати енергосистеми на покриття попиту будуть неоднаковими (капіталовкладення на створення необхідної генеруючої інфраструктури будуть суттєво різними). Крім того, необхідно врахувати наявність тарифних зон споживання електричної енергії.

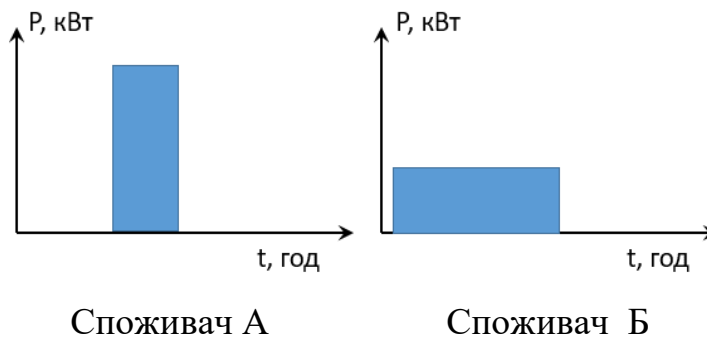


Рисунок Б1 – Графіки режимів споживання електроенергії двома споживачами

Від режиму споживання залежить структура потужності електростанцій енергосистеми, необхідна пропускна здатність ліній електропередачі. Крім того, від режиму споживання залежать не лише затрати на розвиток енергосистеми, а й затрати на експлуатацію діючого обладнання.

Структура потужностей енергетичних станцій визначає стратегію розміщення капіталовкладень, а також і стратегію фінансування розвитку енергосистеми.

Дуже важливо точно визначити об'єми та режими споживання енергії:

- зниження потреби веде до фінансового збитку, оскільки доводиться покривати дефіцит енергії за рахунок закупівлі у інших постачальників або обмежувати споживачів;

- завищення потреби також веде до фінансового збитку за рахунок омертвіння капіталу у додатково встановленому обладнанні, що не повністю використовується, або за рахунок придбаної надлишкової кількості палива.

Складовою частиною графіка навантаження енергосистеми крім попиту споживачів є втрати енергії при її передачі у розподільчих мережах, а також витрати на власні потреби електростанцій систем.

Отже, режими споживання та точність їх визначення суттєво впливають на результати діяльності енергетичного підприємства. Звідси і випливає необхідність вивчення режимів та графіків навантаження електростанцій та інших елементів енергетичної інфраструктури. Одним із способів побудови добового графіка навантаження є реєстрація обсягів споживання енергії - показники електролічильника записують через рівні інтервали часу (зазвичай, через 0,5 години) і для кожного інтервалу визначається середнє навантаження.

Б1. Концепція управління попитом на електроенергію з боку користувача

Управління попитом на електроенергію з боку користувача - зміна споживання електроенергії кінцевими споживачами щодо їх нормального профілю навантаження у відповідь на зміну цін на електроенергію в часі або у відповідь на стимулюючі виплати, передбачені щоб знизити споживання в періоди високих цін на електроенергію на оптовому ринку або коли системна надійність під загрозою. Ринки електроенергії проектуються таким чином, щоб стимулювати учасників підтримувати баланс виробництва та споживання електроенергії. Використання цифрових інтервальних лічильників електроенергії, «інтелектуальних» мереж (Smart Grid, SG) дає можливість підвищення еластичності споживання шляхом цілеспрямованого впливу на обладнання споживача коли це необхідно. Сутність концепції управління попитом з боку користувача (Demand Side Management, DSM) у цілеспрямованому і систематичному впливі енергетичного підприємства на обсяги, структуру і режими енергоспоживання.

Насьогодні більшість споживачів електроенергії масового ринку використовують лічильники з одним регістром даних, який просто накопичує споживання з плином часу. Ці споживачі можуть виставляти рахунки тільки за

електроенергію, яку вони використовують у відповідності з тарифами, які не відображають зміну в часі (наприклад, від години до години, день у день, від сезону до сезону) вартості виробництва електроенергії.

Концепція DSM дає можливість використовувати різні види тарифів, що засновані на часі доби, тобто загальним для них є зміна ціни із часом, різні ціни діють для різних годин в різні дні. Реалізація концепції DSM здійснюється за використанням інтелектуальних програмно-апаратних систем, наприклад «Інтелектуальна будівля», яка забезпечує:

- злагоджену роботу інженерних систем у будинку (систем життєзабезпечення, таких як: освітлення, опалення, вентиляція, кондиціонування, водопостачання, каналізації, і т. д.),
- контроль чинників, що впливають на необхідність включення або відключення вище зазначених систем.

Сутність «Інтелектуальної будівлі» в автоматизації задання і відслідковування режимів роботи інженерних систем і електроприладів згідно з зовнішніми і внутрішніми умовами. Основними цілями створення системи «інтелектуальної будівлі» є:

- забезпечення ефективності енергоспоживання (в тому числі за рахунок участі в управлінні попитом на електроенергію),
- поліпшення комфорту,
- підвищення безпеки.

Одною з складових системи автоматизованого управління інженерним обладнанням будівель інтелектуальної програмно-апаратної системи, що реалізує концепцію DSM, є програмний додаток, який може ефективно планувати потреби споживачів в електроенергії, наприклад для одної будівлі. Він повинен розв'язувати наступні задачі:

1. організація внесення, корекції, зберігання даних про характеристики приладу електроспоживання та тарифи на електроенергію;

2. планування електроспоживання приладів, забезпечуючи мінімізацію фінансових витрат при збереженні комфорту енерговикористання;

3. налаштування користувачем можливості вибору з поміж двох пріоритетів планування вартості електроспоживання або комфорту енерговикористання;

4. генерація звіту про результати планування електроспоживання, що має включати наступні дані: вартість електроспоживання загальна (обраного приладу) за добу (за обраний проміжок часу протягом доби), дискомфорт енерговикористання загальний (обраного приладу) за добу (за обраний проміжок часу протягом доби), пікова споживана потужність загальна.

Б2. Математична модель електроспоживання

З метою розробки алгоритму розрахунку споживчої потужності та вартості електроспоживання за одиницю часу (за годину, за добу) для різних електроприладів - гнучких і негнучких, якій може бути задіяний для побудови типового сценарію роботи користувача при моделюванні процесів споживання електричної енергії, була побудована математична модель електроспоживання різними приладами.

Загальна споживана потужність за годину для всіх приладів:

$$PC_{t \in T} = \sum_{i=1}^n P_i^t \times X_i^t$$

$$X_i^t = \begin{cases} 1 \\ 0 \end{cases} \quad \forall t_s \leq t \leq t_f$$

де P_i^t - споживана потужність приладу i у момент часу t ,

X_i^t - змінна рішення, яка представляє стан приладу i у момент часу t

$X_i^t = 1$ - прилад увімкнений, $X_i^t = 0$ - прилад вимкнений

Загальна споживана потужність за добу для всіх приладів:

$$PC_{total} = \sum_{i=1}^n \sum_{t=1}^{24} P_i^t \times X_i^t$$

$$X_i^t = \begin{cases} 1 \\ 0 \end{cases} \quad \forall t_s \leq t \leq t_f$$

Загальна вартість електроспоживання за добу залежить від двох чинників:

- споживана потужність приладу,
- ціна на електроенергію у цей період часу.

Загальна вартість електроспоживання за добу:

$$Cost_{total} = \sum_{i=1}^n \sum_{t=1}^{24} P_i^t \times X_i^t \times EP_t$$

$$X_i^t = \begin{cases} 1 \\ 0 \end{cases} \quad \forall t_s \leq t \leq t_f$$

де EP_t - ціна на електроенергію у період часу роботи приладу.

С метою вирішення задачі оптимізації - мінімізації витрат на електроенергію при плануванні користувачем розподілу енергопостачання та мінімальній незручності для користувача був використаний генетичний алгоритм, призначений для оптимізації багатопараметричних функцій (Рисунок Б.1).

Б3. Побудова типового сценарію розподілу енергопостачання

Слід враховувати, що певні процеси споживання електричної енергії характеризуються:

- ручним вмиканням та вимиканням;
- ручним вмиканням та автоматичним вимиканням;
- автоматичним вмиканням та автоматичним вимиканням;
- ручним регулюванням параметрів споживання;
- автоматичним регулюванням параметрів споживання.

Для спрощення можна вважати, що є лише один користувач- аналітик, що охоплює поведінку всіх учасників процесу (мешканців квартири або будинку).

Тоді кроки сценарію будуть наступні:

1. Складається перелік приладів-споживачів електричної енергії і згідно переліку формується база даних та заповнюється база знань з можливістю редагування з полями «назва приладу», «електрична потужність», « $\cos \phi$ », ...

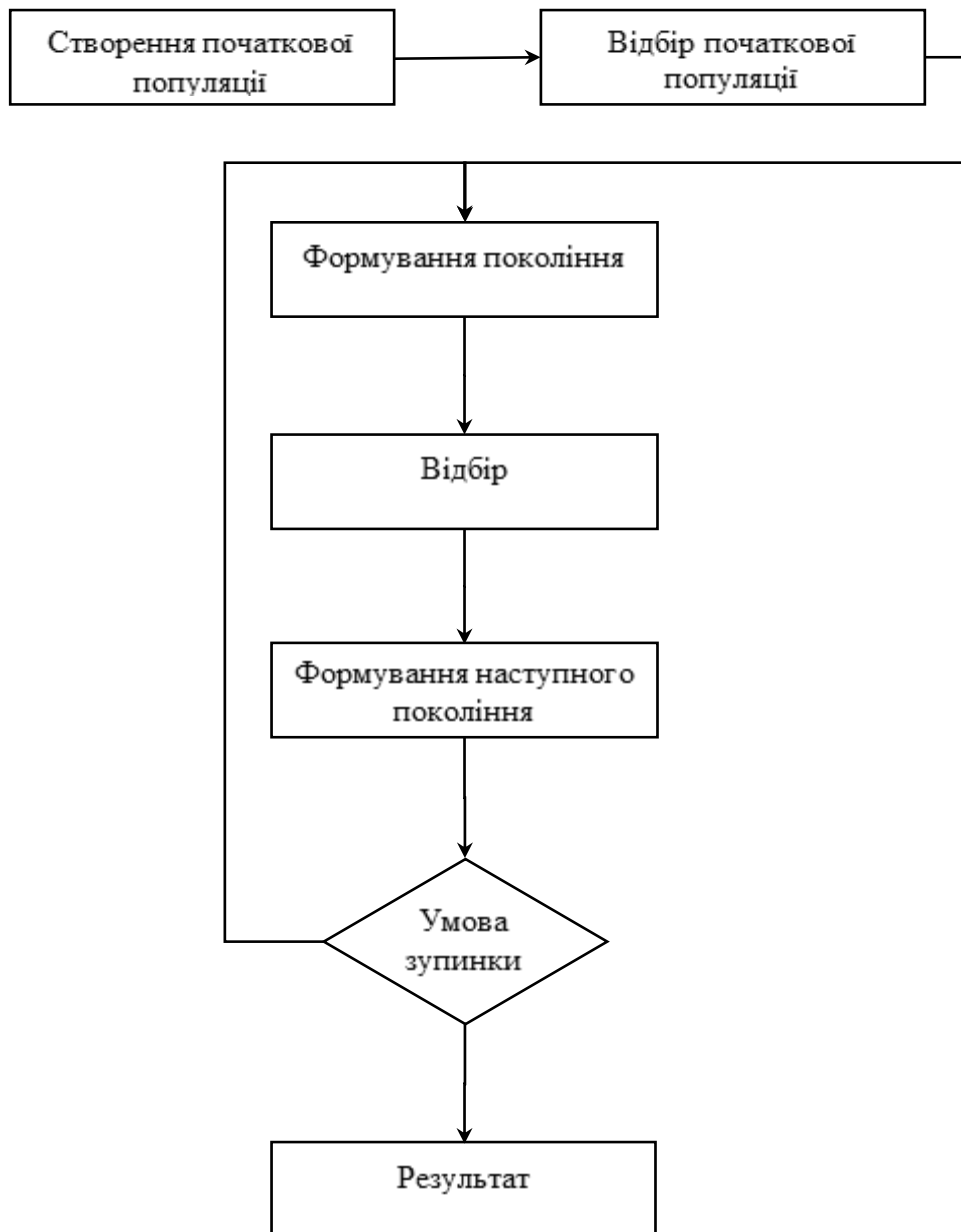


Рисунок Б.1. – Загальна схема генетичного алгоритму

2. За використанням програмного додатку забезпечується введення та редагування індивідуальних графіків активності використання кожного приладу-

споживача з наявної бази даних протягом одного тижня, що дозволяє врахувати особливості ділової активності абонентів-споживачів для кожної доби тижня.

Складається індивідуальний графік електричного навантаження кожного окремого споживача. Реєструється час ввімкнення та вимкнення споживача (або тривалість його роботи) для кожного сеансу активності (споживач може використовуватись кілька разів протягом окремої доби).

3. Передбачається можливість рендомізації часу включення приладу, тривалості роботи до моменту вимикання, величини навантаження (якщо такий режим експлуатації приладу передбачається його конструкцією).

Активність використання приладу задається різними способами (відповідно його конструктивним особливостям), наприклад:

- ручним вмиканням та вимиканням (освітлення, пиросос, ...);
- ручним вмиканням та автоматичним вимиканням (електричний чайник, пральна машинка, ...);
- автоматичним вмиканням та автоматичним вимиканням (електричний бойлер, холодильник, ...);
- ручним регулюванням параметрів споживання (електричний нагрівач, електричний бойлер, ...);
- автоматичним регулюванням потужності споживання (інверторний кондиціонер/тепловий насос, інверторний холодильник, ...);
- автоматичним регулюванням тривалості сеансу активності (тривалість сеансу активності електричного чайника визначається об'ємом залитої води та її початковою температурою; тривалість сеансу активності холодильної камери холодильника визначається її завантаженістю продуктами, їх початковою (на момент включення) температурою, кількістю та тривалістю відкриття дверей камери; ...).

4. По закінченню введення вхідних даних зніціюється розрахунок.

5. Етап 1 – Розрахунок характеристик базового графіка електричного навантаження.

Програмний додаток розраховує та відображає:

- 1) індивідуальні графіки електроспоживання (графік електричного навантаження, ГЕН) для кожного приладу-споживача окремо протягом тижня;
- 2) сумарний графік споживання електричної енергії $P_{\text{слож}} = f(t)$ () всіма приладами-споживачами протягом тижня;
- 3) визначає та відображає максимальні пікові значення потужності споживання (кВт) з сумарного графіка електричного навантаження для кожної доби тижня;
- 4) обчислює та відображає обсяги споживання електричної енергії (кВт/год) з сумарного графіка електричного навантаження для кожної доби тижня;
- 5) визначає та відображає інформацію про найбільш навантажену добу за критерієм обсягів добового споживання електричної енергії;
- 6) обчислює та відображає обсяги витрат коштів (грн) на електричну енергію та місяць при споживанні електричної енергії за однозонним, двозонним та тризонним тарифними планами. (Попередньо створюється програмний інструмент формування/редагування добових тарифних планів споживання електричної енергії);
- 7) формує рекомендацію щодо тарифного плану, який доцільно використовувати;
- 8) складає перелік приладів-споживачів електричної енергії:
 - які формують пікові навантаження (дуже важливо для умов автономного енергозабезпечення без резервування від централізованої електромережі).

Візуалізація даних передбачається з відображенням у таблиці з полями: доба; час включення; час виключення; тривалість роботи у межах відрізка пікового споживання; обсяги енергії спожитої у межах пікового відрізка; частка споживання енергії у межах пікового відрізка.

– які споживають електричну енергію у межах тарифної зони з підвищеною ціною на енергоресурс.

Візуалізація даних передбачається з відображенням у таблиці з полями: доба; час включення; час виключення; тривалість роботи у межах відрізка підвищеного тарифу; обсяги енергії спожитої у межах підвищеного тарифу; частка споживання енергії у межах підвищеного тарифу.

6. Етап 2 – Оптимізація базового графіка електричного навантаження.

Програмний додаток:

– забезпечує можливість редагування індивідуальних графіків навантаження приладів-споживачів протягом тижня;

– здійснює оптимізацію мінімізації витрат на електроенергію при плануванні користувачем розподілу енергопостачання та мінімальній незручності для користувача;

– перераховує показники нового (оптимізованого) сумарного ГЕН та виводить звіт.

Дії користувача-аналітика: вносить зміни у індивідуальні ГЕН приладів-споживачів з метою оптимізації сумарного ГЕН об'єкта за критерієм мінімізації пікового навантаження (для автономних систем) або за критерієм мінімізації фінансових витрат на електричну енергію (для умов централізованого електрозабезпечення з тарифними зонами).

На рисунку Б.2 надана структура процесу комп'ютерного моделювання сценарію розподілу енергопостачання та на рисунку Б.3 надана діаграма сценарію у нотації BPMN. Очевидно, з діаграми видно, що деякі підпроцеси самі є складними процесами, які мають бути теж деталізовані.

Крім того, для виконання деяких завдань необхідно мати доволі велику кількість інформації з ПрО, яку найбільш доцільно формувати у вигляді онтології. Так, наприклад, для коректної побудови графіку споживання електроенергії для кожного споживача необхідно мати достатньо повну інформацію про всі побутові прилади, що використовує кожен з споживачів, а

також додатково їхні аналоги, які могли б бути застосовані для зниження витрат на електроенергію. Як видно з структури сценарію (рисунок Б.2) побудова сценарію базується на онтології Про задачі, що вирішується.

Приклад фрагменту такої онтології приведений на рисунку Б.4.

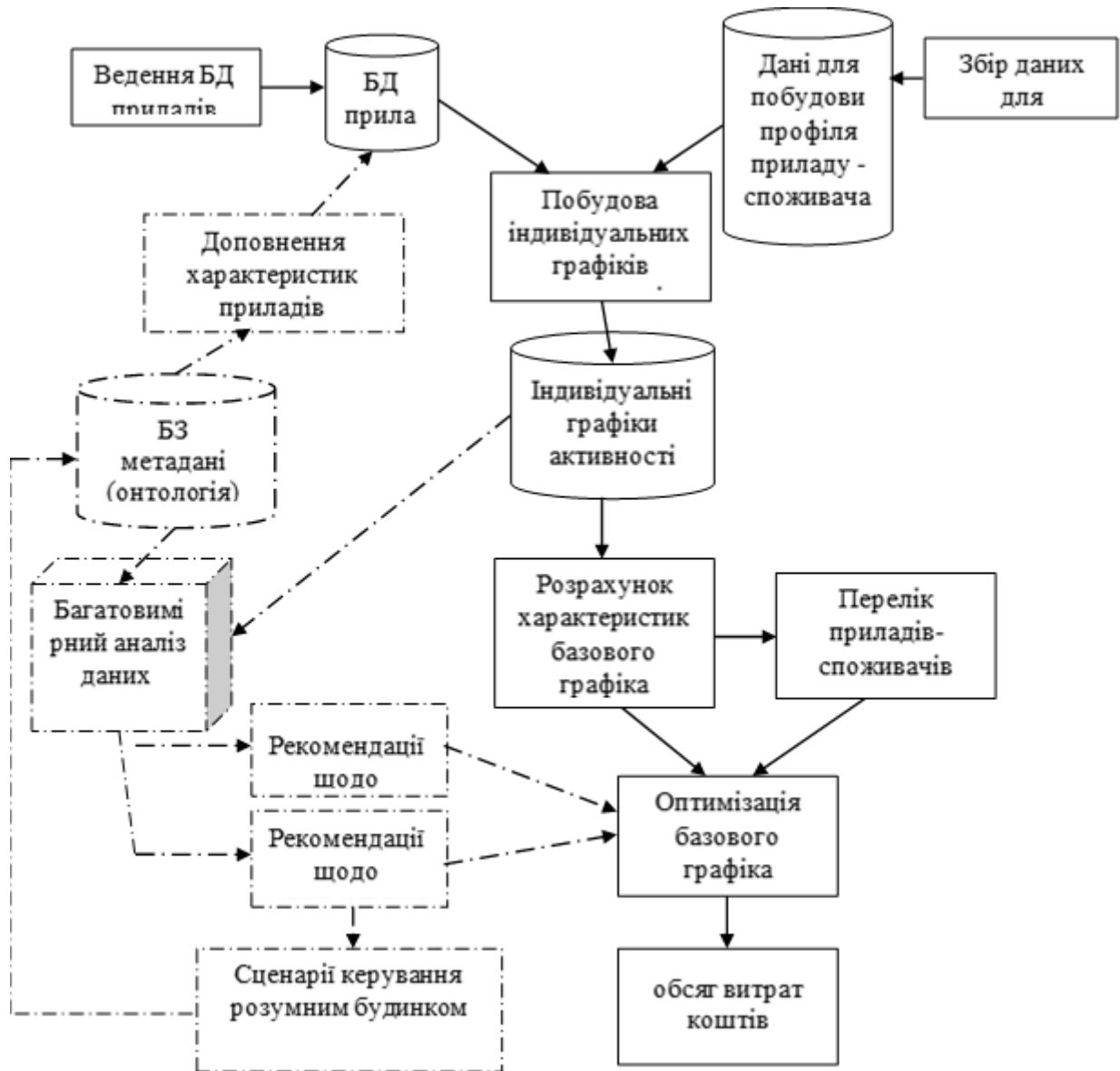


Рисунок Б.2. – Структура процесу комп'ютерного моделювання сценарію розподілу енергопостачання

Вирішення задачі конвертації BPMN-моделі (рисунок Б3) в OWL-модель (рисунок Б4) здійснювалося за використанням розробленого в дисертаційній роботі метода та бібліотеки Python owlready2 та spiffworkflow.

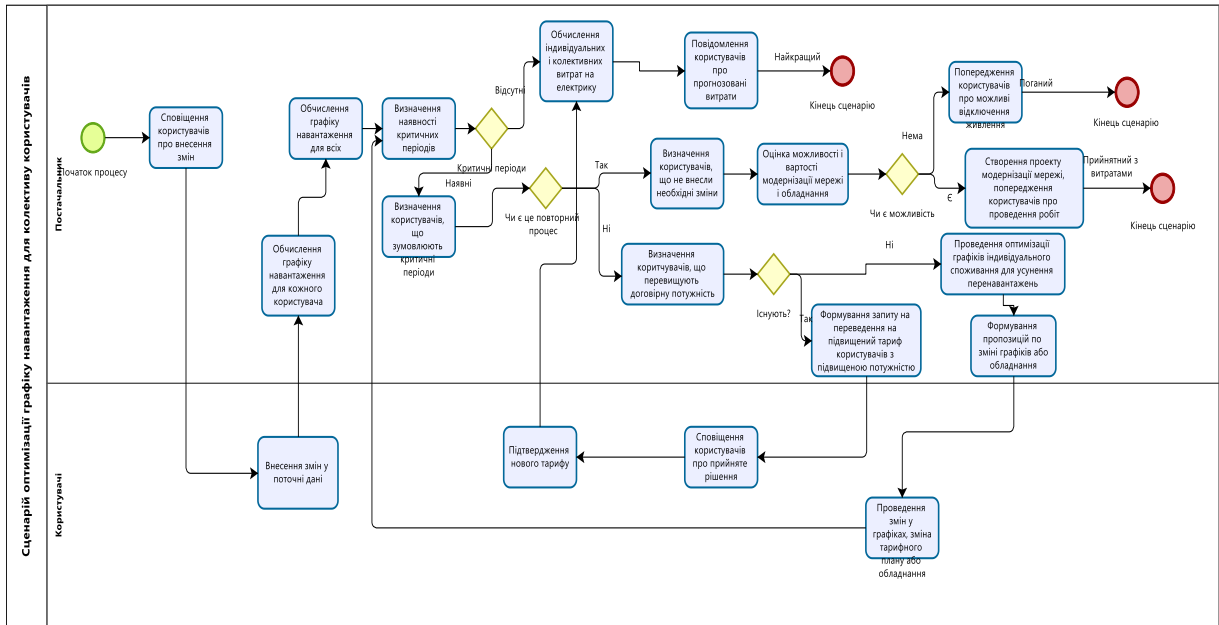


Рисунок Б.3. – Діаграма сценарію у нотації BPMN

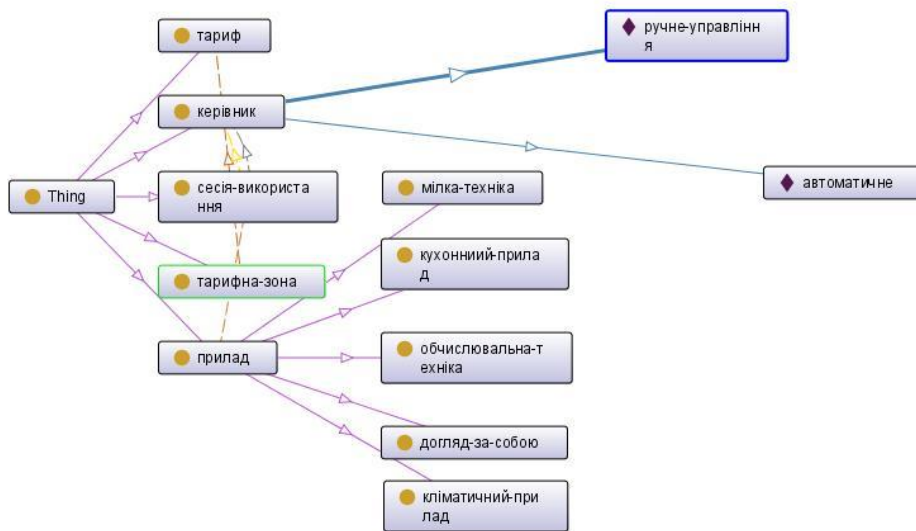


Рисунок Б.4. – Приклад фрагменту верхнього рівня онтології, що включає в себе перелік приладів та опис тарифів, що застосовуються
 На рисунку Б.5 представлена

Б4. Отримані результати планування енергоспоживання за використанням сценарію

Комп'ютерне моделювання сценарію розподілу енергопостачання відбувалося відповідно до наступних видів тарифів на електроенергію для населення:

- однозонного тарифу (за обсяг, спожитий до 100 кВт/год електроенергії на місяць),
- двозонного тарифу,
- трьохзонного тарифу.

У дослідження включені складені шестизонний і двадцятичотирьохзонний тарифи. Тарифи на електроенергію наведені у таблиці Б1 та графічно відображені на рисунку Б.5.

Програмний додаток задає режим роботи для споживачів енергії, який коливається між двома крайніми положеннями:

- «жорсткий» режим (мінімальний рівень свободи) – максимальне скорочення фінансових витрат електроспоживання,
- «гнучкий» режим (максимальний рівень свободи) – мінімальне скорочення незручностей, які спричиняє «жорсткий» режим зміною режимів споживання відповідно до тарифних зон.

Таблиця Б1. Тарифи на електроенергію, в копійках, за 1 кВт/год

Кіл-сть тариф. зон	Час доби																								
	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	0	1	2	3	4	5	6	
	Ціна, коп																								
1	90																								
2	45	90												45											
3	90	135			90						135		90		36										
6	90	150	135	90			65		90		150	135	90	36	20						36				
24	60	155	155	135	90	85	80	75	70	60	65	80	95	155	155	80	36	30	10	10	10	10	10	10	30

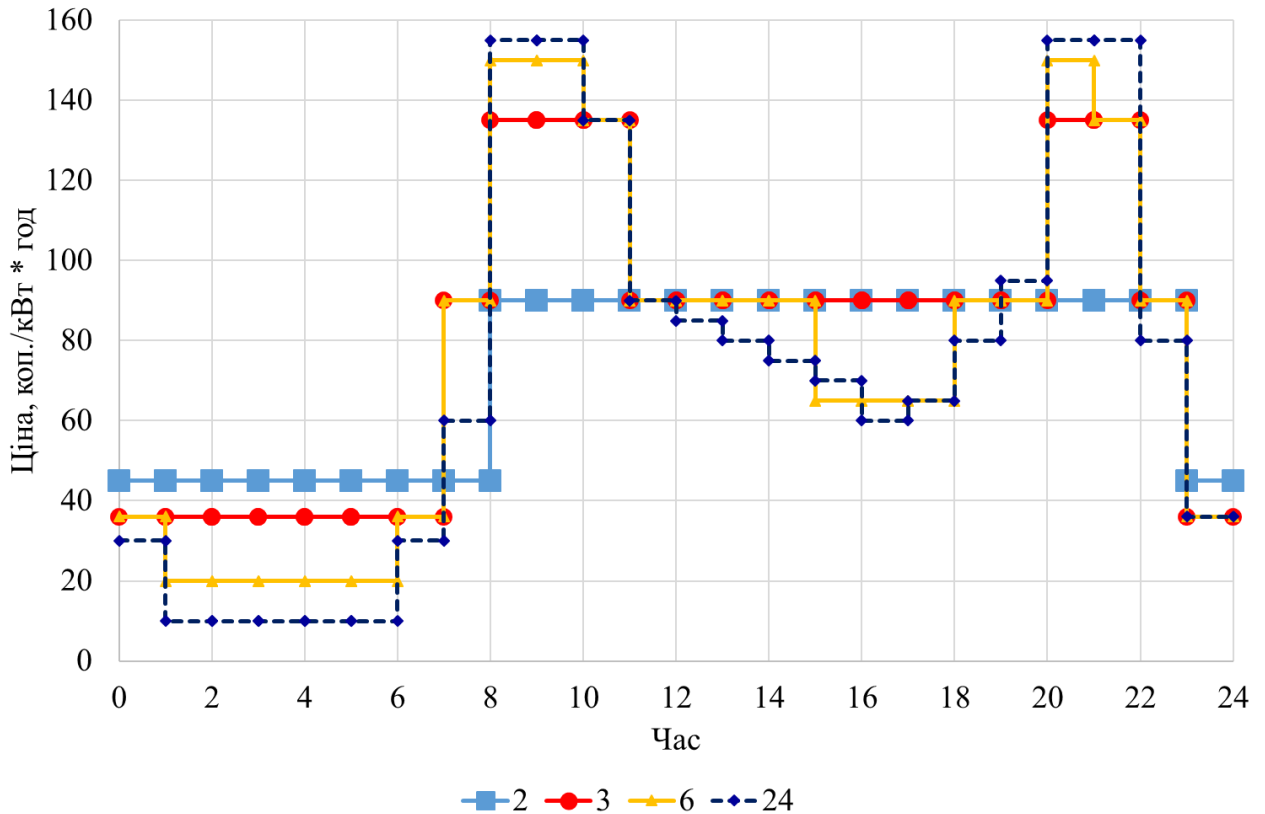


Рисунок Б.5. – Тарифи на електроенергію для різних тарифних зон

Горизонт планування складає добу. Рисунок Б.6 ілюструє, що інтервали часу з 7.00 до 11.00 і з 20.00 до 22.00 є піковими, напроти, інтервали часу з 1.00 до 6.00 є непіковими. Тобто, для досягнення скорочення фінансових витрат енергоспоживання повинно бути

- перенесено з пікових інтервалів на непікові (для гнучких приладів за часом - ГЧ),
- мінімізовано у пікові інтервали (для гнучких приладів за потужністю - ГП).

Причинами, що викликають дискомфорт користувача є

- затримка в роботі ГЧ приладів,
- зниження потужності ГП приладів.

Результати планування електроспоживання для максимального, середнього та мінімального рівня свободи користувача згідно з одно- (рисунки Б.6-Б.7), двох- (рисунки Б.8-Б.9), трьох- (рисунок Б.10), шести- (рисунок Б.11) і двадцятичотирьохзонними (рисунок Б.12) тарифами представлені у таблиці Б.2.

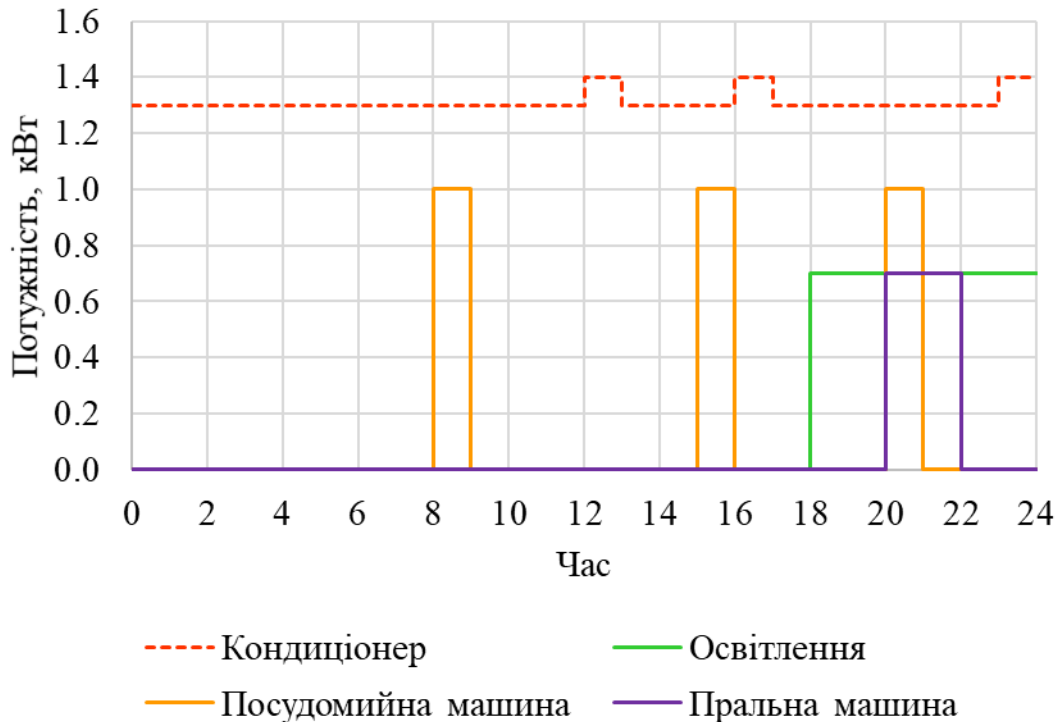


Рисунок Б.6. – Планування електронавантаження ГЧ та ГП приладів згідно з однозонним тарифом на електроенергію для максимального рівня свободи користувача

Результати показують, що ГП пристрої працюють при низькому енергоспоживанні, де ціни високі, і навпаки, прагнуть працювати при встановленому користувачем бажаному рівні споживаної потужності в час низьких цін. ГЧ пристрої здійснюють перенос своєї роботи на непікові години.

Мінімізація енергоспоживання в години максимального навантаження приводить до мінімальних затрат, але впливає на дискомфорт користувача. Рисунок Б.6-10 демонструє, що в години пікового навантаження енергоспоживання ГП приладів зменшується, що призводить до дискомфорту користувача. При порівнянні енергоспоживання пральної машина при середньому рівні свободи користувача на рисунок Б.9 з енергоспоживанням при мінімальному рівні свободи на рисунку Б.10 бачимо, що початок роботи був перенесений з 23.00 на 4.00, що призводить до збільшення дискомфорту.

Енергоспоживання при мінімальному режимі свободи користувача на рис. 2.9

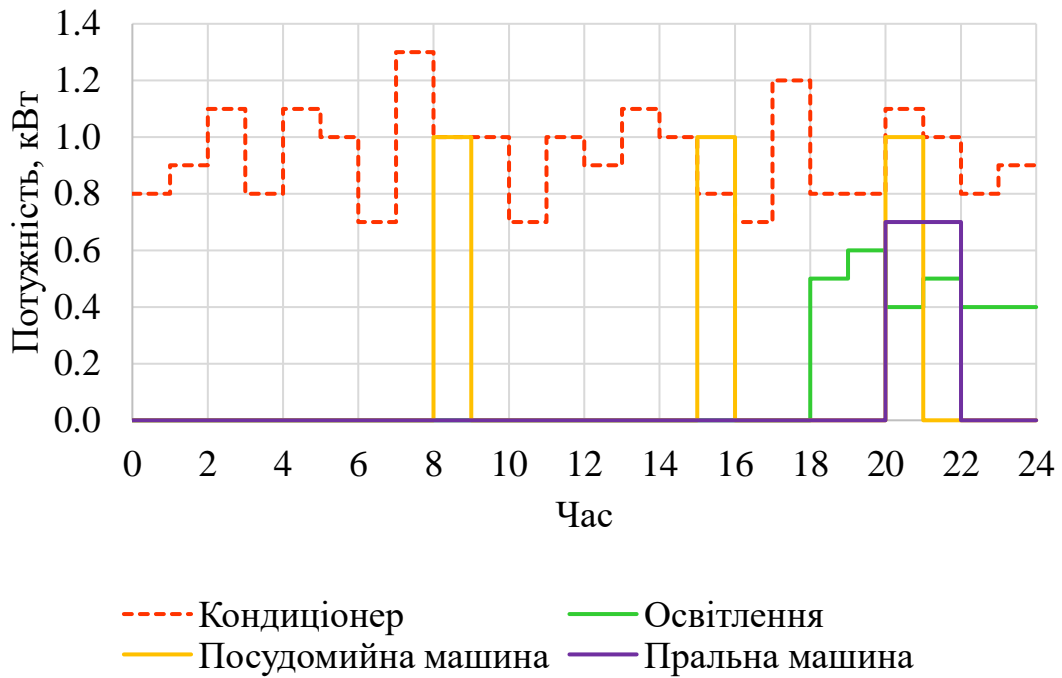


Рисунок Б.6. Планування електронавантаження ГЧ та ГП приладів згідно з однозонним тарифом на електроенергію для середнього рівня свободи користувача

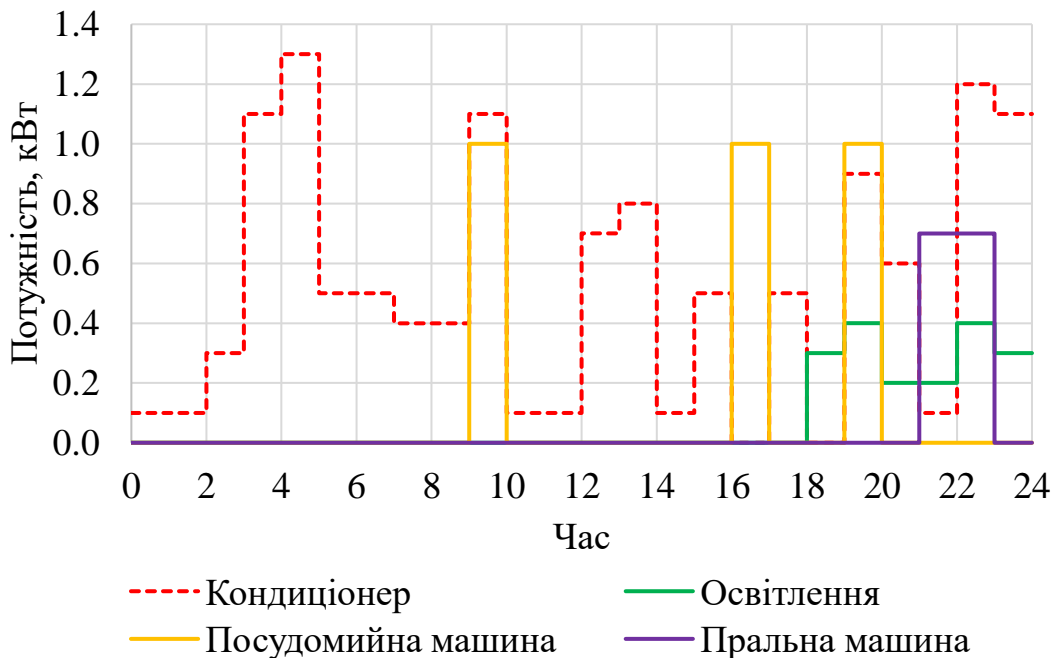


Рисунок Б.7. Планування електронавантаження ГЧ та ГП приладів згідно з однозонним тарифом на електроенергію для мінімального рівня свободи користувача

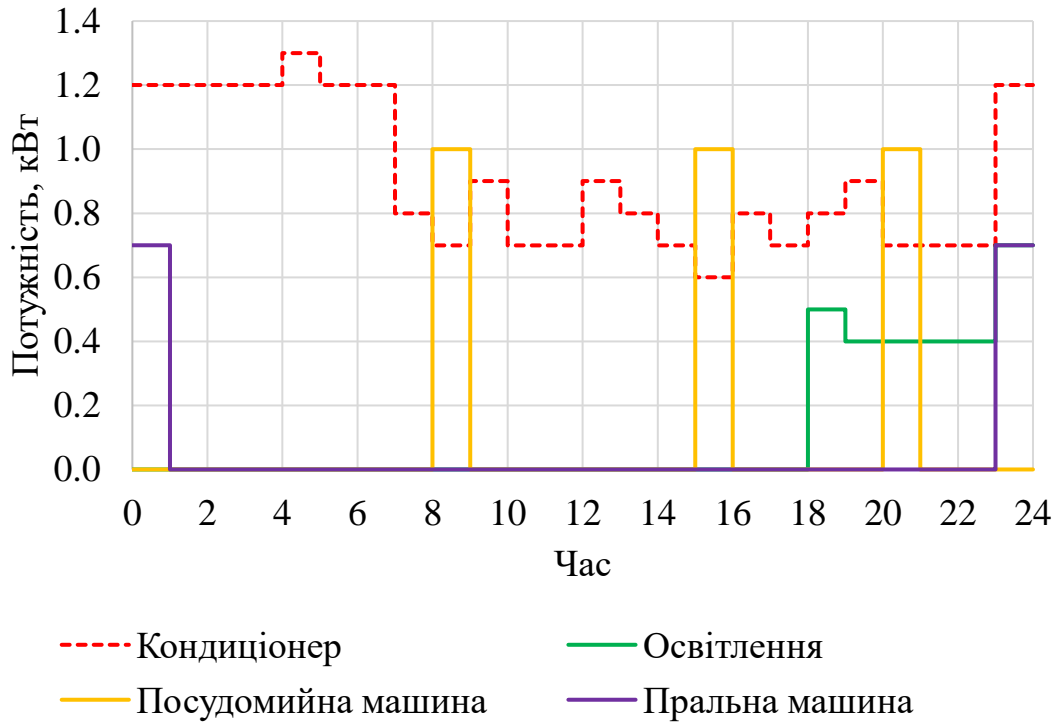


Рисунок Б.8. Планування електронавантаження ГЧ та ГП приладів згідно з двозонним тарифом на електроенергію для середнього рівня свободи користувача

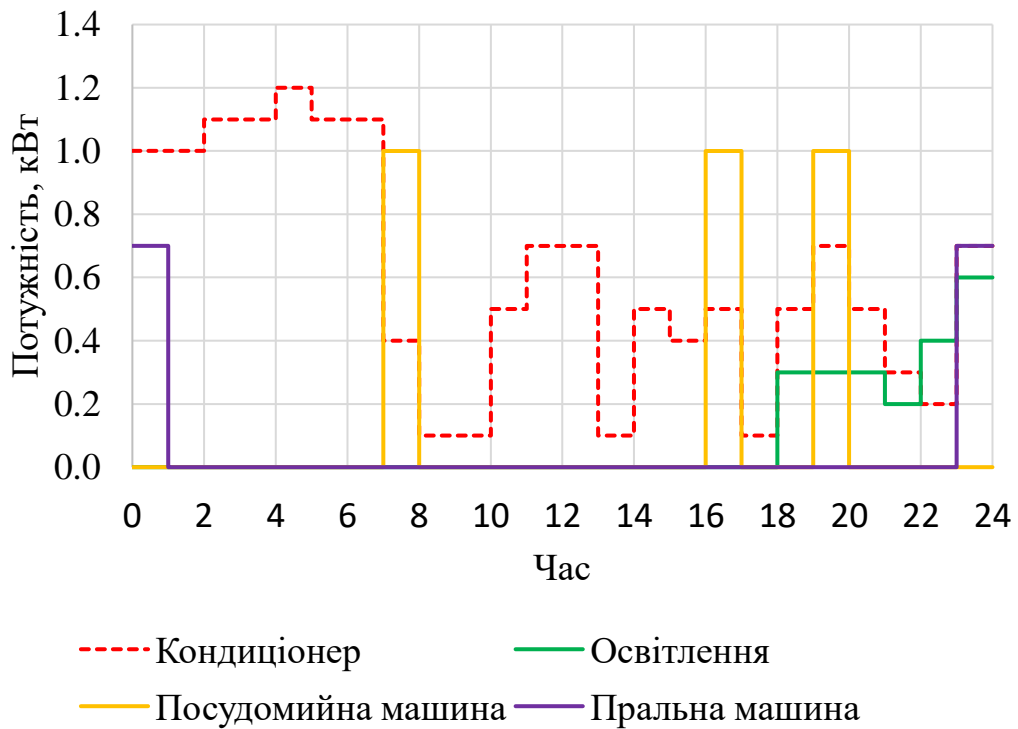


Рисунок Б.9. Планування електронавантаження ГЧ та ГП приладів згідно з двозонним тарифом на електроенергію для мінімального рівня свободи користувача

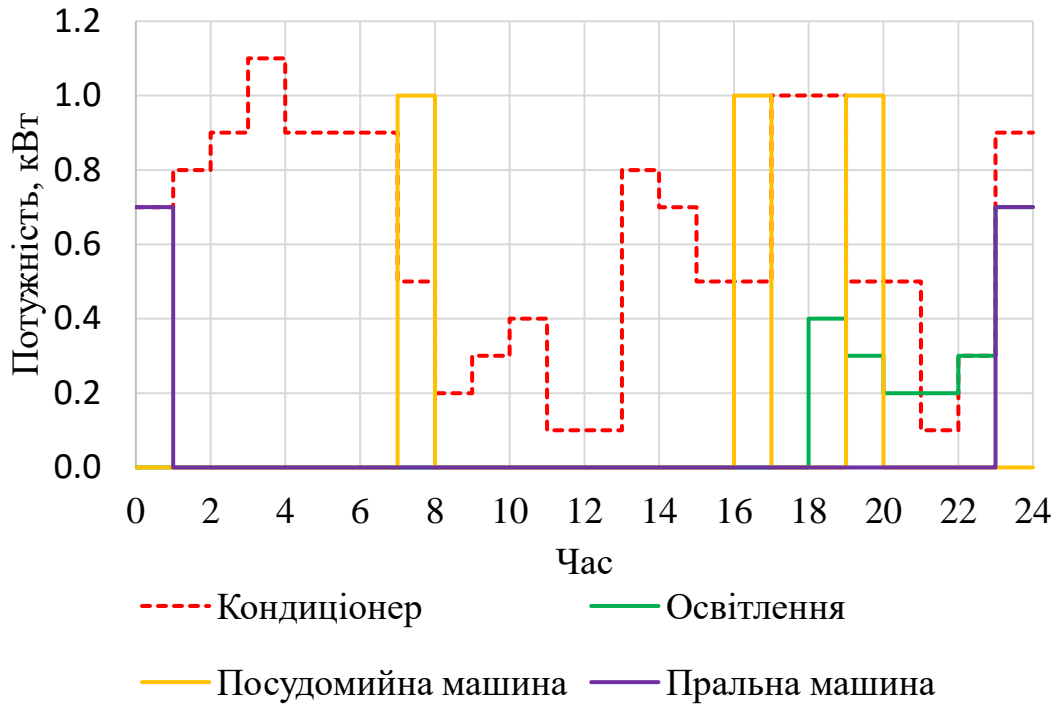


Рисунок Б.10. Планування електронавантаження ГЧ та ГП приладів згідно з трьохзонним тарифом на електроенергію для мінімального рівня свободи користувача

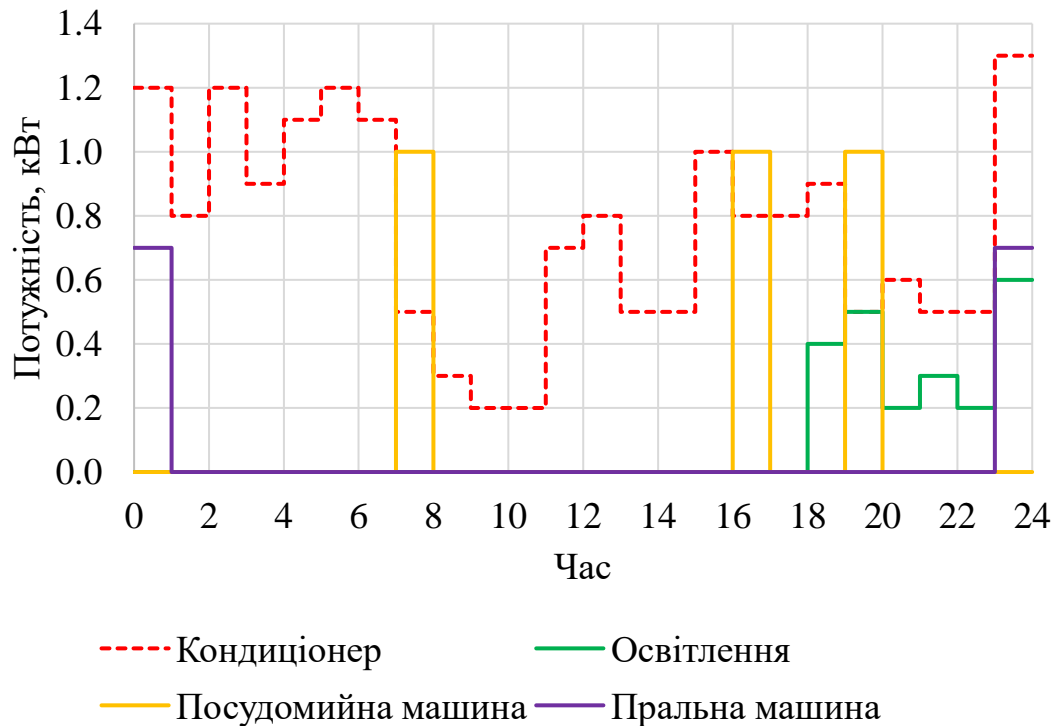


Рисунок Б.11. Планування електронавантаження ГЧ та ГП приладів згідно з шестизонним тарифом на електроенергію для мінімального рівня свободи користувача

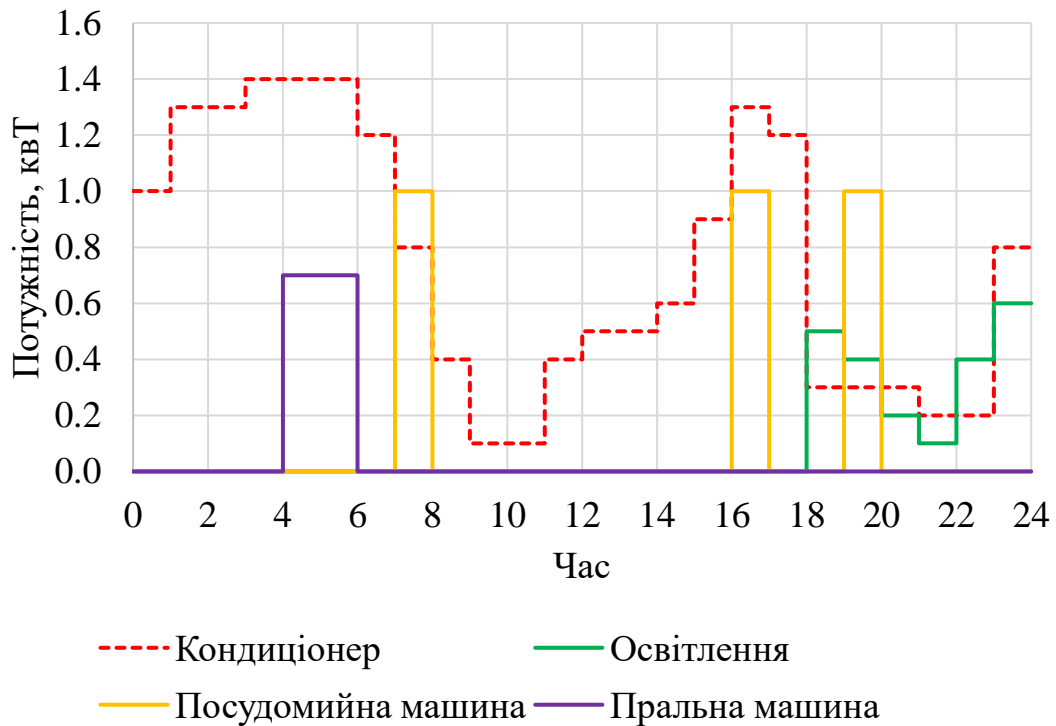


Рисунок Б.12. Планування електронавантаження ГЧ та ГП приладів згідно з двадцятичотирьохзонним тарифом на електроенергію для мінімального рівня свободи користувача

Рисунки Б.13-14 ілюструють залежність вартості електроспоживання, дискомфорту користувача і цільової функції оптимізаційного алгоритму програмного додатку при плануванні електроспоживання кондиціонеру згідно з шестизонним тарифом на електроенергію. У таблиці Б.3 наведені значення вартості електроспоживання і дискомфорту користувача для ГЧ і ГП приладів згідно з одно-, дво-, трьох-, шести- і двадцятичотирьохзонними тарифами.

У таблиці Б.4 наведено результати оптимізації електроспоживання для шести тарифних зон при максимальному.

Щоб оцінити коректність результатів моделювання сценарію задача була розв'язана методом повного перебору. Значення цільової функції повинні включатися у наступний інтервал, що спостерігається при аналізі результатів моделювання (Таблиці Б.4.-Б.6.).

- для кондиціонеру – (-36,52; -4,23);
- для освітлення – (-5,45; -0,27);
- для пральної машини – (-1,23; -0,02);

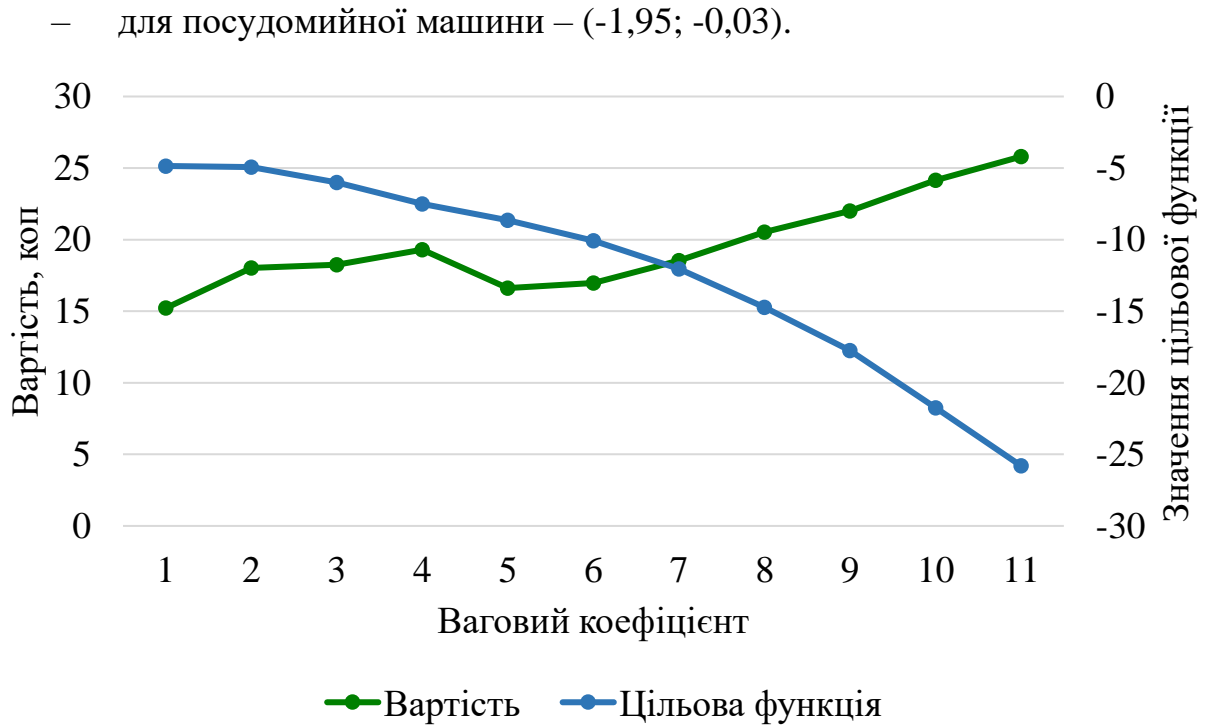


Рисунок Б.13. Залежність вартості і значення цільової функції оптимізаційного алгоритму

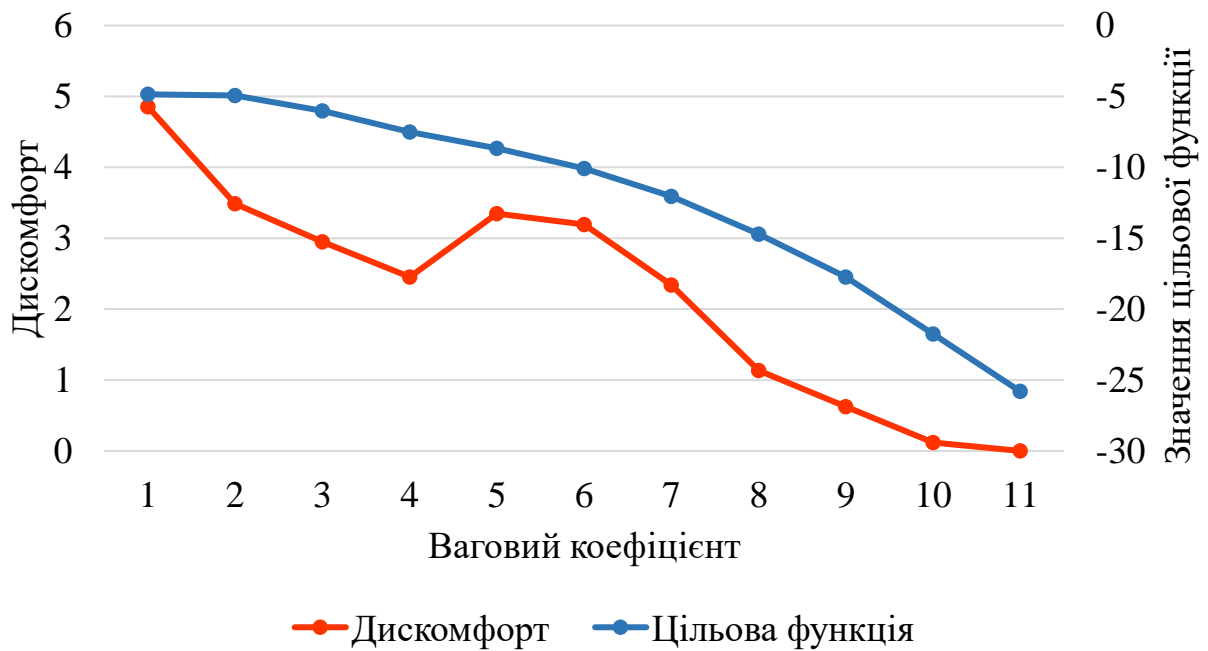


Рисунок Б.14. Залежність дискомфорту користувача і значення цільової функції оптимізаційного алгоритму

Таблиця Б.3 Вартість і дискомфорт запланованої агентом роботи приладів за добу

Прилад	Тар. з.	Макс. рівень свободи		Сер. рівень свободи		Мін. рівень свободи	
		Вартість	Дискомф.	Вартість	Дискомф.	Вартість	Дискомф.
Кондиціонер	1	28	0.13	20	3.45	11	9.16
	2	23	0.11	16	3.19	10	7.67
	3	26	0.13	18	3.08	8	9.86
	6	24	0.12	17	3.19	11	7.11
	24	23	0.13	17	2.62	9	7.71
Освітлення	1	4	0.05	3	0.56	2	1.24
	2	3	0.05	2	0.66	2	1.05
	3	4	0.04	2	0.56	2	1.03
	6	4	0.04	3	0.6	2	1.17
	24	4	0.05	3	0.47	2	1.23
Прилад	Тар. з.	Макс. рівень свободи		Сер. рівень свободи		Мін. рівень свободи	
		Вартість	Дискомф.	Вартість	Дискомф.	Вартість	Дискомф.
Пральна машина	1	1.3	0	1.3	0	1.3	0
	2	1.3	0	0.6	0.009	0.6	0.324
	3	2	0	0.5	0.009	0.5	0.29
	6	2	0	1	0	0.2	0.26
	24	2	0	0.14	0.26	0.14	0.26
Посудом. машина	1	2.7	0	2.7	0	2.7	0.027
	2	2.7	0	2.25	0.009	2.25	0.027
	3	3.6	0	2.7	0.018	2.7	0.026
	6	3.65	0	2.45	0.018	2.45	0.027
	24	3.8	0	2.15	0.027	2.15	0.027
Усього	1	36	0,18	27	4.01	17	10.427
	2	32	0.16	20.85	3.868	14.85	9.071
	3	35.6	0.17	23.2	3.667	13.2	11.206
	6	33.65	0.16	23.45	3.808	15.65	8.567
	24	32.8	0.18	22.29	3.377	13.29	9.227

Таблиця Б.4. Результати оптимізації електроспоживання для різних тарифних зон при максимальному рівні свободи

	Кіл. т. з.	Прилад	Ціл. Ф.	Час доби																									
				0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
				Максимальний рівень свободи	1	Кондиціонер	-25,53	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,4	1,3	1,3	1,3	1,4	1,3	1,3	1,3	1,3	1,3
Освітлення	-3,41	0,0	0,0			0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,7	0,7	0,7	0,7	0,7	0,7
Пральна м.	-0,78	0,0	0,0			0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,7	0,7	0,0	0,0
Посудомийна м.	-0,9	0,0	0,0			0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	
2	Кондиціонер	-20,79	1,4		1,3	1,3	1,3	1,3	1,3	1,4	1,4	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,4	1,3	1,3	1,4	
	Освітлення	-3,12	0,0		0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,7	0,7	0,7	0,7	0,7	0,8	
	Пральна м.	-0,54	0,0		0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,7	0,7	0,0	0,0		
	Посудомийна м.	-1,14	0,0		0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	
3	Кондиціонер	-23,00	1,3		1,3	1,3	1,3	1,4	1,3	1,4	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,4	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	
	Освітлення	-3,67	0,0		0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,7	0,7	0,7	0,7	0,7	0,8	
	Пральна м.	-0,31	0,0		0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,7	0,7	0,0	0,0		
	Посудомийна м.	-1,23	0,0		0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	1,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	
6	Кондиціонер	-21,75	1,3	1,3	1,3	1,4	1,4	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,4	1,3	1,3	1,4	1,3	1,3	1,3	1,3	1,3	1,3	1,3		
	Освітлення	-3,76	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,7	0,7	0,7	0,7	0,7	0,8		

ДОДАТОК В. Формування сценарію інтерактивної обробки багатовимірних даних

Багатовимірна модель даних являє собою подальший розвиток структурованої моделі даних, призначенням якої є інтерактивна аналітична обробка інформації. Багатовимірне концептуальне представлення (multi-dimensional conceptual view) [21], – це множинна, що складається з кількох незалежних вимірювань, уздовж яких можуть бути проаналізовані визначені сукупності даних для аналітичній обробки та прийняття рішень. Основні поняття, що розширюють можливості структурованої моделі даних це: агрегованість, історичність і прогнозованість даних [167].

– Агрегованість даних означає розгляд інформації на різних рівнях її узагальнення. В аналітичних системах ступінь детальності подання інформації залежить від рівня користувача: аналітик, користувач-оператор, керуючий, керівник.

– Історичність даних передбачає забезпечення високого рівня даних і їх взаємозв'язків з прив'язки даних до часу дослідження.

– Статичність даних дозволяє використовувати при їх обробці спеціалізовані методи завантаження, зберігання, індексації, вибірки та візуалізації даних засобами звичної та комфортної двовимірної бізнес-графіки [99]

– Прогнозованість даних передбачає завдання функцій прогнозування і застосування їх до різних тимчасових інтервалах.

Багатомірність моделі даних означає не багатовимірність візуалізації цифрових даних, а багатовимірне логічне представлення структури інформації при описі і в операціях маніпулювання даними. Одночасний аналіз за кількома вимірюваннями визначається як багатовимірний аналіз On-Line Analytical Processing (OLAP) [21].

Основним поняттям багатовимірної моделі даних є гіперкуб даних, що представляє собою множину осередків $H^{fep}(P^{fep}, M)$, які відповідають множині вимірів $P^{fep} = \{p_1, p_2, \dots, p_z\}$ та множині значень вимірів

$$M = M_{p_1} \cup M_{p_2} \cup \dots \cup M_{p_z}, \quad \text{де} \quad M_{p_z} = \{m_{1_z}, m_{2_z}, \dots, m_{kz}\} \quad -$$

множина міток виміру p_z (наприклад, множина міток виміру у регіональному розрізі з розбивкою по місяцям, кварталам, рокам).

Формування «зрізу» (Slice)- це підмножина гіперкуба, яка була здобута внаслідок фіксації значення одного або більшої кількості і вимірів. Формування коректного запиту до багатовимірних даних дає можливість експерту отримати зріз гіперкуба за будь-яким виміром $H'(P_z^{fep}, M_{p_z}) | H' \subseteq H$, який містить саме необхідні для дослідження дані.

Операція агрегації (Drill Up) – це операція підйому за рівнями консолідації даних у процесі аналізу або переходу від деталізованих даних до агрегованих.

Операція деталізації (Drill Down) – це операція спуску за рівнями консолідації даних або переходу від агрегованих до деталізованих даних.

Державний бюджет – це основний фінансовий план держави, який складається з дохідної й видаткової частин та затверджений законом. Моніторинг стану виконання бюджету визначається як систематичний збір та обробка інформації, яка застосовується для оцінки ефективності використання бюджетних коштів, а також підготовки та прийняття управлінських рішень [].

Інформаційна модель моніторингу держбюджету будується на базі формування багатовимірних даних, які описують бюджетні відносини:

P_z^{fip} - множина фінансових показників, через які визначають стан бюджетних відносин в країні;

F^{adb} - напрями і функції фінансового моніторингу показників, що визначають бюджетні відносини.

Зміст функцій обробки та аналізу даних F^{asb} , які характеризують моніторинг та аналіз виконання бюджету, наведено у таблиці 4.1.

Таблиця 4.1.

№	Найменування функції	Зміст та основні властивості
1.	$F_{inf\ ep}^{asb}$ - прийом та верифікація даних від різних джерел	1. Отримання звітності про виконання бюджету від різних джерел $\{Q_{\eta}^{ap}, Q_{\eta}^{fo}, Q_{\eta}^{ac}\}$ 2. Верифікація та очищення даних для поповнення БД показників бюджетного процесу $P_z^{fip} = \{P_{inc}^{fip}, P_{exp}^{fip}, P_{fin}^{fip}, P_{deb}^{fip}\}$ 3. Технологія агрегації даних та створення гіперкубів $H^{fip}(P^{fip}, M)$ для здійснення аудиту бюджету за різними напрямками
2.	$F_{db}^{asb}(P_z^{fip})$ - моніторинг виконання державного бюджету	1. Моніторинг та аудит виконання доходів $F_{db}^{asb}(H_{db}^{fip}(P_{inc}^{fip}, M))$ 2. Моніторинг та аудит виконання видатків $F_{db}^{asb}(H_{db}^{fip}(P_{exp}^{fip}, M))$ 3. Моніторинг та аудит фінансування $F_{db}^{asb}(H_{db}^{fip}(P_{fin}^{fip}, M))$ 4. Моніторинг кредитування $F_{db}^{asb}(H_{db}^{fip}(P_{deb}^{fip}, M))$ 5. Виявлення відхилень від планових показників
3.	$F_{zb}^{asb}(P_z^{fip})$ - моніторинг виконання зведеного бюджету	1. Моніторинг та аудит виконання доходів $F_{zb}^{asb}(H_{zb}^{fip}(P_{inc}^{fip}, M))$ 2. Моніторинг та аудит виконання видатків $F_{zb}^{asb}(H_{zb}^{fip}(P_{exp}^{fip}, M))$ 3. Моніторинг та аудит фінансування $F_{zb}^{asb}(H_{zb}^{fip}(P_{fin}^{fip}, M))$ 4. Моніторинг кредитування $F_{zb}^{asb}(H_{zb}^{fip}(P_{cr}^{fip}, M))$ 5. Виявлення відхилень від планових показників
4.	$F_{mb}^{asb}(P_z^{fip})$ - моніторинг виконання місцевих бюджетів	1. Моніторинг та аудит виконання доходів $F_{mb}^{asb}(H_{mb}^{fip}(P_{inc}^{fip}, M))$ 2. Моніторинг та аудит виконання видатків $F_{mb}^{asb}(H_{mb}^{fip}(P_{exp}^{fip}, M))$ 3. Моніторинг та аудит фінансування $F_{mb}^{asb}(H_{mb}^{fip}(P_{fin}^{fip}, M))$ 4. Моніторинг та аудит фінансування $F_{mb}^{asb}(H_{mb}^{fip}(P_{cr}^{fip}, M))$ 5. Виявлення відхилень від планових показників
5.	$F_{rb}^{asb}(P_z^{fip})$ - моніторинг виконання місцевих бюджетів у розрізі регіонів	1. Моніторинг та аудит виконання доходів у розрізі регіонів $F_{mb}^{asb}(H_{rb}^{fip}(P_{inc}^{fip}, M))$ 2. Моніторинг та аудит виконання видатків у розрізі регіонів $F_{mb}^{asb}(H_{rb}^{fip}(P_{exp}^{fip}, M))$ 3. Виявлення відхилень від планових показників

№	Найменування функції	Зміст та основні властивості
6.	$F_{deb}^{asb}(P_z^{fip})$ - моніторинг заборгованості	1. Моніторинг заборгованості бюджетних установ $F_{deb}^{asb}(H_{deb}^{fip}(P_{cr}^{fip}, M))$ 2. Моніторинг фінансових зобов'язань $F_{deb}^{asb}(H_{deb}^{fip}(P_{obl}^{fip}, M))$
7.	F_{an}^{asb} - поглиблений аналіз причин відхилень	1. Вибірковий контроль бухгалтерської звітності розпорядників бюджетних коштів 2. Виявлення порушень 3. Організація контрольних перевірок
8.	F_{ef}^{acb} - оцінка ефективності використання бюджетних коштів	1. Оцінка ефективності використання коштів за бюджетними програмами $\{F_{an}^{asb}(P_z^{mer})\}$ 2. Дослідження впливу бюджетних програм на соціальні показники P_z^{fip} and P_z^{mer}
9.	F_{rp}^{acb} - генерація звітних форм	1. Підготовка аналітичних матеріалів за результатами моніторингу Φ_{η}^{rp} 2. Публікація графічних даних у різних додатках, включаючи картографічне відображення

В основі моделі даних P_z^{fip} лежить структура бюджетних відносин, яка визначається показниками доходів, видатків та кредитування, фінансування бюджету та показників, що характеризують боргові зобов'язання [Д1.1]

$$P_z^{fip} = \{P_{inc}^{fip}, P_{exp}^{fip}, P_{fin}^{fip}, P_{deb}^{fip}\}.$$

Показники множини $\{P_{inc}^{fip}\}$ відображають доходи бюджету та розрізняються за податковими і неподатковими надходженнями, доходами від операцій з капіталом та трансфертами.

Основним поняттям багатовимірної моделі даних є поняття гіперкуба даних, що представляє собою множину осередків $H^{fip}(P^{fip}, M)$, які відповідають множині вимірів (напрямам) та функціям дослідження фінансових показників виконання державного бюджету, зведеного бюджету, місцевих бюджетів в цілому та у розрізі регіонів, а також заборгованості бюджетних установ. Кожен з вимірів, у більшості випадків, має ієрархічну структуру. Наприклад, ієрархічна структура «бюджет» має, як мінімум, два види ієрархічних рівнів «бюджет–

державний–місцевий–загальний фонд–спеціальний фонд» та «бюджет–доходи–видатки–заборгованість–трансфери» і таке інше.

$P^{fip} = \{p_1, p_2, \dots, p_z\}$ та множині значень вимірів (стану фінансових показників згідно з бюджетною класифікацією, прийнятою у державі)

$M = M_{p_1} \cup M_{p_2} \cup \dots \cup M_{p_z}$, де $M_{p_z} = \{m_{1_z}, m_{2_z}, \dots, m_{kz}\}$ – множина позначок виміру p_z (наприклад, множина позначок виміру у регіональному розрізі з розбивкою по місяцям, кварталам, рокам).

Враховуючи особливості бюджетного устрою України гіперкуб даних моніторингу виконання бюджетів - $H^{fip}(P_z^{fip}, M)$ можна представити як сукупність проблемно-орієнтованих гіперкубів, які відповідають напрямам моніторингу:

де $H_{db}^{fip}(P_{db}^{fip}, M)$ - гіперкуб фінансових показників виконання державного бюджету;

$H_{zb}^{fip}(P_{zb}^{fip}, M)$ - гіперкуб фінансових показників виконання зведеного бюджету;

$H_{mb}^{fip}(P_{mb}^{fip}, M)$ - гіперкуб фінансових показників виконання місцевих бюджетів;

$H_{rb}^{fip}(P_{rb}^{fip}, M)$ - гіперкуб фінансових показників виконання місцевих бюджетів у розрізі регіонів;

$H_{deb}^{fip}(P_{deb}^{fip}, M)$ - гіперкуб показників заборгованості бюджетних установ.

Найважливішим показником систем з використанням кубічних сховищ та OLAP-технології обробки даних є час відгуку, який бажано мати настільки малим, щоб не встигали розмикатися асоціативні зв'язки, які виникають в аналітика в процесі осмислення проблеми. У більшості OLAP-систем процеси

формування запитів до багатовимірних даних рознесені в часі та відокремлені від процедур відображення зрізів даних у зручному для сприйняття виді. Тому з метою зменшення часу відгуку запропоновано комбінований підхід, який на відміну від існуючих методів, об'єднує в одному алгоритмі процедури агрегації даних (формування зрізів гіперкубів) з процедурами їх візуалізації для подальшого семантичного осмислення отриманих даних.

Формування коректного запиту до багатовимірних даних дає можливість аналітику з питань бюджетного процесу отримати зріз гіперкуба за будь-яким виміром $\mathbf{H}'(P_z^{fip}, M_{P_z}) | \mathbf{H}' \subseteq \mathbf{H}$, який містить саме необхідні для дослідження дані. При цьому використання традиційних методів для реалізації гіперкуба, наприклад, набір тимчасових таблиць (набір компонент Decision Cube), не є ефективним, тому що не забезпечує необхідну швидкодію отримання необхідних даних, що дуже важливо для реалізації оперативного аналізу.

Тому в системі моніторингу застосована OLAP-технології (On-Line Analytical Processing), удосконалена за рахунок методу додаткової індексації. Суть методу полягає у додаванні до бази даних фінансових показників відповідних метаданих, які описують основні сутності предметної області, що згодом використовуються для пошуку інформації на проіндексованих базах даних. Застосування цього методу дозволило забезпечити час отримання зрізу гіперкуба за будь-яким виміром не більш 5 секунд, хоча бази даних фінансових показників налічує мільйони записів.

Склад базових модулів інтелектуальної системи «Моніторинг фінансових показників виконання бюджетів всіх рівнів» надано у Додатку 4.

Запропоновані для опису предметної області інформаційна модель дає можливість представляти процес моніторингу у вигляді сукупності типових сценаріїв поведінки експерта при дослідженні даних предметної області. Засобом досягнення цієї мети є інтуїтивно-зрозумілий інтерфейс системи моніторингу, організований у такий спосіб, що експерт оперує не з базами даних або інформаційними ресурсами системи, а з типовими сценаріями аналітичної

діяльності, завдяки яким експерт отримує потрібні зрізи даних, фактично, за один «дотик». Переваги такого підходу очевидні.

Аналітику надається Web-інтерфейс, якій віддзеркалює предметну область в цілому, тобто напрями дослідження, функції процедури тощо. Графічний інтерфейс користувача побудований у такий спосіб, що аналітик без використання мови програмування може формувати складний запит $Q_i^{fep} = (p_1, m_1) \wedge (p_2, m_2)$ за будь-якими параметрами (бюджети, доходи, видатки, розписи, призначення, звіти, роки, квартали, коди показників КЕК, КФК, КВК в бюджетній класифікації) для вибірки потрібного для дослідження зрізу даних $\mathbf{H}'(P_z^{fip}, M_{p_z}) | \mathbf{H}' \subseteq \mathbf{H}$.

Аналіз даних починається з вибору напрями дослідження, тобто вибору проблемно-орієнтованого гіперкубу $H^{fip}(P_z^{fip}, M)$. Далі експерту потрібно визначитися з функцією дослідження $F_k^{acb}(P_z^{fip}) \cup F^{acb}$ та сукупністю показників $(P_1^{fip} \wedge P_2^{fip} \wedge P_3^{fip}) \cup P_z^{fip}$, а також визначити основні виміри дослідження $P^{fip} = \{p_1, p_2, \dots, p_z\}$, та позначки вимірів $M_{p_z} = \{m_{1_z}, m_{2_z}, \dots, m_{kz}\}$ для змістовного дослідження фінансових показників. Система моніторингу автоматично формує запит Q_z^{fip} із визначеними параметрами (p_z, M_{p_z}) з метою отримання конкретного зрізу даних $\mathbf{H}'(P_z^{fip}, M_{p_z})$ та їх подальшого відображення для семантичного осмислення. Якщо отриманих даних недостатньо, формуються додаткові умови запиту Q_{z+1}^{fep} .

Сценарна модель аналізу багатовимірних даних бюджетного процесу складається з наступних кроків (Рисунок В.1):

Крок 1. Початком роботи сценарної моделі є вибір напрями дослідження фінансових показників (блок А1), та формування робочої гіпотези - тобто вибору проблемно-орієнтованого гіперкубу $H^{fip}(P_z^{fip}, M)$. Наприклад, необхідно дослідити заборгованість бюджетних установ на звітну дату. Для

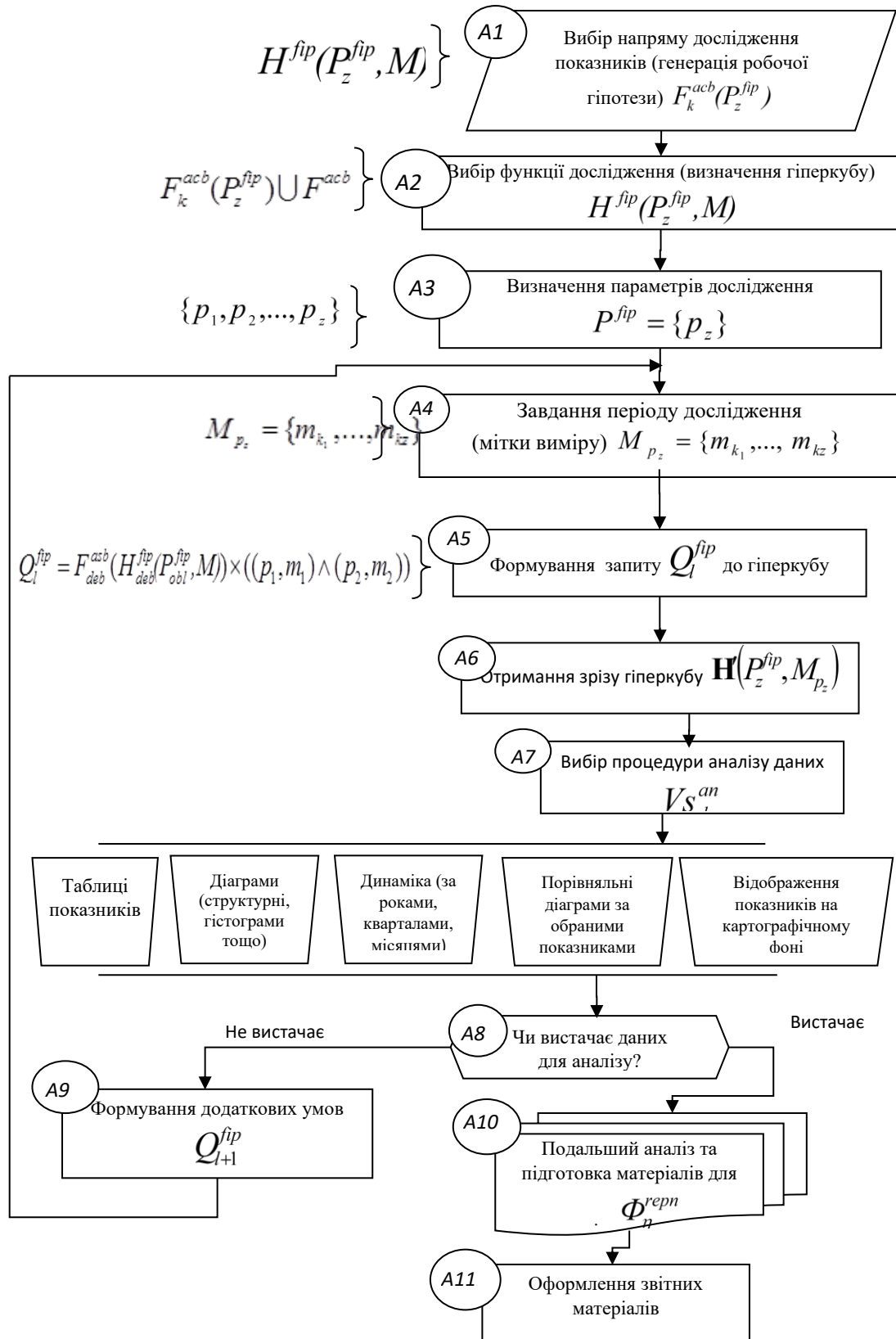


Рисунок В.1 – Сценарій аналізу багатовимірних даних бюджетного процесу цього застосування підключається до гіперкуб показників заборгованості бюджетних установ - $H_{deb}^{fip}(P_{db}^{fip}, M)$.

Крок 2. У блоці А2, на підставі обраного гіперкубу, розкривається список типових функцій, доступних експерту при дослідженні обраного напрямку. Аналітик визначає функцію моніторингу $F_k^{acb}(P_z^{fip}) \cup F^{acb}$. Наприклад, моніторинг фінансових зобов'язань $F_{deb}^{asb}(H_{deb}^{fip}(P_{obl}^{fip}, M))$.

Крок 3. У блоці А3 та А4 визначається перелік показників $P_1^{fip} \wedge P_2^{fip} \wedge P_z^{fip}$, за якими провадиться дослідження та визначається вимір $\{P_1, P_2, \dots, P_z\}$, за яким має виконуватися запит до гіперкубу та встановлюються позначки виміру $M_{P_z} = \{m_{k_1}, \dots, m_{k_z}\}$.

Крок 4. У блоці А5 формується автоматичний запит $Q_l^{fip} = F_{deb}^{asb}(H_{deb}^{fip}(P_{obl}^{fip}, M)) \times ((P_1, m_1) \wedge (P_2, m_2))$ для отримання визначеного зрізу даних гіперкубу.

Крок 5. У блоці А6 безпосередньо виконується звернення до гіперкубу і формується зріз даних $H'(P_1^{fip}, M_{P_1} \wedge P_2^{fip}, M_{P_2} \wedge P_z^{fip}, M_{P_z})$ для відображення у зручному для аналізу вигляді.

Крок 6. У блоці А7 обираються процедури візуального аналізу даних - V_s^{an} . Аналітик може обрати будь-яку форму відображення за допомогою аналітичних процедур: у вигляді таблиць, гістограм, кругових діаграм, порівняльних діаграм, діаграм динаміки за кількома вимірами, діаграм з прив'язкою до картографічної основи.

З метою усунення надлишкової функціональності перелік аналітичних процедур динамічна корелюється з напрямом та функцією моніторингу. Наприклад, процедура відображення даних на картографічному фоні притаманна тільки дослідженням місцевих бюджетів у регіональному розрізі – гіперкуб $H_{rb}^{fip}(P_{rb}^{fip}, M)$.

Крок 7. У блоці А8 (за результатами відображення даних у зручному для сприйняття виді із застосуванням різноманітних процедур аналізу даних)

виконується аналіз отриманого запиту зрізу даних $\mathbf{H}'(P_1^{fip}, M_{p1} \wedge P_2^{fip}, M_{p2} \wedge P_z^{fip}, M_{pz})$ та оцінка чи вистачає даних для аналізу висунутої гіпотези. Якщо даних недостатньо – виконується перехід до блоку А9 для формування додаткових умов. Якщо вистачає – перехід до блоку А10.

Крок 8. Формування додаткових умов з метою отримання зрізу даних з іншою позначкою виміру $M_{pz} = \{m_{k_1}, \dots, m_{k_z}\}$ виконується у блоці А9.

Кроки 3–8 виконуються в циклі, доки не отримаємо зріз $\mathbf{H}'(P_1^{fip}, M_{p1} \wedge P_2^{fip}, M_{p2} \wedge P_z^{fip}, M_{pz})$, з достатніми для аналізу даними.

Крок 9. Кінцевим кроком сценарію є експорт даних до обраної форми звітності - Φ_{η}^{rep} : доповідна записка, аналітична довідка, аналітичний звіт, експертний висновок та інші.

Цикл агрегації та аналізу багатовимірних даних повторюється для всіх актуальних показників. Після завершення моніторингу показників виконується генерування узагальненого звіту. Як правило, узагальнений звіт містить як текстові, так і графічні фрагменти аналітичних матеріалів.

Архітектура системи моніторингу бюджетного процесу (діюча аналітична система Рахункової палати України) побудована як трирівнева модель <Клієнт – Web-сервер (відображення) – База даних> на принципах вільного зв'язування сервісів, та можливості повторного їх використання і композиції.

Рівень <Бази даних> реалізує задачі з надійного збереження різноманітних показників бюджетного процесу, представлених багатовимірною моделлю, санкціонованого доступу, модифікації, архівування/відновлення даних. Сервер баз даних реалізовано на основі СУБД Oracle Database Enterprise Edition 11G із застосуванням механізмів створення кубів даних – OLAP cube data. БД моніторингу бюджетного процесу складається з

Tables: 60 Columns: 736 Constraints: 48

Та за станом на 05.2018 року містить більш 16 000 000 записів.

Використання механізму OLAP cube data дозволяє реалізувати вищеописану багатовимірну модель даних бюджетного процесу за допомогою стандартних процедур Oracle 11G:

```
-- System privileges granted to OLAP_DBA
GRANT CREATE ANY CUBE TO OLAP_DBA;
GRANT CREATE ANY CUBE BUILD PROCESS TO OLAP_DBA;
GRANT CREATE ANY CUBE DIMENSION TO OLAP_DBA;
GRANT CREATE ANY MEASURE FOLDER TO OLAP_DBA;
GRANT CREATE ANY TABLE TO OLAP_DBA;
GRANT CREATE ANY VIEW TO OLAP_DBA;
GRANT CREATE JOB TO OLAP_DBA;
GRANT CREATE SEQUENCE TO OLAP_DBA;
GRANT DELETE ANY CUBE DIMENSION TO OLAP_DBA;
GRANT DELETE ANY MEASURE FOLDER TO OLAP_DBA;
GRANT DELETE ANY TABLE TO OLAP_DBA;
GRANT DROP ANY CUBE TO OLAP_DBA;
GRANT DROP ANY CUBE BUILD PROCESS TO OLAP_DBA;
GRANT DROP ANY CUBE DIMENSION TO OLAP_DBA;
GRANT DROP ANY MEASURE FOLDER TO OLAP_DBA;
GRANT DROP ANY TABLE TO OLAP_DBA;
GRANT DROP ANY VIEW TO OLAP_DBA;
GRANT INSERT ANY CUBE DIMENSION TO OLAP_DBA;
GRANT INSERT ANY MEASURE FOLDER TO OLAP_DBA;
GRANT INSERT ANY TABLE TO OLAP_DBA;
GRANT SELECT ANY CUBE TO OLAP_DBA;
GRANT SELECT ANY CUBE DIMENSION TO OLAP_DBA;
GRANT SELECT ANY TABLE TO OLAP_DBA;
GRANT UPDATE ANY CUBE TO OLAP_DBA;
GRANT UPDATE ANY CUBE BUILD PROCESS TO OLAP_DBA;
GRANT UPDATE ANY CUBE DIMENSION TO OLAP_DBA;
GRANT UPDATE ANY TABLE TO OLAP_DBA;
```

Механізм OLAP cube data дозволяє формувати складні запити до ДБ бюджетного процесу, використовуючи стандартні SQL-запити СУБД Oracle 11G та отримувати потрібні зрізи даних $\mathbf{H}(P_1^{fip}, M_{p1} \wedge P_2^{fip}, M_{p2} \wedge P_z^{fip}, M_{pz})$.

На основі зазначеної СУБД Oracle 11G розгорнуто Real Application Cluster з метою забезпечення масштабованості системи моніторингу, а також високої доступності рівня збереження даних, мінімізації ризиків втрати даних.

Середній рівень <Відображення> реалізує процеси звернення до ДБ бюджетного процесу на основі технології Universal Description, Discovery and Integration (UDDI), яка передбачає ведення реєстру Web-сервісів. Web-сервер обслуговує запити від клієнта з метою отримання зрізів даних від сервера БД, проводить необхідні обчислення і перетворення інформації, формує та публікує у власній файлової системі Web-сторінку, яка відображається на АРМі користувача стандартними засобами WEB-браузера (Microsoft Internet Explorer). Фізично рівень <Відображення> складається із Web-серверу програмних застосувань для відображення Web-сервісів на АРМі кінцевого користувача.

Клієнтський рівень складається з типових сценаріїв та інтерфейсів, які саме й реалізують закладену функціональність щодо дослідження багатовимірних даних бюджетного процесу. В інтерфейсі сценарної моделі багатовимірних даних визначені параметри звернення до OLAP cube data нього через множину вимірів (напрямів) та функцій дослідження фінансових показників виконання державного бюджету (зведеного бюджету, місцевих бюджетів в цілому та у розрізі регіонів, а також заборгованості бюджетних установ), тобто інтерфейс визначає суть сервісу, а не технологію його реалізації.

Аналітик за допомогою процедури SYS_GRAPH_ANALIZ може налаштувати форму відображення зрізу даних, використовуючи наступні атрибути:

- KOD_AN IS 'Код аналізу';
- TYPE_BY_BUD IS 'Тип за бюджетом (державний /зведений/ місцевий)';

- TYPE_BY_PER IS 'Тип за періодом (квартальний/міжквартальний/міжрічний)';
- TYPE_BY_STAT IS 'Тип за статтею (видатки/доходи)';
- TYPE_BY_FOND IS 'Тип за фондом (загальний /спеціальний /разом)';
- CAPTION_AN IS 'Повна назва';
- SHORT_AN IS 'Коротка назва';
- YEAR_AN IS 'Рік';
- KVARTAL_AN IS 'Квартал';
- DRAW_TYPE_AN IS 'Тип діаграми';
- DRAW_LEGEND_AN IS 'Відобразити легенду (так/ні)';
- DRAW_LEGEND_POS_AN IS 'Позиція легенди';
- DRAW_BY_ROW_AN IS 'Діаграма по рядкам/стовбцям';
- DRAW_CAPTION_FROM_COLUMN_AN IS 'З якого поля брати назву для діаграми.

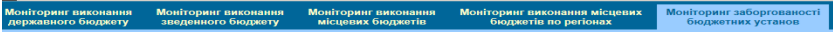
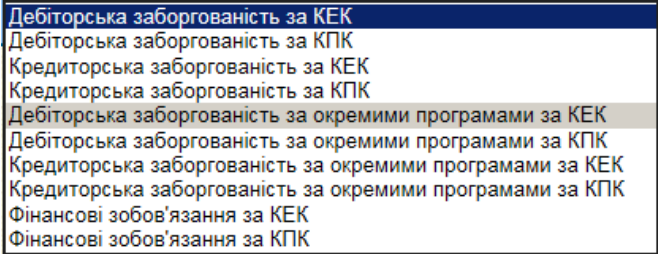
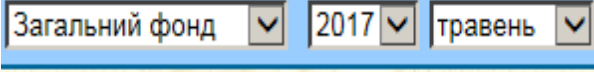
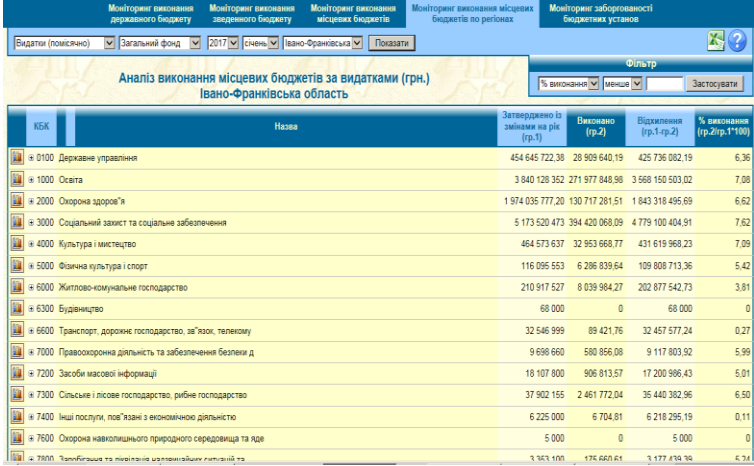
А також у вигляді таблиць, гістограм, порівняльних діаграм, діаграм динаміки за кількома вимірами, – SYS_GRAPH_ANALIZ_SERIES, кругових діаграм SYS_GRAPH_ANALIZ_STRUCT, діаграм з прив'язкою до картографічної основи – SYS_MAP_ANALIZ.

Web-сервер пропонує єдину схему взаємодії сервісів незалежно від того, чи знаходиться сервіс в тому ж самому програмному застосуванні, або в іншому адресному просторі. Кінцеві користувачі взаємодіють з Web-сервісом у спосіб, зазначений в його опису, використовуючи повідомлення в стандарті SOAP, що передаються з використанням HTTP і XML і в поєднанні з іншими стандартами, що відносяться до Web-технології.

Оскільки обсяг БД моніторингу постійно зростає для прискорення обміну використовуються сучасні Web-браузери Internet Explorer 9, 10 та вище. Цей Web-браузер оснащений новим движком JavaScript «Chakra», що збільшує продуктивність виконання сценаріїв на мові JavaScript в окремих випадках до 10 разів у порівнянні з Internet Explorer 6. Web-браузер також оснащений

процесором підтримки мови ECMAScript 5, яка застосовується як мова опису сценаріїв взаємодії різноманітних Web-сервісів.

На рисунку В.2 наведено покрокову реалізації моделі формування сценарію аналізу багатовимірних даних на прикладі бюджетного процесу. Слід підкреслити, що сценарії аналізу розглядається з позиції інтерфейсу користувача – тобто бюджетного аналітика.

<p>Крок 1. Вибір напрямку дослідження (моніторингу) - проблемно-орієнтованого гіперкубу $H^{fip}(P_z^{fip}, M)$.</p>	<p>Головне меню ситеми моніторингу</p>  <p>$H_{deb}^{fip}(P_{db}^{fip}, M)$ - гіперкуб показників заборгованості бюджетних установ, який відповідає напрямку</p>
<p>Крок 2. Вибір функції дослідження $H^{fip}(P_z^{fip}, M)$</p>	<p>Меню вибору функції дослідження, яке відповідає напрямку моніторингу (для кожного напрямку формується окреме меню функцій)</p> 
<p>Крок 3. Визначення показників $H^{fip}(P_z^{fip}, M)$, виміру $\{P_1, P_2, \dots, P_z\}$ та позначок виміру $H^{fip}(P_z^{fip}, M)$.</p>	<p>Панель задання показників, виміру та позначок виміру</p> 
<p>Крок 4. Формування автоматичного запиту Q_i^{fip} для отримання зрізу даних</p>	<p>Показати функція виконання запиту з визначеними параметрами</p> $Q_i^{fip} = F_{deb}^{asb}(H_{deb}^{fip}(P_{obl}^{fip}, M)) \times ((p_1, m_1) \wedge (p_2, m_2))$
<p>Крок 5. Виконання запиту Q_i^{fip} та відображення зрізу даних у табличному вигляді</p>	<p>Відображення зрізу даних</p>  <p>Зріз - $H'(P_1^{fip}, M_{p_1} \wedge P_2^{fip}, M_{p_2} \wedge P_z^{fip}, M_{p_z})$</p>

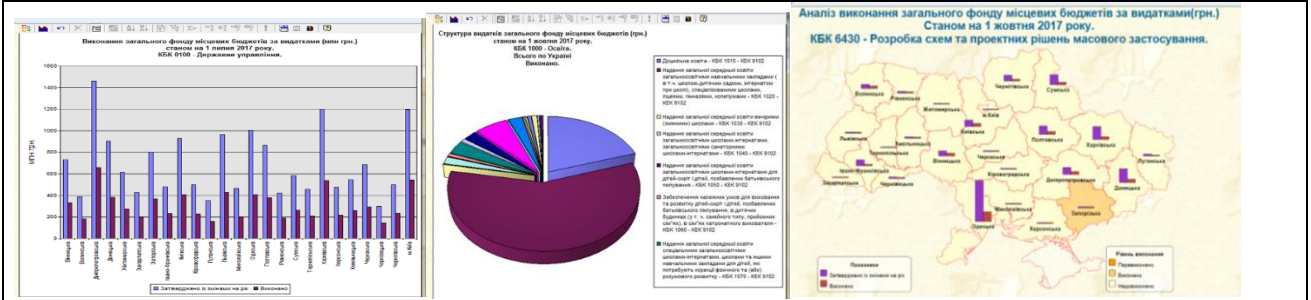
Крок 6. Вибір процедури візуального аналізу даних $V S_d^{an}$

Меню вибору процедур візуального аналізу даних

- Динаміка (помісячна)
- Динаміка (поквартальна)
- Структура
- Видатки за КЕКВ
- Порівняння за регіонами (діаграма)**
- Порівняння за регіонами (табличний аналіз)
- Порівняння за регіонами (Картографічний аналіз)

3 метою усунення надлишкової функціональності перелік аналітичних процедур динамічне корелюється з напрямом та функцією моніторингу

Приклади візуального аналізу отриманого зрізу даних



Крок 7 та 8 – оцінка та аналіз отриманого зрізу даних

За результатами відображення даних у зручному для сприйняття виді (із застосуванням різноманітних процедур аналізу даних) виконується аналіз отриманого запиту зрізу даних $H'(P_1^{fip}, M_{p1} \wedge P_2^{fip}, M_{p2} \wedge P_z^{fip}, M_{pz})$ та оцінка чи вистає даних для аналізу висунутої гіпотези. Якщо даних недостатньо, формування додаткових умов з метою отримання зрізу даних з іншою позначкою виміру $M_{pz} = \{m_{k1}, \dots, m_{kz}\}$

Крок 9 експорт даних до обраної форми звітності - Φ_{η}^{rep} :

Приклад перенесення зрізу даних для подальшого аналізу в Excel

КЕК	Назва	Затверджено на замовлення на рік (р.1)	Виконано (р.2)	Відхилення (р.1-р.2)	% виконання (р.2/р.1*100)
010000	Державне управління	702 847 118,46	565 042 803,91	137 804 314,55	80,45
060000	Правоохоронна діяльність та забезпечення безпеки держави	17 821 931,00	13 323 102,53	4 498 828,47	74,76
060702	Місцева пожежна охорона	17 821 931,00	13 323 102,53	4 498 828,47	74,76
061007	Інші правоохоронні заходи і заклади	0,00	0,00	0,00	0,00
070000	Освіта	3 776 664 941,43	3 130 798 235,93	645 866 705,50	82,90
080000	Охорона здоров'я	2 397 983 398,63	2 016 678 607,84	381 304 790,79	84,11
080101	Лікарні	1 234 098 228,09	1 045 202 323,32	188 895 904,77	84,69
080201	Спеціалізовані лікарні та інші спеціалізовані заклади (центри,	249 698 697,00	235 780 822,52	13 917 874,48	96,42
080203	Перинатальні центри, пологові будинки	29 568 441,00	25 483 511,72	4 084 929,28	86,22
080205	Санаторії для дітей та підлітків (нетуберкульозні)	38 921 700,00	32 899 602,31	6 022 097,69	84,53
080207	Будинки дитини	25 437 100,00	22 349 883,10	3 087 216,90	87,86
080208	Станції переливання крові	12 844 200,00	11 396 198,90	1 448 001,10	88,73
080209	Центри екстреної медичної допомоги та медицини катастроф.	1 465 973 386,50	1 239 623 403,57	226 349 982,93	88,20
080300	Поліклініки і амбулаторії (крім спеціалізованих поліклінік та	25 284 257,00	20 949 772,21	4 334 484,79	82,86
080400	Спеціалізовані поліклініки (в т.ч., диспансери, медико-санітарні	11 312 400,00	8 583 620,73	2 728 779,27	75,88
080500	Загальні і спеціалізовані стоматологічні поліклініки	17 941 805,00	14 236 369,17	3 705 435,83	79,35
080600	Фельдшерсько-акушерські пункти	690 800,00	562 979,71	127 820,29	72,81
080704	Центри здоров'я і заклади у сфері санітарної освіти	659 100,00	534 260,05	124 839,95	81,06
080800	Центри первинної медично (медико-санітарної) допомоги	378 150 645,04	310 877 165,40	67 273 483,64	82,21
081001	Медико-соціальні експертні комісії	6 923 300,00	6 129 360,40	793 939,60	88,53

Рисунок В.2. – Покрокова реалізація моделі сценарного аналізу багатомісних даних на прикладі бюджетного процесу з позиції інтерфейсу користувача

Використання багатомісних моделі у системах інтерактивної аналітичної обробки даних має певні переваги:

Зручність і ефективність аналітичної обробки великих обсягів даних, пов'язаних з часом. Середній час відгуку на нерегламентовані запити при використанні багатовимірної моделі на один-два порядки менше, ніж у випадку звичайній реляційної моделі з нормалізованою схемою даних.

Багатовимірне представлення даних дозволяє реалізувати інструментарій аналітика на основі вбудованих функцій аналізу даних, які не підтримуються засобами реляційних СУБД.

Недоліком багатовимірної моделі даних є її громіздкість для вирішення найпростіших завдань звичайної оперативної обробки інформації.

ДОДАТОК Г. Побудова та виконання сценаріїв аналітичної діяльності в задачах просторового моделювання.

На формування сценаріїв аналітичної діяльності впливають не тільки знання, пов'язані з особливостями функціонування предметної області, але й технологічне середовище, в якому має відбуватися реалізація такого сценарію [1, 2]. Найбільш яскравим прикладом впливу технології на сценарії вирішення специфічних завдань аналізу наслідків природних явищ є галузь GIS-аналітики. Методологічно GIS-технології прямо чи опосередковано пов'язані з такими областями знань як обчислювальна, аналітична і диференціальна геометрія, машинна графіка, розпізнавання образів, аналіз сцен, цифрова фільтрація, автоматична класифікація, обробка цифрових зображень растрових GIS [3]. Як показує досвід, головною перешкодою широкого застосування GIS-технології є, по-перше, занадто складна організація інструментальних засобів, яка потребує ретельного вивчення можливостей GIS-технології, а по-друге, відсутність типових сценаріїв обробки просторових даних, що спрямовані на вирішення певного колу задач та зрозумілі аналітику. Зокрема, прогнозування наслідків повеней, забруднення або інших екологічних задач. Тобто сценарії GIS-аналітики обов'язково мають враховувати технологічні особливості просторової організації та обробки даних, які виконуються різними фахівцями, у співвідношенні з послідовністю дій аналітика при вирішенні функціональних задач та прийняття рішень за результатами їх аналізу.

Сценарієм GIS-аналітики слід вважати не тільки опис послідовності виконання аналітичних кроків, GIS функції, процедур та використання відповідних інформаційних ресурсів, але також й зміни станів об'єкту дослідження, які можуть виникнути в процесі виконання сценарію GIS-аналітики. Це потребує залучення нових технологій, зокрема таких які використовуються для опису бізнес процесів, з метою формування сценаріїв взаємодії різних за фахом та кваліфікацією експертів, наприклад технології BPMN, яка поєднується с семантичною моделлю ПрО.

Підхід, що пропонується для формування багаторівневих сценаріїв Анд у галузі GIS-технологій, ґрунтуються на методології нотації BPMN 2.0, яка заснована на таких фундаментальних положеннях сучасних технологій проектування, а саме:

1. Уявленні будь-якого процесу (сценарію) як сукупності взаємопов'язаних та взаємодіючих елементів (сутностей предметної області), спрямованих на досягнення кінцевого результату. В якості елементів припускаються комбінації різноманітних сутностей – завдання, процеси, підпроцеси, виконавці, ролі, інформація, дані, ПЗ, логічні оператори, умови логічних переходів тощо.

2. Відображенні моделі процесу у вигляді спрямованого графу, що однозначно відтворює функціональну взаємодію елементів, з яких складається процес.

3. Об'єднанні в нотації BPMN різних типів процесів як внутрішніх (Private Process) для опису потоку операцій в межах окремого процесу, так і публічних (Public Process) для відображення взаємодії між Private Process.

4. Строгості та точності виконання правил опису процесів, графічних елементів та їх взаємодії, без внесення в той же час надмірних обмежень на дії аналітика.

5. Підтримці концепції ітеративного моделювання, що дозволяє виконувати покрокове формування моделі процесу з детальним обговоренням та редагуванням запропонованих варіантів.

6. Серіалізації візуальних моделей сценаріїв аналітики у форматі мов XML та BPMML для їх подальшої комп'ютерної обробки та виконання в BPMS-IT – системах.

Враховуючи складність процесу формування сценаріїв, які описують взаємодію різних за фахом та кваліфікацією експертів-аналітиків при вирішенні завдань просторового моделювання наслідків НС, робоче місце експертів оснащено безкоштовним графічним редактором BizAgi Modeler. Цей редактор

дозволяє побудувати та візуалізувати багаторівневий сценарій GIS-аналітики за стандартом методології BPMN. Принциповою особливістю методології BPMN, є можливість моделювати сценарії за допомогою графічного редактора, що оперує з елементами нотації BPMN 2.0. Таки елементи мають стандартні позначення, строго визначену семантику та, як правило, однозначно сприймаються аналітиком. Крім того, модель сценарію АнД в нотації BPMN може серіалізується редактором Bizagi Modeler у файли різних форматів XML, XPDL та BPMN, які базуються на синтаксисі мови XML та можуть бути опубліковані в Web-браузері або на сайті SharePoint, а також представлені в форматах MS Word, MS Excel, PDF.

Г.1 Формування типового сценарію GIS-аналітики при просторовому моделюванні наслідків природних явищ

Розробку моделі сценарію GIS-аналітики доцільно починати з визначення глибини і масштабу моделі аналітичного процесу. Тобто, модель сценарію є, по суті, відповіддю на декілька основних питань, які дозволять провести всебічний аналіз з усіх точок зору на сценарії та деталізувати його кроки. Найбільш вагомими серед них є такі питання, які подані в порядку їх важливості:

1. Що є початковими та кінцевими умовами функціонування сценарію GIS-аналітики та як сформульована кінцева мета дослідження?
2. Які вихідні документи, інформація або рішення є результатом виконання сценарію GIS-аналітики?
3. Які учасники (participant) мають бути залучені для виконання сценарію та які ролі вони виконують?
4. Які завдання, функції, процедури (Task) та в якій послідовності необхідно виконати учасникам для отримання бажаного кінцевого результату?
5. Які інформаційні ресурси, бази даних, знання або події-умови потрібно залучати для виконання кожного окремого кроку сценарію?
6. Які параметри потрібні для формування взаємодії та переходів на альтернативні шляхи розгалуженого сценарію?

7. Які нормативні документи, інформація або інші ресурси необхідні для виконання кожної процедури сценарію?

Безумовно, відповіді на ці питання змінюються протягом процесу проектування моделі розгалуженого сценарію, але в будь-який момент часу вони дозволяють обмежити масштаб моделі, якщо вона стає занадто складною.

Розглянемо побудову сценарію GIS-аналітики при прогнозуванні наслідків паводкової ситуації. Вхідними даними для просторового моделювання є триангуляційна модель поверхні води і модель рельєфу рік басейну Тиси. В результаті моделювання з'являється зона затоплення позначеної території, яка зберігається в просторовій БД та використовується у подальшому для аналізу наслідків НС і оцінки ризиків паводків, а також можливості виникнення каскадних подій, пов'язаних з наслідками НС. GIS-компоненти використовують багатий арсенал методів просторового аналізу, які реалізовані в програмній компоненті ArcObject.

Для моделювання сценарію GIS-аналітики залучаються три різних за фахом учасника: експерт з надзвичайних ситуацій – для аналізу розвитку ситуації та підготовки рішень щодо усунення наслідків; експерт картограф – сертифікований фахівець у галузі GIS-технологій та актуалізації даних території; експерт гідролог – фахівець з гідрологічного моделювання та прогнозування. Загальний сценарій GIS-аналітики при вирішенні завдань просторового аналізу наслідків паводкової ситуації складається з наступних кроків:

1. Отримання синоптичних повідомлень про можливість виникнення НС.
2. Аналіз синоптичних повідомлень та даних з гідропостів на предмет виявлення надзвичайної ситуації з катастрофічними наслідками.
3. Актуалізація цифрової моделі території НС засобами ArcGIS Desktop.
4. Моніторинг гідрологічної ситуації з залученням засобів ArcGIS Portal.
5. Гідрологічне моделювання різних варіантів розвитку паводкової ситуації.

6. Прогнозування наслідків паводкової ситуації з метою оцінки можливості виникнення каскадних подій.

7. Інтерпретація результатів моделювання із залученням методів інтерполяції та аналітичних функцій GIS.

8. Аналіз наслідків НС та публікація результатів моделювання у Web-Portal для надання доступу для певного кола користувачів.

Для створення та редагування безпосередньо моделі сценарію GIS-аналітики та його візуального відображення використовувався графічний редактор BizAgі Process Modeler Version 3.6. (Рисунок Г.1).

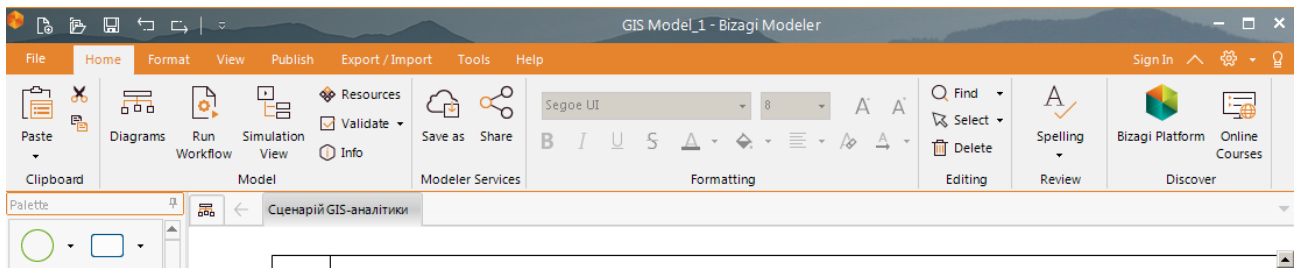


Рисунок Г.1 – Графічний редактор BizAgі Process Modeler Version 3.6.

Формування сценарію GIS-аналітики у графічному редакторі BizAgі Modeler являє собою ітераційний процес.

Першим кроком є визначення унікальної назви сценарію та його базових властивостей, які цей сценарій ідентифікують.

Наступним кроком – визначення кола учасників (participant) та їх ролей. Найбільшою проблемою є саме узгодження розподілу робіт між учасниками та визначення умов взаємодії з урахуванням рівня компетенції, прав доступу до інформації, повноважень, а також настання відповідних умов для переходів до різних рівнів. Механізмом, який дозволяє це реалізувати, є поняття "доріжка" – <Lane> – область, яка відображає послідовність кроків, що виконує конкретний учасник. Кожний учасник або роль позначається окремим ім'ям, яке привласнюється пулу з графічним відображення взаємодії між елементами нотації, що описують послідовність дій саме цього учасника.

Згідно стандарту ISO/IEC 19510 [8] будь-який процес в нотації BPMN обов'язково починається із стартової події – <Start Events>. Для цього з палітри

<Events> необхідно перетягнути елемент <Start Events>, використовуючи механізм «drag-and-drop», в головний пул та визначити його назву.

Для продовження формування сценарію з меню панелі <Activities> обирається графічний елемент <Task> - задача (Рисунок Г.2). BizAgi Modeler розрізняє сім типів задач, що можуть виконуватися безпосередньо користувачем або зовнішніми процесами. Обраний елемент супроводжується додатковою інформацією - тип задачі, яка пояснює особливості його використання, наприклад, <Receive Task> або <User Task>.

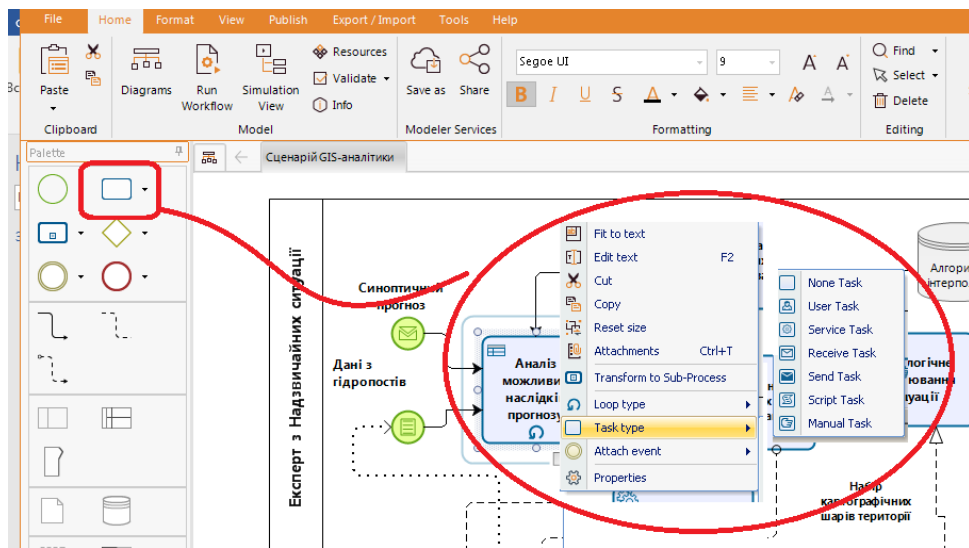


Рисунок Г.2 – Приклад завдання параметрів елементу <Task> при побудові сценарію

В нотації BPMN 2.0 кожне завдання може виконуватися або одноразове, або в режимі циклічного повторення. Крім того, кожне завдання може мати окремі екземпляри (примірники). На семантичному рівні примірники можна порівняти з виконавцями процесу, що мають ім'я та відрізняються конкретними умовами виконання, які, зазвичай, задаються логічними операторами шлюзів (Gateway) або тригерами.

Приклад відображення сценарію GIS-аналітики в нотації BPMN показано на рисунку Г.3. Моделювання на платформі BizAgi являє собою ітераційний процес.

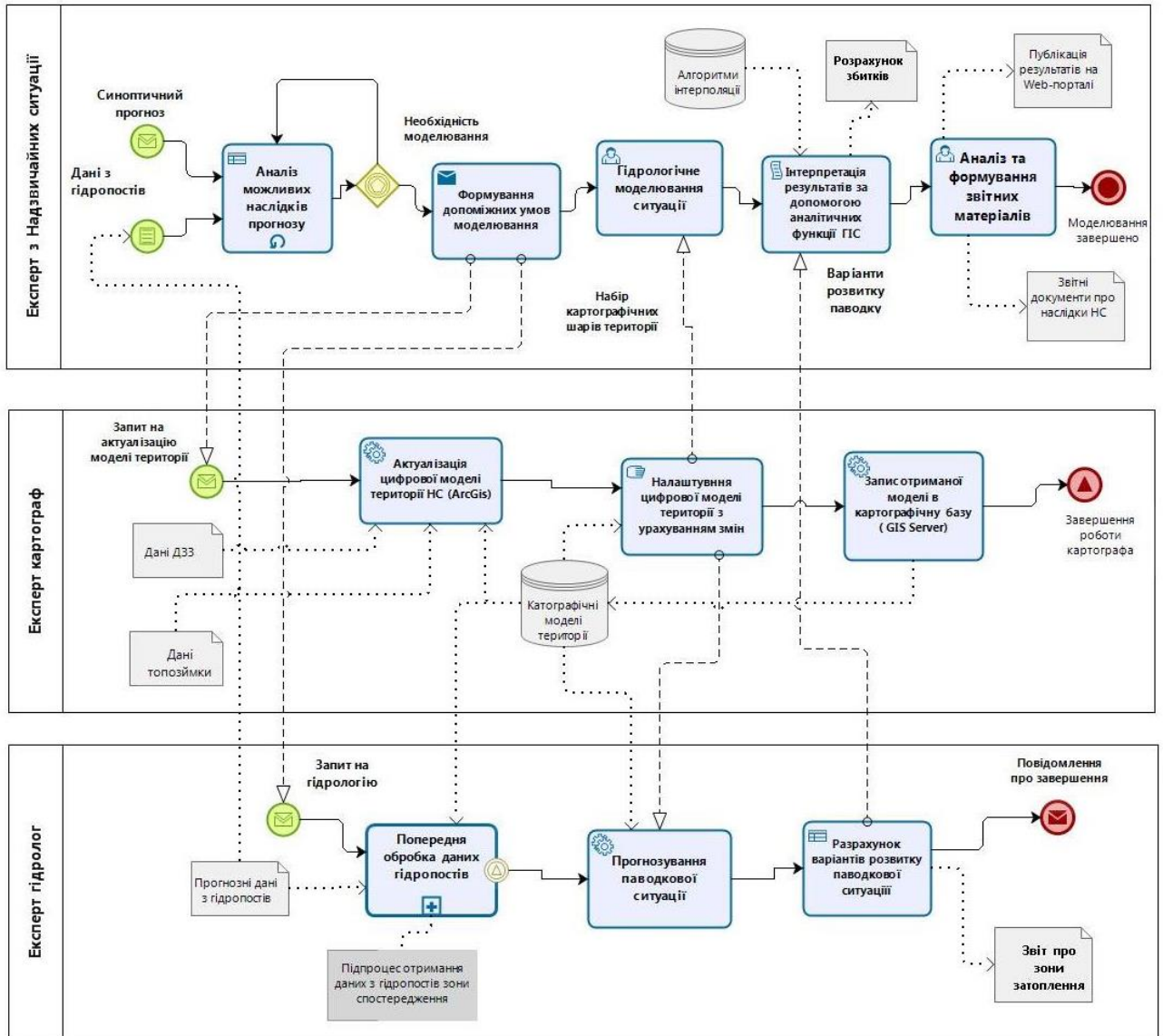


Рисунок Г.3 – Приклад відображення сценарію GIS-аналітики в нотації BPMN при просторовому моделюванні наслідків паводкової ситуації

Окрім візуального відображення, сценарії GIS-аналітики може бути експортовано в базу даних сценаріїв у форматів XML, XPDЛ. Нижче наведено фрагмент тексту моделі сценарію <Аналіз запитів>, який експортовано в файл формату XPDЛ (XML Process Definition Language), який згенеровано в середовищі Bizagi Modeler.

Ця мова використовується для визначення описів графічних елементів BPMN-діаграм та обміну виконуваних процесів між різними інструментами моделювання та наборами процедур управління в виконавчій системі:

<PackageHeader>

```

<XPDLVersion>2.2</XPDLVersion>
<Vendor>Bizagi Process Modeler.</Vendor>
<Created>2020-09-25T10:55:43.1350451+03:00</Created>
<ModificationDate>2020-11-
03T12:12:58.1038572+02:00</ModificationDate>
<Description>Сценарій GIS-аналітики</Description>
<Documentation />
</Pool>
<Pool Id="91f66491-9b5b-46a8-a26a-1891066044c7" Name="Експерт з
Надзвичайних ситуації " Process="ba9066e9-7d86-4138-b406-f5b04946c7a0"
BoundaryVisible="true">
  <Lanes />
  <NodeGraphicsInfos>
    <NodeGraphicsInfo ToolId="BizAgi_Process_Modeler" Height="337"
Width="1058" BorderColor="-16777216" FillColor="-1">
      <Coordinates XCoordinate="31" YCoordinate="4" />
      <TextDirection xsi:nil="true" />
    </NodeGraphicsInfo>
  </NodeGraphicsInfos>
</Pool>
<Pool Id="2a6fab97-f128-46dc-99c0-aea7a1a4fe7c" Name="Експерт
картограф " Process="af24eebd-5b22-4024-b967-7ac682379fd0"
BoundaryVisible="true">
  <Lanes />
  <NodeGraphicsInfos>
    <NodeGraphicsInfo ToolId="BizAgi_Process_Modeler" Height="280"
Width="1060" BorderColor="-16777216" FillColor="-1">
      <Coordinates XCoordinate="33" YCoordinate="379" />
      <TextDirection xsi:nil="true" />
    </NodeGraphicsInfo>
  </NodeGraphicsInfos>
</Pool>
<Pool Id="dfa555f5-f275-4076-997a-17b163c35ed6" Name="Експерт
гідролог " Process="080dad41-5b2d-4891-be73-c49502e6ce21"
BoundaryVisible="true">
  <Lanes />
  <NodeGraphicsInfos>
    <NodeGraphicsInfo ToolId="BizAgi_Process_Modeler" Height="272"
Width="1063" BorderColor="-16777216" FillColor="-1">
      <Coordinates XCoordinate="33" YCoordinate="689" />

```

Г.2 Виконання сценарію просторового моделювання наслідків надзвичайної ситуації

Виконання сценарію розглянемо на прикладі просторового моделювання ризиків повинні в басейні річки Тиса. На території України басейн річки Тиса повністю розташований в межах однієї області – Закарпатської. Площа басейну Тиса в межах України складає близько 12,8 тис. км² (рис. 3). Більшість території басейну знаходиться в горах та передгір'ї Карпат, решта – на Угорській рівнині. Середні багаторічні значення кількості атмосферних опадів становлять від 870 мм (Великий Березний – передгірська частина) до 1600 мм (Плай – середньогір'я). На навітряних схилах гір кількість опадів може досягати 1100-1200 мм за рік. Живлення водоносного комплексу відбувається за рахунок атмосферних опадів і поверхневих вод, що дуже часто призводить до ризику виникнення повинні.

Відповідно до загального сценарію просторового моделювання гідрологічної ситуації експерт з НС отримує метеорологічний прогноз даного регіону. Метеорологічний радар надає інформацію про стан атмосфери над Карпатами і прилеглими територіями (місцезнаходження і стан дощових хмар, їх висота, інтенсивність опадів та інше). Дані метеорологічного прогнозу уточнюються та доповнюються поточними даними з гідропостів протиповіневого моніторингу стану рівнів річок басейну (Рисунок Г.5). Експерт з НС спільно з гідрологом аналізує дані рівня опадів, а також прогнозовані рівні отримані з гідропостів з метою визначення ймовірності виникнення паводкової ситуації (Рисунок Г.6). Для класифікації категорії та масштабу паводкової ситуації використовуються класифікатор з визначеними критичними рівнями – рівнями історичних паводків:

- 20% рівень – паводки що відбувалися 1 раз в 5 років;
- 10% рівень – паводки що відбувались 1 раз в 10 років;
- 5% рівень - паводки що відбувались 1 раз в 20 років (сильні паводки);

– 1% рівень – паводки що відбувались 1 раз в сто років (дуже сильні паводки).

Якщо в процесі попередньої оцінки метеорологічних даних про небезпечні прояви погоди (грози, шквали, хмарність, опади) та даних з гідропостів виявляється висока вірогідність виникнення паводкової ситуації, експерт з НС приймає рішення щодо запуску процесу просторового моделювання наслідків НС.

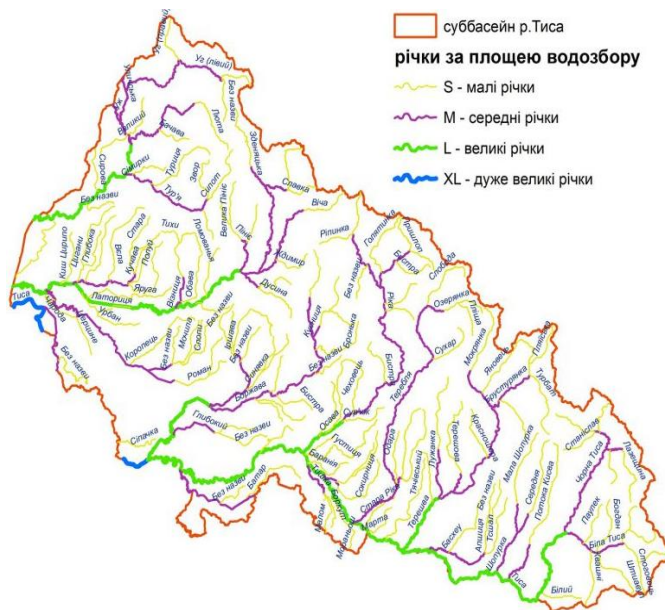


Рисунок Г.5 – Карта басейну Тиса в межах України

Рівні на гідропостах Редагувати дані гідропостів

Річка	Гідропост	№ гідропоста	Рівень
Тиса	Чорна Тиса-Ясіня	1	400
Тиса	Тиса-Рахів	2	527
Тиса	Вел.Бичків	3	552
Тиса	Тячів	4	693
Тиса	Тиса-Хуст	5	487
Тиса	Вилок	6	695
Тиса	Вашарошнамень	7	932
Тиса	Тиса-Чоп	8	1315
Біла Тиса	Лути	9	195
Тересва	Руська-Мокра	10	296
Тересва	Усть-Чорна	11	374
Тересва	Нересниця	12	332
Теребля	Колочава	13	363
Ріка	Верхній Бистрий	14	397
Ріка	Міжпр'я	15	428
Ріка	Нижній Бистрий	16	362
Ріка	Ріка-Хуст	17	669
Боржава	Довге	18	535

Рисунок Г.6 – Поточні дані з гідропостів

Дуже важливим та відповідальним кроком сценарію GIS-аналітики є актуалізація цифрової моделі території. Актуалізація цифрової моделі пов'язана з оновленням та підтримкою достовірних даних, своєчасним виявленням змін територій: забруднення річок, появи нових споруд та забудов, доріг, зміни ландшафту тощо.

Зрозуміло, що якість прогнозного моделювання (особливо паводкової ситуації) цілком залежить від точності 3D моделі рельєфу басейну річок, де саме відбувається моделювання. Тому для актуалізації 3D моделі територій обов'язково залучається кваліфікований експерт картограф, який використовує ліцензійні інструменти ArcGIS Desktop – ArcGIS 3D Analyst, ArcGIS

Geostatistical Analyst та інших для тривимірного відображення та редагування даних. За допомогою цих інструментів картограф формує точну 3D модель рельєфу території (Рисунок Г.7), використовуючи дані дистанційного зондування, дані топоїомки або повітряного спостереження, якщо картографічні дані застарілі.

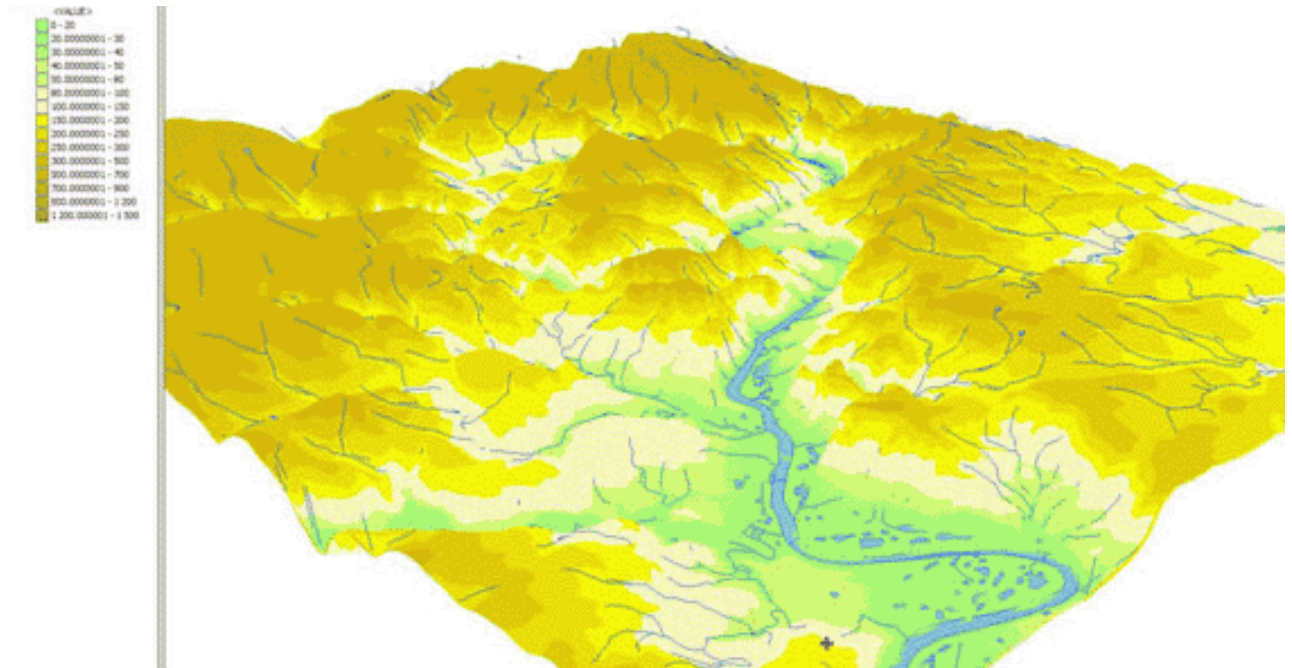


Рисунок Г.7 – Тривимірна модель рельєфу басейну з накладенням векторних шарів річок

Результатом роботи картографа є оновлена цифрова 3D-модель масштабу 1:200 000, яка відображує рельєф досліджуваної території на час повинні та може використовуватися для просторового моделювання розвитку паводкової ситуації.

Г.3 Гідрологічне моделювання

Експерт гідролог на основі одержаного прогнозу повинен прийняти рішення про рівень історичного паводку і виконати розрахунок рівнів води на поперечниках. При нечіткому прогнозі експерт може прорахувати найменш небезпечний прогноз і найбільш небезпечний прогноз. При обмеженій кількості гідропостів використовуються евристичні моделі інтегровані в ArcGis, така модель використовується для рік басейну Тиси. Для басейну Дністра де достатня

кількість автоматичних гідропостів використовуються моделі побудовані на вирішенні диференційних рівнянь реалізовані вітчизняними математиками і гідрологами, або пакети зарубіжних математиків і програмістів такі як МІКЕ II, HECRAS.

Результатом гідрологічного моделювання є наступна функція:

$$H_{\text{пор}} = F\{H_{\text{гп}}\},$$

де $H_{\text{пор}}$ – множина значень рівнів на поперечниках, $H_{\text{гп}}$ – рівні на гідропостах.

Метою інтерпретації результатів гідрологічного моделювання є одержання водної поверхні в вигляді полігону який в подальшому буде вхідними даними для аналізу наслідків НС. Такий полігон можна отримати шляхом:

- побудови триангуляційної моделі вхідними даними, для якої є рівні на поперечниках,
- перетворення триангуляційної моделі в растрову модель поверхні,
- одержання растрової поверхні затоплення шляхом співставлення растрової моделі рельєфу та поверхні води,
- перетворення растрової поверхні затоплення в полігон.

Г.4 Просторове моделювання

Вхідними даними для просторового моделювання ризиків паводкової ситуації є триангуляційна модель поверхні води і модель рельєфу басейну Тиси та поверхні води, а саме:

- базова електронна карта території України масштабу 1:200 000;
- цифрова 3D-модель рельєфу досліджуваної території;
- профілі русла річок басейну Тиси;
- точкові дані про очікувані або наявні рівні води в річках басейну – за даними офіційних гідрологічних прогнозів, результатами натурних спостережень на гідропостах, гідрологічного одномірного моделювання, статистичними даними про історичні максимуми тощо.

Використання карти території України масштабу 1:200 000 дає можливість автоматично формувати необхідне просторове оточення паводкової ситуації, що включає інформацію про положення водних об'єктів, доріг, інженерних комунікацій, населених пунктів, рельєфу, промислових підприємств тощо, які оточують місце виникнення НС.

При просторовому моделюванні паводкової ситуації басейну Тиси були прийняті реальні значення допустимих рівнів підйому води, отримані з УкрГМЦ по гідропостах на підставі даних спостереження за попередні роки. Для отримання значень рівня в поперечниках з вказаними параметрами проведено розрахунок значень для інтерполяції даних вздовж русла ріки через кожні 5 км. На підставі цих розрахунків засобами Spatial Analyst ArcGIS була побудована статистична поверхня гіпотетичного рівня води при піднятті на 5 м, яка у перетині з поверхнею землі створила полігони зон можливого затоплення.

Блок просторового моделювання функціонує з використанням GIS-інтерфейсу, який забезпечує:

- створення геоінформаційного середовища, яке містить перелік картографічних шарів, що необхідні для функціонування комплексу за даними картографічного фонду;
- позиціонування карти на вибрану територію;
- автоматичне введення та редагування параметрів, які необхідні для проведення моделювання, а саме:
 - поточних рівнів річок басейну Тиси на гідропостах за даними УкрГМЦ;
 - історичних максимумів по гідропостах;
 - введення або коригування даних по гідропостах експертом гідрологом;
 - вибір необхідної інформації з БД надзвичайних ситуацій у якості прецеденту, який стався в минулому, для порівняння;
 - перегляд, редагування та використання існуючих сценаріїв моделювання.

В результаті моделювання з'являється зона затоплення території, яка зберігається в просторовій БД та використовується у подальшому для аналізу наслідків і оцінки ризиків повинні (Рисунок Г.8).

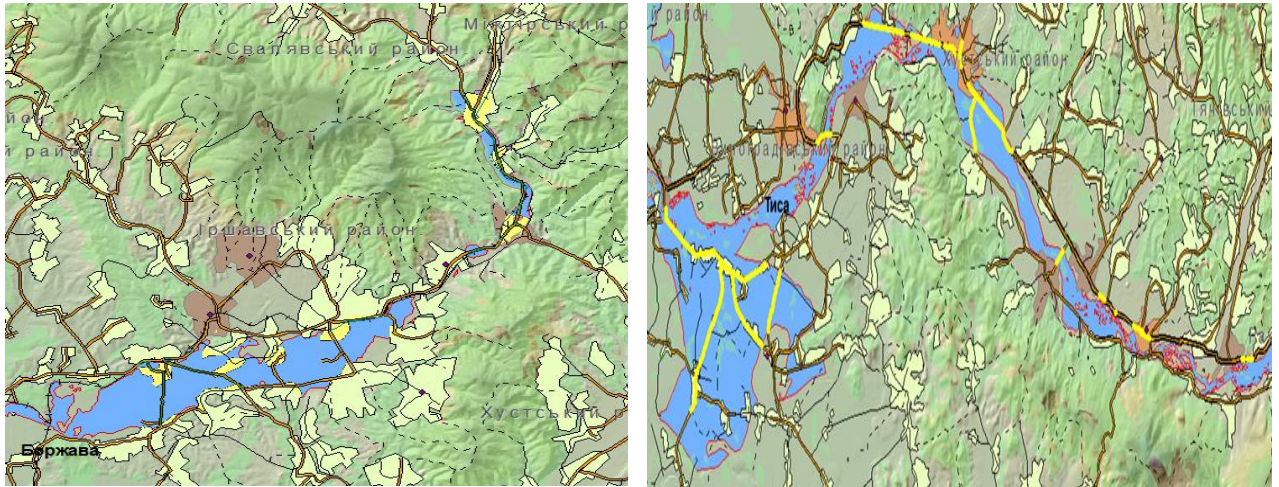


Рисунок Г.8 – Приклади відображення поверхня затоплення, одержані на підставі розрахунків засобами Spatial Analyst ArcGIS

Компонента ArcGIS Desktop також використовується для управління БД, аналізу наслідків повеневих ситуацій та редагування карт території.

Відповідно до технології просторового моделювання, зона затоплення території формується у вигляді полігону та може відображується на картографічному інтерфейсі GIS ArcView 8.1. (Рисунок Г.9).

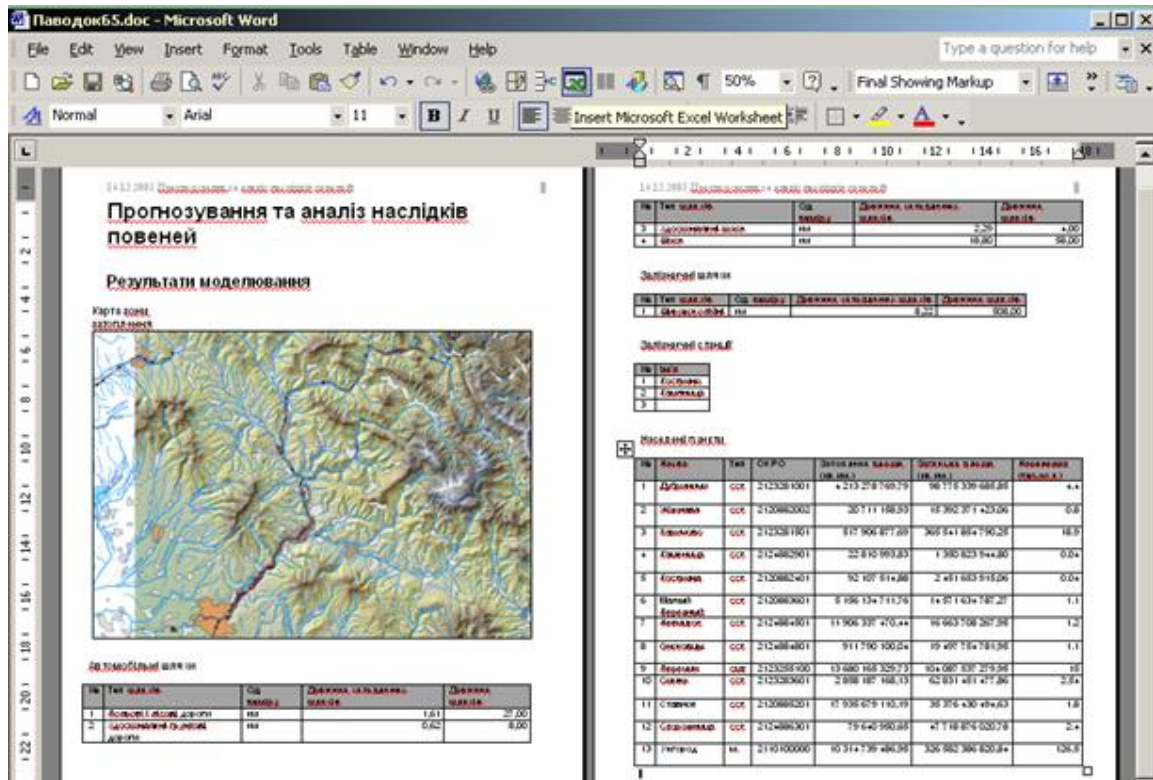


Рисунок Г.9 – Приклад публікації результатів просторового моделювання

Процес аналізу ймовірних наслідків НС завершується публікацією результатів на Web-порталі або у формат MS Office з автоматичним формуванням карти зони НС із різними шарами:

- зон затоплення;
- населених пунктів;
- промислових підприємств;
- залізничних станцій;
- потенційно небезпечних об'єктів;
- автомобільних та залізничних шляхів;
- інженерних комунікацій; водних об'єктів;
- рослинного покриву.

Для створення форм звітності, придатних для підготовки рішень, результати просторового моделювання можуть бути оформлені у вигляді зведених таблиць (таблиця Г.1).

Автомобільні шляхи		Залізничні станції		Залізничні шляхи		Населені пункти		Небезпечні об'єкти	
№	ОКРО	Назва	Тип	Затоплена площа (кв. м.)	Загальна площа (кв. м.)	Насе			
1	2123281001	Дубриничи	сст	1 053 319 692,45	24 693 834 921,46				
2	2120882002	Жорнава	сст	20 711 158,93	15 392 371 423,06				
3	2123281501	Заричово	сст	73 986 696,81	52 220 264 970,04				
4	2124882901	Каменица	сст	22 810 993,83	1 350 823 944,80				
5	2120882401	Кострина	сст	46 053 757,44	1 225 826 957,53				
6	2120883601	Малый	сст	5 156 134 711,76	14 571 634 787,27				
7	2124884501	Невицкое	сст	11 906 337 470,44	16 663 708 267,95				
8	2124884801	Оноковцы	сст	911 790 100,04	19 497 754 781,95				
9	2123255100	Перечин	сст	6 840 082 664,86	52 043 768 639,97				
10	2123283601	Симер	сст	952 729 056,04	20 943 817 159,29				
11	2120885201	Ставное	сст	17 935 679 110,19	35 376 430 494,63				
12	2124886301	Сторожница	сст	39 820 475,32	23 859 438 010,39				
13	2110100000	Ужгород	м.	10 314 739 486,95	326 582 386 820,84				

Окрім публікації на Web-порталі або у форматі MS Office результати просторового моделювання зберігаються у вигляді таблиць оперативної БД.

Запропонований підхід до моделювання сценаріїв аналітики на основі графічних методів нотації BPMN та OWL-моделей показав принципову можливість порівняння моделей сценаріїв АнД, створених в нотації BPMN-моделі, з конвертованою OWL-моделлю сценарію. Для спрощення самого процесу формування сценаріїв залучаються графічна нотація BPMN 2 [13], яка дозволяє не тільки моделювати сценарії АнД у вигляді BPMN-діаграм, але й серіалізувати в XML-файл для відпрацювання в комп'ютерному середовищі із залученням стандартного ПЗ BPM-систем. Завдяки конвертації BPMN-моделі, OWL-модель сценарію АнД, фактично, перетворюється в базу знань організації, яка може розширюватися за рахунок додавання нових сценаріїв АнД.

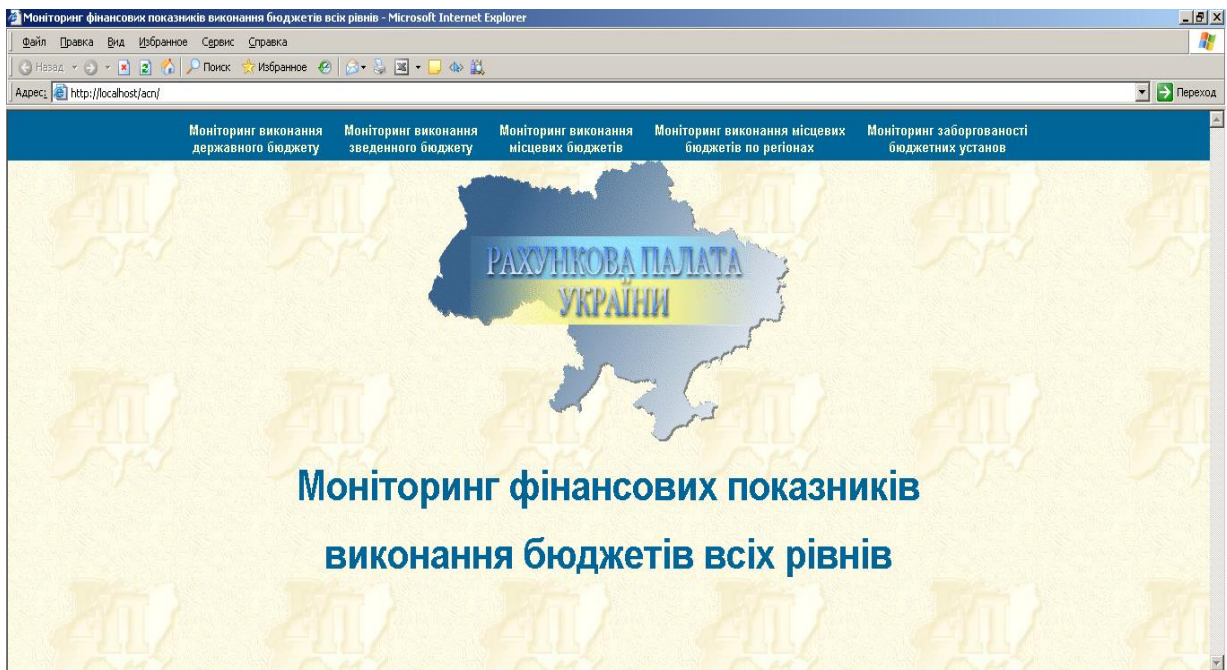
Отримана OWL-модель сценарію АнД може досліджуватися в інструментальному середовищі будь-якого зі стандартних редакторів онтології, зокрема платформою Protégé 5. Залучаючи Web-технології, розробник сценарію АнД має можливість більш детально візуалізувати OWL-моделі, а також проводити семантичний аналіз коректності логічно-зв'язаних фрагментів сценарію (особливо багаторівневих) використовуючи інструментарій редакторів онтології.

Запропонований підхід дає низку переваг – починаючи з суттєвого спрощення самого процесу моделювання та верифікації сценаріїв в інтегрованому середовищі BPMN–OWL і, закінчуючи, реалізацією моделі в середовищі стандартних BPM-систем.

ДОДАТОК Д Моніторинг фінансових показників виконання бюджетів всіх рівнів

Д1. Фрагмент програмного додатку конфігурації інтелектуальної системи «Моніторинг фінансових показників виконання бюджетів всіх рівнів»

Головна сторінка програмного забезпечення «Моніторинг фінансових показників виконання бюджетів всіх рівнів»:



```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

Note: As an alternative to hand editing this file you can use the web admin tool to configure settings for your application. Use the Website->Asp.Net Configuration option in Visual Studio. A full list of settings and comments can be found in machine.config.comments usually located in

```
\\Windows\Microsoft.Net\Framework\v2.x\Config
```

```
-->
```

```
<configuration>
```

```
  <system.web>
```

```
    <!--
```

Set compilation debug="true" to insert debugging symbols into the compiled page. Because this affects performance, set this value to true only during development.

```
-->
```

```
    <globalization culture="uk-UA" fileEncoding="utf-8" uiCulture="uk-UA" />
```

```
    <compilation debug="true" defaultLanguage="c#" targetFramework="4.0">
```

```

    <assemblies>
        <add assembly="Microsoft.JScript, Version=10.0.0.0, Culture=neutral,
        PublicKeyToken=b03f5f7f11d50a3a" />
        <add assembly="Microsoft.Vsa, Version=8.0.0.0, Culture=neutral,
        PublicKeyToken=b03f5f7f11d50a3a" />
        <add assembly="WindowsBase, Version=4.0.0.0, Culture=neutral,
        PublicKeyToken=31bf3856ad364e35" />
    </assemblies>
</compilation>
<!--

```

The <authentication> section enables configuration of the security authentication mode used by ASP.NET to identify an incoming user.

```
-->
```

```

    <authentication mode="Windows" />
<!--

```

The <customErrors> section enables configuration of what to do if/when an unhandled error occurs during the execution of a request. Specifically,

it enables developers to configure html error pages to be displayed in place of a error stack trace.

```

<customErrors mode="RemoteOnly" defaultRedirect="GenericErrorPage.htm">
    <error statusCode="403" redirect="NoAccess.htm" />
    <error statusCode="404" redirect="FileNotFound.htm" />
</customErrors>
-->

```

```

    <customErrors mode="Off" />
    <httpHandlers>
        <add verb="POST" path="HighchartsExport.axd"
type="Tek4.Highcharts.Exporting.HttpHandler, Tek4.Highcharts.Exporting" />
    </httpHandlers>
    <httpRuntime requestValidationMode="2.0" />
    <pages controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID" /></system.web>
    <connectionStrings>
        <remove name="LocalSqlServer" />
        <add connectionString="Data Source=acn;Persist Security Info=True;User
ID=vgaidar;Password=ADMFEPO" name="OraConn" />
    </connectionStrings>

```

```

<system.webServer>
  <staticContent>
    <remove fileExtension=".xml" />
    <mimeTypeMap fileExtension=".xml" mimeType="text/xml; charset=utf-8" />
  <mimeTypeMap fileExtension=".json" mimeType="application/json" />
</staticContent>
  <handlers>
    <add name="*.vbhtml_*" path="*.vbhtml" verb="*" type="System.Web.HttpForbiddenHandler"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.vbhtml_*" path="*.vbhtml" verb="*" type="System.Web.HttpForbiddenHandler"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.cshtml_*" path="*.cshtml" verb="*" type="System.Web.HttpForbiddenHandler"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.cshtml_*" path="*.cshtml" verb="*" type="System.Web.HttpForbiddenHandler"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.aspx_*" path="*.aspx" verb="*" type="System.Web.HttpForbiddenHandler"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.xaml_*" path="*.xaml" verb="*" type="System.Xaml.Hosting.XamlHttpHandlerFactory,
System.Xaml.Hosting, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.xoml_*" path="*.xoml" verb="*" type="System.ServiceModel.Activation.HttpHandler,
System.ServiceModel.Activation, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.svc_*" path="*.svc" verb="*" type="System.ServiceModel.Activation.HttpHandler,
System.ServiceModel.Activation, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.soap_*" path="*.soap" verb="*"
type="System.Runtime.Remoting.Channels.Http.HttpRemotingHandlerFactory, System.Runtime.Remoting,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.rem_*" path="*.rem" verb="*"
type="System.Runtime.Remoting.Channels.Http.HttpRemotingHandlerFactory, System.Runtime.Remoting,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.asmx_*" path="*.asmx" verb="*" type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
preCondition="integratedMode, runtimeVersionv2.0" />
    <add name="ScriptResource.axd_GET,HEAD" path="ScriptResource.axd" verb="GET,HEAD"
type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" preCondition="integratedMode, runtimeVersionv2.0"
/>
    <add name="*_AppService.axd_*" path="*_AppService.axd" verb="*"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" preCondition="integratedMode, runtimeVersionv2.0"
/>

```

Д2. Фрагмент складу базових модулів інтелектуальної системи «Моніторинг фінансових показників виконання бюджетів всіх рівнів»

FILELIST

```
<?xml version="1.0" encoding="utf-8" ?>
  <DOCUMENT>
    <FILELIST>
      <!-- National -->
        <FILE budget="national" type="incoming" fund="common"
chart_file="national/incoming/common/chart.xml"
param_file="national/incoming/common/param.xml"
view_file="national/incoming/common/view.xml" view_layer="group1"/>
        <FILE budget="national" type="incoming" fund="special"
chart_file="national/incoming/special/chart.xml"
param_file="national/incoming/special/param.xml"
view_file="national/incoming/special/view.xml" view_layer="group1"/>
        <FILE budget="national" type="incoming" fund="united"
chart_file="national/incoming/united/chart.xml"
param_file="national/incoming/united/param.xml"
view_file="national/incoming/united/view.xml" view_layer="group1"/>
        <FILE budget="national" type="outgoing" fund="common"
chart_file="national/outgoing/common/chart.xml"
param_file="national/outgoing/common/param.xml"
view_file="national/outgoing/common/view.xml" view_layer="group1"/>
        <FILE budget="national" type="outgoing" fund="special"
chart_file="national/outgoing/special/chart.xml"
param_file="national/outgoing/special/param.xml"
view_file="national/outgoing/special/view.xml" view_layer="group1"/>
        <FILE budget="national" type="outgoing" fund="united"
chart_file="national/outgoing/united/chart.xml"
...
      <!-- Regions -->
        <FILE budget="regions" type="incoming" fund="common"
chart_file="regions/incoming/common/chart.xml"
param_file="regions/incoming/common/param.xml"
view_file="regions/incoming/common/view.xml" view_layer="group1"/>
        <FILE budget="regions" type="incoming" fund="special"
chart_file="regions/incoming/special/chart.xml"
param_file="regions/incoming/special/param.xml"
view_file="regions/incoming/special/view.xml" view_layer="group1"/>
        <FILE budget="regions" type="incoming" fund="united"
chart_file="regions/incoming/united/chart.xml"
param_file="regions/incoming/united/param.xml" view_file="regions/incoming/united/view.xml"
view_layer="group1"/>
        <FILE budget="regions" type="outgoing" fund="common"
chart_file="regions/outgoing/common/chart.xml"
```

```

param_file="regions/outgoing/common/param.xml"
view_file="regions/outgoing/common/view.xml" view_layer="group1"/>

    <FILE budget="regions" type="outgoing" fund="special"
chart_file="regions/outgoing/special/chart.xml"
param_file="regions/outgoing/special/param.xml"
view_file="regions/outgoing/special/view.xml" view_layer="group1"/>
...
<!-- Borg -->

    <FILE budget="borg" type="kek_deb" fund="common"
chart_file="borg/kek_deb/common/chart.xml" param_file="borg/kek_deb/common/param.xml"
view_file="borg/kek_deb/common/view.xml" view_layer="group1"/>

    <FILE budget="borg" type="kek_deb" fund="special"
chart_file="borg/kek_deb/special/chart.xml" param_file="borg/kek_deb/special/param.xml"
view_file="borg/kek_deb/special/view.xml" view_layer="group1"/>

    <FILE budget="borg" type="kek_deb" fund="united"
chart_file="borg/kek_deb/united/chart.xml" param_file="borg/kek_deb/united/param.xml"
view_file="borg/kek_deb/united/view.xml" view_layer="group1"/>

    <FILE budget="borg" type="kpk_deb" fund="common"
chart_file="borg/kpk_deb/common/chart.xml" param_file="borg/kpk_deb/common/param.xml"
view_file="borg/kpk_deb/common/view.xml" view_layer="group1"/>

    <FILE budget="borg" type="kpk_deb" fund="special"
chart_file="borg/kpk_deb/special/chart.xml" param_file="borg/kpk_deb/special/param.xml"
view_file="borg/kpk_deb/special/view.xml" view_layer="group1"/>

    <FILE budget="borg" type="kpk_deb" fund="united"
chart_file="borg/kpk_deb/united/chart.xml" param_file="borg/kpk_deb/united/param.xml"
view_file="borg/kpk_deb/united/view.xml" view_layer="group1"/>

    <FILE budget="borg" type="kek_kred" fund="common"
chart_file="borg/kek_kred/common/chart.xml" param_file="borg/kek_kred/common/param.xml"
view_file="borg/kek_kred/common/view.xml" view_layer="group1"/>
...
</FILELIST>
</DOCUMENT>

```

Д3. Фрагмент програмного додатку формування сценаріїв аналітичної діяльності для системи «Моніторингу фінансових показників виконання бюджетів всіх рівнів»

Програма налаштування сценаріїв аналітичної діяльності з урахуванням особливостей системи «Моніторингу фінансових показників виконання бюджетів всіх рівнів» впроваджена в Рахунковій палаті України та наведена нижче:

```
<?xml version="1.0" encoding="UTF-8"?>
```



```
<!--
```

IIS configuration sections.

For schema documentation, see %windir%\system32\inetsrv\config\schema\IIS_schema.xml.

Please make a backup of this file before making any changes to it.

```
-->
```

```
<configuration>
```

```
<!--
```

The <configSections> section controls the registration of sections. Section is the basic unit of deployment, locking, searching and containment for configuration settings.

Every section belongs to one section group.
A section group is a container of logically-related sections.

Sections cannot be nested.
Section groups may be nested.

```
<section
  name="" [Required, Collection Key] [XML name of the section]
  allowDefinition="Everywhere" [MachineOnly|MachineToApplication|AppHostOnly|Everywhere]
[Level where it can be set]
  overrideModeDefault="Allow" [Allow|Deny] [Default delegation mode]
  allowLocation="true" [true|false] [Allowed in location tags]
/>
```

The recommended way to unlock sections is by using a location tag:

```
<location path="Default Web Site" overrideMode="Allow">
  <system.webServer>
    <asp />
  </system.webServer>
</location>
```

```
-->
```

```
<configSections>
  <sectionGroup name="system.applicationHost">
    <section name="applicationPools" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
    <section name="configHistory" allowDefinition="AppHostOnly" overrideModeDefault="Deny"
/>
    <section name="customMetadata" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
    <section name="listenerAdapters" allowDefinition="AppHostOnly" overrideModeDefault="Deny"
/>
    <section name="log" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
```

```

<section name="serviceAutoStartProviders" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
<section name="sites" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />
<section name="webLimits" allowDefinition="AppHostOnly" overrideModeDefault="Deny" />

```

Д4. Програмні засоби виконання типового сценарію автоматизованого вводу звітності до БД системи «Моніторингу фінансових показників виконання бюджетів всіх рівнів»

Програмні засоби призначені для автоматизованого вводу звітності Державної казначейської служби України до БД фінансово-економічних показників інформаційно-аналітичної системи РП до СУБД Oracle 11g без застосування ручних процедур.

Перевагою ПЗ автоматизованого введення казначейських звітів є суттєве скорочення часу. Завдяки цим засобам введення даних з обсягом понад 120000 записів/на місяць (рядків) у БД Oracle 11g виконується за 1 – 2 хвилини. Ручне введення такої кількості показників з наступною їх верифікацією потребує декілька робочих днів.

Для потокового введення даних в БД Oracle 11g використовуються процедури у вигляді макросів MS Excel. Система інтелектуального вводу автоматично налаштовуються на початок и кінець казначейських звітів, очищення даних від зайвих символів, виправлення помилок (включи й подвійні), перетворення даних у відповідні формати та ввід перетворених та очищених даних до таблиць БД Oracle 11g. Список макросів MS Excel, згрупований за змістовним принципом, наведено в таблиці Д2-1.

Таблиця Д2-1

Аркуш 1 «Звіти по доходах» містить макроси, налаштовані на автоматичне введення звітів про виконання бюджетів всіх рівнів за доходами до БД ФЕП:		
1.	Макрос введення звіту про виконання доходної частини державного бюджету	INPUT_INCOM
2.	Макрос введення зведеного звіту про виконання доходної частини місцевих бюджетів	INPUT_INCOMM
3.	Макрос введення звіту про виконання доходної частини зведеного бюджету	INPUT_INCOMZ
Аркуш 2 «Звіти по видатках» містить макроси, налаштовані на автоматичне введення звітів про виконання бюджетів всіх рівнів за видатками до БД ФЕП:		

4.	Макрос введення звіту про виконання видатків державного бюджету за КПК	INPUT_EXPENS
5.	Макрос введення звіту про виконання видатків державного бюджету за КЕКВ по загальному фонду	INPUT_ZAGKEK
6.	Макрос введення звіту про виконання видатків державного бюджету за КЕКВ по спеціальному фонду	INPUT_SPECKEK
7.	Макрос введення зведених звітів про виконання видатків державного, зведеного та місцевих бюджетів за КЕКВ	INPUT_BUDGKEK
8.	Макрос введення звітів про виконання видатків державного, зведеного та місцевих бюджетів за КФК	INPUT_BUDGKFK
9.	Макрос введення звітів про виконання захищених статей видатків ДБУ	INPUT_ZAHSTAT
Аркуш 3 «Звіти по МБ (регіони)» містить макроси, які налаштовані на автоматичне введення звітів про виконання місцевих бюджетів за доходами та видатками до БД ФЕП:		
10.	Макрос введення місячного звіту про виконання доходної частини місцевих бюджетів за регіонами	INPUT_INCOMREG
11.	Макрос введення місячного звіту про виконання видаткової частини місцевих бюджетів за регіонами	INPUT_EXPENSREG
12.	Макрос введення квартального звіту про виконання доходів та видатків місцевих бюджетів за регіонами	INPUT_KVARTREG
Аркуш 4 «Звіти про фінансування» містить макроси, які налаштовані на автоматичне введення звітів про фінансування державного, зведеного та місцевих бюджетів до БД ФЕП:		
13.	Макрос введення звітів про фінансування державного, зведеного та місцевих бюджетів за типом кредитора	INPUT_FINKRED
14.	Макрос введення звітів про фінансування державного, зведеного та місцевих бюджетів за типом боргового зобов'язання	INPUT_FINKRED
Аркуш 5 «Звіти про кредитування» містить макроси, налаштовані на автоматичне введення звітів про кредитування державного, зведеного та місцевих бюджетів:		
15.	Макрос введення звітів про кредитування Державного, Зведеного та місцевих бюджетів за функціональною класифікацією кредитування бюджету (за КФК) (NEW)	INPUT_FUNKKRED
16.	Макрос введення звіту про кредитування Державного бюджету за КПК (NEW)	INPUT_PROGKRED
17.	Макрос введення звітів про кредитування Державного, Зведеного та місцевих бюджетів за КБК (NEW)	INPUT_KREDDBU
Аркуш 6 «Звіти про заборгованість» (NEW) містить макроси, налаштовані на автоматичне введення звітів про заборгованість бюджетних установ		

18.	Макрос введення звітів про заборгованість бюджетних установ за КПК файл F73Zrmm.xls	INPUT_ZABBUST
19.	Макрос попередньої ДІАГНОСТИКИ звіту про заборгованість за КПК файл V7DZrmm.xls	DIAG_ZABBUSTZF
20.	Макрос введення звітів заборгованість за КПК (фінансові зобов'язання) файл V7DZrmm.xls	INPUT_ZABFINZO

ДОДАТОК Е. Стандарти та засоби процесного моделювання

Е.1 Огляд сучасних засобів моделювання бізнес-процесів

На даний час на ринку ІТ-технології сформувалися і використовуються декілька стандартів опису бізнес-процесів це BPMN, UML, BPEL, XPDL, WS-CDL, JPDL, Xlang, BPMML, WSFL, WSCL, BPSS, WSCI. В більшості, ці стандарти підтримують тенденцію розвитку технологій компонентно-орієнтованого програмування. Але перехід ІТ-систем до парадигми сервіс-орієнтованої архітектури (SOA) вимусив бізнес аналітиків звертати увагу саме на стандарти та засоби моделювання і опису процесів, які реалізують веб-сервіси бізнес-процесів. Виходячи з цього, список стандартів та мов моделювання скочується до BPMML (BPMN), JPDL (Java Process Definition Language), WSFL, WSCL, BPSS, WSCI. Тому огляд сучасних засобів моделювання бізнес-процесів проводився саме виходячи з міркувань підтримки цими програмними платформами SOA. За результатами попереднього аналізу для порівняння відібрані найбільш відомі засоби моделювання бізнес-процесів, які потенційно можуть використовуватися для опису багаторівневих сценаріїв Анд. Отже, до цього списку входять: графічний Modeler від Software company Bizagi [Bizagi], WebSphere Business Modeler від компанії IBM [Advanced V7.0], візуальний дизайнер Bonita Studio від Bonita BPM [], Hyperion Business Modelling від компанії Oracle [], Process Modeler від ERWin [], дизайнер ARIS Express від компанії Software AG[].

Слід нагадати, що існуючі платформи використовують різні мови моделювання, термінології, нотації, застосування яких потребує набуття відповідних навиків та підготовки.. У якості критеріїв для оцінки інструментальних засобів моделювання бізнес-процесів застосувалися наступні: В таблиці наведено найбільш відомих засобів, які підтримують парадигму графічного моделювання бізнес-процесів.

1. Підтримка сучасних стандартів та мов проектування – тобто можливість використання стандартних мов та нотацій для опису бізнес-процесів, включаючи й нотацію BPMN 2.0 та XML-мову, а також мову BPMML виконання

бізнес-процесів у середовищі BPMS-IT – систем, сумісність мов проектування. Цей критерій також визначає доступність компонентів платформи як в настільних, так і в онлайн-версіях, Web based, Client/server, Online, наявність Cloud версії.

2. Інтеграція з іншими корпоративними додатками – критерій, який визначає наявність інструментарію взаємодії діаграм процесів з іншими рівнями інтегрованої платформи, які забезпечують не тільки моделювання процесів у графічному редакторі платформи, але й виконання процесів у BPMS-IT – системі. Крім того, обов'язково розглядається взаємодія з зовнішніми додатками за допомогою мови XPDL (XML Process Definition Language), яка є міжнародним стандартом обміну моделями процесів між різними інформаційними системами. Цей критерій також визначає нарощування функціональності платформи за рахунок модульної організації її ПЗ, тобто вертикальної інтеграції платформи.

3. Експорт/імпорт в зовнішні файли – критерій який визначає сумісність форматів представлення та публікації результатів моделювання з іншими платформами. Цей критерій також наявність механізмів імпорту та експорту моделі бізнес-процесу в інших мовах: MS Word, MS Excel, BPMN, TXT, Web (HTML), RTF тощо.

4. Можливість динамічної зміни бізнес-процесу – найважливіший критерій при аналізі інструментальних засобів, який враховує такі важливі аспекти як: можливість динамічної зміни та корегування діаграми та модернізацію процесу, як реакцію на виклики часу, а також динамічне перепрограмування виконавчих модулів; наявність технологічних рішень, що підтримують реалізацію змін процесів.

5. Простота використання – критерій, який визначає зручність інтерфейсу користувача та можливість візуального проектування на основі графічного редактора, наявність інтуїтивно-зрозумілого інтерфейсу користувача, повноту опису засобів моделювання, наявність методичних

матеріалів, які сприяють впровадженню методології проектування, а також навчання користувачів, бажано візуальних.

6. Виробник і ліцензія – назва розробника засобу та термін його присутності на ринку технологій, умови розповсюдження, технічна підтримка інструментальних засобів, спрямованість на певну ПрО або універсальність, а також джерела фінансування проекту.

7. Коротко зупинимось на засобах та платформах моделювання бізнес-процесів, наведених у таблиці E1.

8. Аналіз показує, що критерію «Підтримка сучасних стандартів та мов проектування» таких як BPMN 2.0, XPDЛ, BPEL відповідає більшість платформ. Більш того, інструментальні засоби мають або настільні версії, або Web-версії, що дозволяє функціонування редакторів моделювання як в онлайн режимі, так і в автономному. Але такі платформи як IBM WebSphere, BPwin або занадто перевантажені стандартами типу ITIL, TEAF/FEAF.

9. За критерієм «Інтеграція з іншими системами та додатками» майже всі розглянуті платформи моделювання Bizagi, Bonita BPM, Oracle Hyperion, ARIS демонструють високий рівень інтеграції – тобто наявність інструментарію взаємодії діаграм процесів з іншими рівнями інтегрованої платформи. Так, наприклад, платформа IBM WebSphere поєднує в собі такі засоби як WebSphere Business Modeler, Websphere Business Events, WebSphere Commerce, що забезпечує як моделювання процесу, так і його виконання в одному середовищі.

10. Критерію «Експорт/імпорт в зовнішні файли» в обсягу, який забезпечує сумісність форматів представлення та публікації результатів моделювання з іншими платформами, відповідають практично всі наведені в таблиці платформи.

11. Ще одним важливим фактором, який визначає успішність застосування засобів моделювання є, безумовно, можливість динамічної зміни бізнес-процесу. Як показує досвід Такому критерію – найважливіший критерій при аналізі інструментальних засобів, якій враховує такі важливі аспекти як:

можливість динамічної зміни та корегування діаграми та модернізацію процесу, як реакцію на виклики часу, а також динамічне перепрограмування виконавчих модулів; наявність технологічних рішень, що підтримують реалізацію змін процесів.

Для успішного впровадження та застосування інструментальних засобів дуже велика увага приділяється до критерію «Простота використання». Більшість розглянутих засобів оснащена графічними редакторами, засобами візуалізації та корегування моделей бізнес-процесів та властивостей елементів нотації BPMN 2.0. За цим критерієм найбільш відомими засобами є BizAgi Process Modeler [Bizagi] та Bonita Studio з інтуїтивно-зрозумілим та зручним інтерфейсом користувача, який дає можливість візуального проектування на основі графічного редактора. Ці засоби також пропонують наявність доступних та безкоштовних методичних матеріалів використання і навчання (в більшості візуальних) платформ моделювання.

І нарешті критерій «Виробник і ліцензія» - назва розробника, термін присутності на ринку технологій, умови розповсюдження, технічна підтримка. Цей критерій має дуже важливе місце в комплексній оцінці інструментальних засобів моделювання бізнес-процесів. За цим критерієм найбільш відомими інтегрованими платформами є Oracle Hyperion, IBM WebSphere, BPwin та ARIS, які пропонують комплексні рішення для BPM-систем. При прийнятті рішення про вибір конкретної платформи для моделювання бізнес-процесів слід оцінювати не тільки ціну поставки платформи, а й витрати на наповнення корисною інформацією; навчання персоналу; підтримку програмних засобів у робочому стані. З урахуванням цих чинників найбільш доцільним для багатьох компаній не використовувати складні і дорогі програмні продукти для моделювання бізнес-процесів, а моделювати за допомогою таких інструментальних засобів як BizAgi Process Modeler [Bizagi] або Bonita Studio [Bonita Studio], для яких ліцензія на рівні для моделювання є безкоштовною.

Таблиця Е1

Система	Підтримка стандартів та мов проектування	Інтеграція з іншими системами та додатками	Експорт/імпорт в зовнішні файли	Можливість динамічної зміни бізнес-процесу	Простота використання	Виробник і ліцензія
Bizagi Process Modeler	BPMN 2.0, XPDЛ, BPEL	<ul style="list-style-type: none"> • Modeler Services documentation • BizAgi Studio • BizAgi Engine • Bizagi Business Process Management Suite 	<ul style="list-style-type: none"> • MS Word • XPDЛ • BPM • Web (HTML) • BPMN • MS Visio 	За рахунок вертикальної інтеграції підтримується життєвий цикл процесів від моделювання до виконання та моніторингу	Зручний графічний drag & drop редактор нотації, який підтримує нотацію BPMN 2.0	BizAgi Corporation Великобританія Безкоштовна ліцензія на Bizagi Modeler на BizAgi Studio, BizAgi Engine – Платна
Bonita BPM – платформа розробки	BPMN 2.0, XPDЛ, jBPM, Eclipse	<ul style="list-style-type: none"> • Bonita Studio • Bonita BPM Engine • Bonita Portal Створення бізнес-додатків на основі робочого процесу	<ul style="list-style-type: none"> • PDF • XML реалізація скрипкових задач на Java 	За рахунок інтеграції додатків підтримується життєвий цикл ПЗ від моделювання до виконання та моніторингу	Графічний drag & drop редактор нотації BPMN	BonitaSoft, French National Institute for Research in Computer Scienc Платна, існує open source версія
Oracle Hyperion Business Modelling	Hyperion Performance Scorecard Oracle BPM Suite 11gR1	Інтеграція моделі з ERP-системою Інтеграція з реляційною СУБД	<ul style="list-style-type: none"> • MS Word • <input type="checkbox"/> MS Excel 	Підтримує життєвий цикл моделювання та виконання ERP процесів	Спеціальний Hyperion редактор ERP процесів	Oracle Corporation Ліцензія від \$700 на Hyperion Business Modelling
IBM WebSphere Business Modeler	BPMN 2.0, UML	WebSphere Business Events WebSphere Commerce WebSphere Business Modeler	<ul style="list-style-type: none"> • <input type="checkbox"/> MS Word • <input type="checkbox"/> PDF • <input type="checkbox"/> XML • <input type="checkbox"/> інші системи 	Підтримує життєвий цикл від моделювання до виконання та моніторингу БП	Графічний редактор моделювання, аналізу, документування	IBM (США) Ліцензія Basic від \$1500 до Advanced — \$11500

Система	Підтримка стандартів та мов проектування	Інтеграція з іншими системами та додатками	Експорт/імпорт в зовнішні файли	Можливість динамічної зміни бізнес-процесу	Простота використання	Виробник і ліцензія
BPwin - ERWin Process Modeler	IDEF0, IDEF3, DFD, Basic Flowchart, Cross Functional Flowchart BPMN, BPEL	Інтеграція з ERWin Model Manager, Paradigm Plus, ERWin Model Navigator, Arena	<ul style="list-style-type: none"> • MS Excel • TXT • HTML • RTF 	Підтримує цикл UML моделювання в нотації IDEF0, IDEF3, DFD BPMN, BPEL	Графічне моделювання в нотації IDEF0, IDEF3, DFD	Logic Works CA Technologies Ліцензія від \$1500
ARIS Express	EPC, UML, BPEL, BPMN, ITIL, TOGAF, DoDAF, TEAF/FEAF, ArchiMate та Zachman ARIS складає основу пакету Business Process Analysis Suite корпорации Oracle	ARIS BSC Portal SAP R/3, Oracle, AllFusion, ERStudio, Power Designer, OracleDesigner, Rational Rose Інтеграція з СУБД Oracle	<ul style="list-style-type: none"> • <input type="checkbox"/> MS Word • <input type="checkbox"/> MS Excel • <input type="checkbox"/> TXT • <input type="checkbox"/> HTML • <input type="checkbox"/> PDF • <input type="checkbox"/> XML 	ARIS BSC Підтримує життєвий цикл ПЗ від моделювання до виконання та моніторингу, так і подальшу реалізацію IT-рішень	Графічний редактор Інтелектуальні веб-технології	IDS Scheer AG (Німеччина) Ліцензія від 2600€. Безкоштовна ліцензія на моделер ARIS Express

Виходячи з проведеного аналізу сучасних інструментальних засобів моделювання бізнес-процесів (таблиця Е1), найбільш перспективними для використання у якості базового інструментарію визнано платформи BizAgi Process Modeler [Bizagi] та Bonita Studio [Bonita Studio]. Кожна з цих платформ реалізує повну функціональність та сучасні стандарти і мови моделювання BPMN 2.0, XPDL, BPEL. Більш того, кожна платформа має графічний редактор із зручним і інтуїтивно-зрозумілим інтерфейсом та є вертикально інтегрованою, що забезпечує підтримку життєвого циклу інформаційних систем згідно стандарту []. Також кожна платформа пропонує безкоштовні ліцензії на використання редакторів моделювання. Однак, за даними gartner magic quadrant 2019 for bpm [Magic Quadrant] платформа BizAgi, у порівнянні з Bonita Studio, вважається більш привабливою. Платформа BizAgi розташована на графіку Magic Quadrant навіть вище Oracle та входить до складу провідних виробників платформ BPMS. Крім того, BizAgi вважається лідером з інтелектуальної автоматизації процесів (Рисунок Е.1).

Отже, враховуючи вищенаведене BizAgi Process Modeler прийнято у якості базового інструментарію моделювання сценаріїв аналітичної діяльності.

Е.2 Платформа BizAgi, як засіб інтелектуального моделювання сценаріїв

Платформа **BizAgi** — це BPM-система, а також однойменна компанія з розробки програмного забезпечення, яка заснована в 1989 році. Штаб-квартира **BizAgi** розташована у Великобританії, а представництва – у США, Німеччині, Іспанії та Латинській Америці [Bizagi]. Назва BizAgi об'єднує два поняття "business" and "agility". ("бізнес" та "спритність").

Принциповою особливістю методології BPM, яка реалізована в **BizAgi**, є можливість моделювати сценарії за допомогою графічного редактора, що оперує з елементами нотації BPMN 2.0. Таки елементи мають стандартні позначення, певну семантику (Таблиця Е.1) та, як правило, однозначно

сприймаються людиною – як аналітиком, так і керівником. Для більш глибокого розуміння особливостей ПрО вхідні документи, необхідні для проектування, можуть бути доповнені антологіями ПрО у вигляді OWL-файлів. Це дає можливість проектування сценаріїв (алгоритмів процесів) безпосередньо аналітиком (фахівцем в ПрО), а засоби інтелектуалізації самого циклу проектування дозволяють отримувати виконувану модель процесу, навіть без участі програміста.



Рисунок Е.1 – Графік зі звіту “BPMS Gartner Magic Quadrant” від 30 січня 2019

Система VizAgі складається з трьох окремих засобів, які забезпечують існування всього життєвого циклу програмного забезпечення від розробки та

моделювання безпосередньо самого бізнес-процесу і до його функціонування в BPMS-IT – системі, включаючи й модернізацію процесів:

- BizAgi Process Modeler – повнофункціональне середовище в нотації BPMN (графічний редактор) для формування, моделювання та опису процесів у відповідних форматах;

- BizAgi Studio – середовище розробки бізнес-процесів ;

- BizAgi Engine – середовище виконання процесів, яка доступна користувачам в будь-якому браузері з будь-якого пристрою.

BizAgi Process Modeler є безкоштовним додатком для моделювання бізнес-процесів і процесної документації. Modeler дозволяє побудувати та візуалізувати діаграми і моделі процесів за стандартом нотації BPMN 2.0.

BizAgi Process Modeler пропонує такий функціонал:

- графічний редактор при роботі з нотацією BPMN 2.0.
- перевірка (валідація) створених моделей;
- автоматична генерація документів звітності;
- управління атрибутами елементів моделей;
- можливість додавати свої елементи в моделі;
- вивантаження моделі в графічному вигляді;
- мультімовна підтримка;
- можлива спільна робота над моделями.

Головною перевагою графічного редактора BizAgi Process Modeler є інтуїтивно зрозумілий інтерфейс, заснований на концепції «Drag & Drop», що дозволяє забезпечити швидке і легке формування діаграм бізнес-процесів з графічних елементів нотації BPMN 2.0.

Process Modeler Version 3.6 складається з п'яти основних панелей: <Toolbar>, <Ribbon>, <Palette>, <Element Properties> and <View> (Рисунок Е.2).

Панель <**Toolbar**> «Інструменти» містить команди швидкого доступу до підмножини будь-якого меню в Bizagi Modeler.

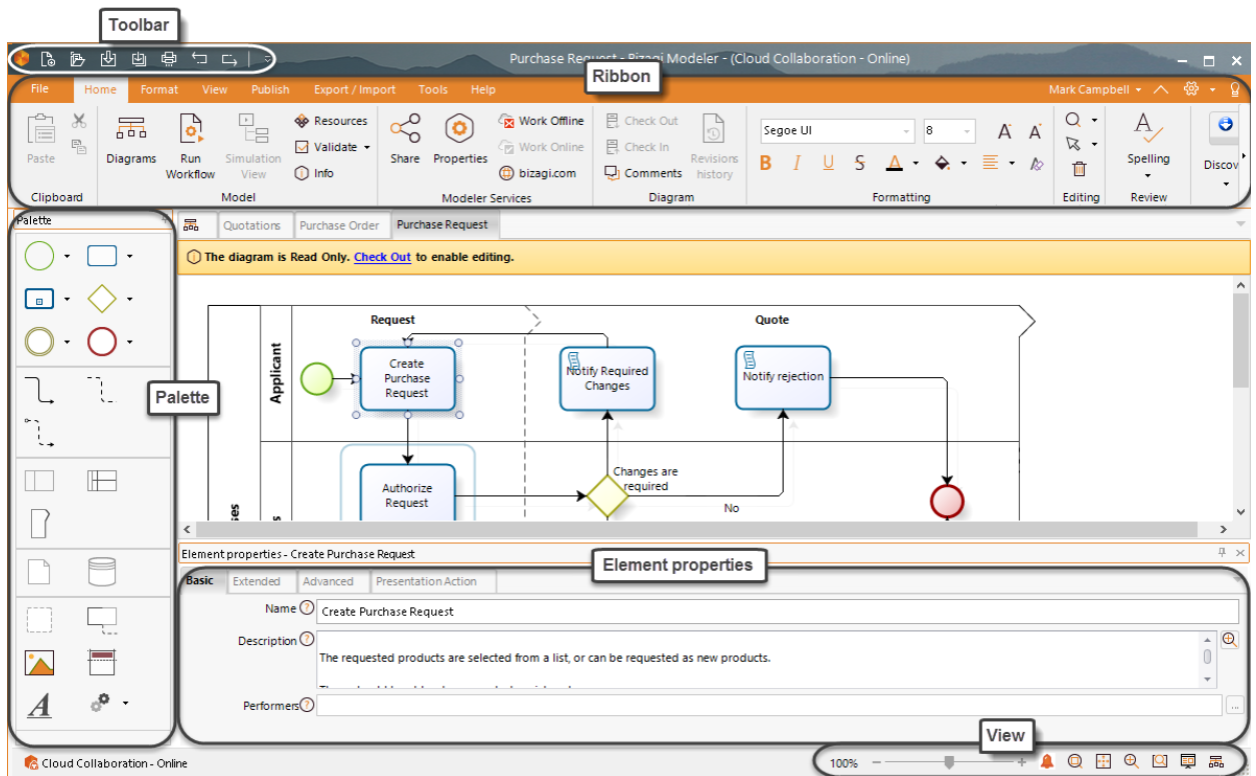


Рисунок Е.2 – Інтерфейс графічного редактора BizAgi Process Modeler
Version 3.6

Панель <**Ribbon**> «Стрічка» містить основні елементи управління процесом моделювання, які організовані в різні групи (вкладки). Серед основних елементів управління найбільш важливою є група <Validate>. Ця група надає можливість перевірити наявність помилок моделювання на активній діаграмі процесу.

Слід підкреслити, що Process Modeler Version 3.6 дозволяє отримувати результати моделювання в форматах MS Word, MS Excel, PDF або публікувати результати в Web-браузері, на сайті SharePoint або в Wiki.

Modeler володіє розвиненим механізмом експорту/імпорту таблиця Е.2.

XPDL (XML Process Definition Language) – формат для обміну на основі XML. XPDL-файл описує всі елементи нотації, які входять до складу моделі процесу в нотації BPMN 2.0, включаючи графічні описи діаграм, а також виконувани властивості, що використовуються під час виконання. Цей формат стандартизовано Workflow Management Coalition (WfMC) [User Guide].

Таблиця Е.2.

Export (Експорт)	Image	Експорт діаграми моделі у вигляді зображення
	Visio	Експорт моделі у вигляді Visio файлу
	XPDL	Експорт моделі процесу у вигляді XPDL файлу
	Attributes	Зберігання розширених атрибутів моделі процесу в форматі XML
	BPMN	Експорт моделі у вигляді BPMN файлу
Import (Імпорт)	Visio	Імпорт діаграми у вигляді Visio файлу
	XPDL	Імпорт XPDL файлу з діаграмою процесу із іншого додатку
	Bizagi	Імпорт файлу діаграми процесу, створеної у Bizagi
	Attributes	Імпорт XML файлу з описом розширених атрибутів процесу
	BPMN	Імпорт BPMN файлу з діаграмою процесу

Е.3 Основні елементи інтерфейсу користувача Process Modeler

Панель **<Palette>** (палітра) містить графічні елементи, які використовуються для визначення моделі процесу. Ці графічні елементи розділені на групи (панелі) та відповідають розподілу в стандарті BPMN: Діяльність (Activity), Події (Event), Шлюзи (логічні оператори – Gateway), Артефакти (Association), Доріжки (Lane), Дані (Data), З'єднувачі (Sequence flow).

Панель <Activities> (Діяльність) містить набір графічних елементів для позначення різних типів завдань (Task) на діаграмі процесу. BizAgi Process Modeler розрізняє сім типів задач, що призначені для ручних або автоматичних завдань, що виконуються зовнішньою системою або користувачем (Рисунок Е.3).

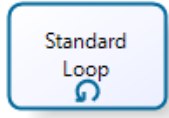
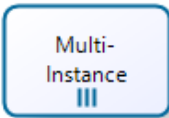


Рисунок Е.3 – Позначення типів завдань (Task) в BizAgi Process Modeler

Вище перелічені сім типів задач <Task> мають таку інтерпретацію:

1. <User Task> – задача, яка виконується виконавцем за сприяння інших акторів або програмного забезпечення.
2. <Service Task> (сервісна). Цей тип задачі призначений для надання послуги, яка може бути як веб-сервісом, так і автоматизованим додатком.
3. <Receive Task> (отримання повідомлення). Завдання вважається виконаним, якщо повідомлення отримано хоча б один раз.
4. <Send Task> (відправка повідомлення). Завдання вважається виконаним, якщо повідомлення надіслано хоча б один раз.
5. <Script Task> (сценарій). Цей тип задачі мстить завдання та порядок виконання операцій, який описаний на мові виконання, наприклад, BPEL. Зазвичай використовується для задач, які виконуються автоматичними засобами.
6. <Manua Task> (ручне виконання). Задача, яка виконується виконавцем без будь-яких засобів автоматизації.
7. <Business-Rule Task> (бізнес-правило). Задача, технологія виконання якої залежить від поточних обставин і вибирається на основі заданого бізнес-правила.

В нотації BPMN 2.0 кожне вище перелічене завдання може виконуватися або одноразове, або в режимі циклічного повторення. Крім того, кожний процес (завдання) може мати окремі екземпляри (примірники). На семантичному рівні примірники можна порівняти з виконавцями процесу, що можуть мати ім'я та відрізнятися конкретними умовами виконання, які, зазвичай, задаються логічними операторами шлюзів (Gateway) або тригерами.

Стандартний цикл – завдання можуть повторюватися послідовно в циклі із застосуванням булевої функції – з повторенням циклу на умові доки функція буде істиною	
Багато-екземплярний цикл – повторюється послідовно для всіх визначених примірників. Ітерації можуть відбуватися або послідовно, або паралельно (одночасно) в залежності від умов формування циклу	

Панель <Events> (Подія) містить набір графічних елементів для позначення різних типів подій: <Start Events> (Початкові події), Intermediate events (Проміжні події), Intermediate Events Attached to an Activity Boundary Intermediate events (Проміжні події, які приєднуються до подій та обмежують їх діяльність), End Events (Кінцеві події).

Початкові (стартові) події (Start Events) в нотації BPMN містять набір графічних елементів, які визначають початкові умови процесу а саме: безперечного старту, очікуванні сигналу, за прийомом сповіщення, за настанням певних умов, при спрацюванні логічної диз'юнкції тощо (Рисунок Е.4). Стартова подія (Start Events) – ініціює початковий процес (діаграми). Із стартової події потік управління може лише виходити, а потік повідомлень – як входити, так і виходити. На діаграмі процесу, зазвичай, показано лише одне стартове середовище, але воно може бути відсутнє або їх може бути декілька присутніх процесів з пулами, доріжками або розрізненими підпроцесами. Контур події позначається одинарною тонкою лінією.



Рисунок Е.4 – Позначення початкових умов запуску процесу

Проміжні події (Intermediate events) вказують на те, які події можуть відбуватися між початком і кінцем Процесу. Проміжні події впливають на хід процесу, але не можуть безпосередньо припинити процес. Панель містить такий набір графічних елементів проміжних подій як: таймер, сигнал, зв'язок,

повідомлення, умови, компенсація, ескалація та інші (Рисунок Е.5). Кожна з проміжних подій має певні умови настання та описується тригером, який подію запускає.



Рисунок Е.5 – Графічне позначення настання проміжних подій

Кінцеві події (End Event) не тільки свідчать про закінчення процесу, але й вказують яким чином закінчується процес. В нотації BPMN розрізняють дев'ять причин закінчення процесу, які на семантичному рівні визначають причину завершення процесу. Графічне позначення елементів кінцевих подій наведено на рисунку Е.6.



Рисунок Е.6 – Графічне позначення елементів кінцевих подій

Панель <Gateway> (Шлюз) містить набір графічних елементів, що використовуються для управління потоками послідовностей у при побудові діаграмі процесу. Вони визначають умови розгалуження, комбінації або злиття потоків. В контексті нотації BPMN термін «шлюз» означає, що існує механізм решітки – тригер, який або дозволяє, або забороняє прохід через шлюз. BizAgi Process Modeler пропонує шість різних графічних елементів для опису шлюзів – логічних операторів умов проходження потоків через шлюз (Таблиця Е.4).

Таблиця Е.4

Графічне позначення	Назва елементу	Опис
	Exclusive Gateway ексклюзивний Gateway	Використовується для створення альтернативних шляхів у процесі, але вибирається лише один. Використовується для об'єднання альтернативних шляхів.
	Event Based Gateway	Представляє точку розгалуження, де альтернативні шляхи базуються на подіях, що виникли.

		Коли спрацьовує перша подія, буде використаний шлях, що впливає з цієї події. Усі решта шляхів більше не дійсні.
	Exclusive Event Based Gateway	Шлюз на основі подій використовується для підтвердження процесів на підставі настання певної події
	Parallel Event Based Gateway	На відміну від ексклюзивного Gateway на основі подій, ВСЕ Події конфігурації шлюзу повинні бути спровоковані для створення екземпляра Process. У ньому не повинно бути Вхідних переходів.
	Parallel Gateway	Розгалуження: використовується для створення альтернативних шляхів без перевірки будь-яких умов. Об'єднання: використовується для об'єднання усіх вхідних потоків
	Complex Gateway	Розгалуження: контроль складних точок рішення у процесі для створення альтернативних шляхів в процесі, використовуючи вирази. Об'єднання: дозволяє продовження до наступного пункту Процесу, коли умова бізнесу стає справжньою.
	Inclusive Gateway	Процес розгалуження базується на умовних виразах. А ІСТИНА однієї умови не виключає оцінку інших умов. Об'єднання: використовується для злиття комбінації альтернативні та паралельних шляхів

Панель <Artifacts> (Артефакти) містить набір графічних елементів <Group> група, <Annotation> анотація, <Image> імідж, <Header> заголовок, <Formatted Text> форматування тексту, <Custom Artifacts>

Артефакт <Group> забезпечує візуальний механізм неформального групування елементів BPMN-діаграми.

Артефакт <Annotation> являє собою механізм надання додаткової інформації для читача BPMN-діаграми.

Артефакт <Header> відображає властивості діаграми (автор, версія, опис) та автоматично оновлюється інформацією, що міститься в цих властивостях.

<Custom Artifacts> Допомагає визначити та використовувати власні артефакти. Артефакти надають можливість відображати додаткову інформацію про Процес, яка безпосередньо не пов'язана з потоком.

Панель <Data> Дані містять два графічних елемента:

<Data Objects> Об'єкт даних – надає інформацію про те, як документи, дані та інші об'єкти використовуються при опису процесу;

<Data Store> Забезпечує механізм опису Баз даних.

Панель <Swimlanes> Доріжки містить елементи:

<Pool> пул – контейнер, який містить графічне відображення взаємодії між елементами нотації, що описують процес;

<Lane> використовуються для розмежування елементів між процесами – внутрішніх ролей, посади, відділи тощо;

<Milestone> – розділяє процес на окремі зони. Він може вказувати на різні етапи процесу.

Панель <Connectors> містить три типи стрілок, що визначають різний характер зав'язків між графічними примітивами (елементами нотації) на BPMN-діаграмі:

<**Sequence Flow**> використовується для визначення активного порядку взаємодії між графічними елементами BPMN-діаграми;

<**Association**> використовують для пов'язування інформації та артефактів із графічними елементами BPMN-діаграми;

<**Message flow**> використовується для показу потоку повідомлень між графічними примітивами (двома об'єктами), які готові їх надсилати та отримувати за логікою BPMN-діаграми.

Панель <Element Properties>

Кожен графічний елемент нотації BPMN 2.0 має свої властивості та залежно від типу елемента відображаються вкладками з додатковою інформацією та функціональністю. Інформація про властивості графічних елементів застосовується для документування процесу в виді XML-файлу та

поділяється на: основну, розширену, додаткову та презентаційну інформацію. Такий поділ дозволяє дуже точно описувати властивості графічних елементів моделі процесу, включаючи й значення конкретних атрибутів нотації BPMN 2.0 до кожного окремого елемента діаграми.

Панель <View> надає користувачу BizAgi Process Modeler широкий набір можливостей для огляду та представлення BPMN-діаграм, а також навігації.

ДОДАТОК Ж Побудова моделей обробки даних засобами програмної платформи RapidMiner

З позиції отримання кінцевого результату, процес обробки великих даних складається з виконання послідовності окремих операцій (операторів) над даними, які пов'язані семантичними відношеннями та мають на меті досягнення визначених цілей аналізу. В залежності від цілей та мети аналізу, а також природи даних (структуровані /неструктуровані), для аналізу можуть залучатися різноманітні методи та способи обробки, наприклад, очищення, нормалізація, перетворення форматів, кластерний аналіз, навчання, моделювання, прогнозування, оцінка якості тощо. Крім того, для аналізу різних типів даних мають використовуватися саме ті моделі, які відповідають як типу даних так й цілям та мети аналізу. Тому, створення моделі процесу обробки даних *та експериментальне підтвердженням якості аналізу на контрольних вибірках* є актуальною задачею для аналітика, оскільки така модель в подальшому використовується не тільки для аналізу поточних даних, але дає можливість знаходження залежностей у великих розмірах даних., а також їх відображення в зручному для сприйняття вигляді.

В якості інструментального засобу моделювання та інтелектуального аналізу даних обрано програмну платформу RapidMiner, яку в дослідженнях компанії Gartner за 2014 [117] та 2017 роки [76] визнано лідером в області Advanced Analytics. RapidMiner написана на мові Java і поширюється за ліцензією GNU Affero General Public License (вільне розповсюдження) [111]. RapidMiner реалізує більш 400 різноманітних операторів обробки як структурованих, так і неструктурованих даних (включаючи текстові) [110]. Вихідні дані та результати аналізу можуть бути представлений різними форматами CSV, Excel, XML, SAS, Access, AML, ARFF, XRFF, Database, SPSS, Stata, Sparse, dBase, C4.5, BibTeX, DasyLab, URL, включаючи роботу з хмарними сховищами та управління процесами і групами об'єктів.

RapidMiner є програмною платформою Technical University of Dortmund, яка забезпечує інтегроване середовище для моделювання процесів аналізу даних для машинного навчання, інтелектуального аналізу даних, аналізу тексту, прогностичної аналітики і бізнес-аналітики. швидкої розробки додатків і підтримує всі етапи процесу видобутку даних, включаючи результати візуалізації, перевірки та оптимізації (Рисунок Ж.1). Програмна платформа написана на мові Java і в даний час поширюється за ліцензією GNU Affero General Public License. Залежно від конфігурації системи є вільно поширювані і комерційні ліцензії. До всіх основних функцій RapidMiner забезпечує доступ через Java API і версію програми для командного рядка (а не тільки через загальний користувальницький інтерфейс).

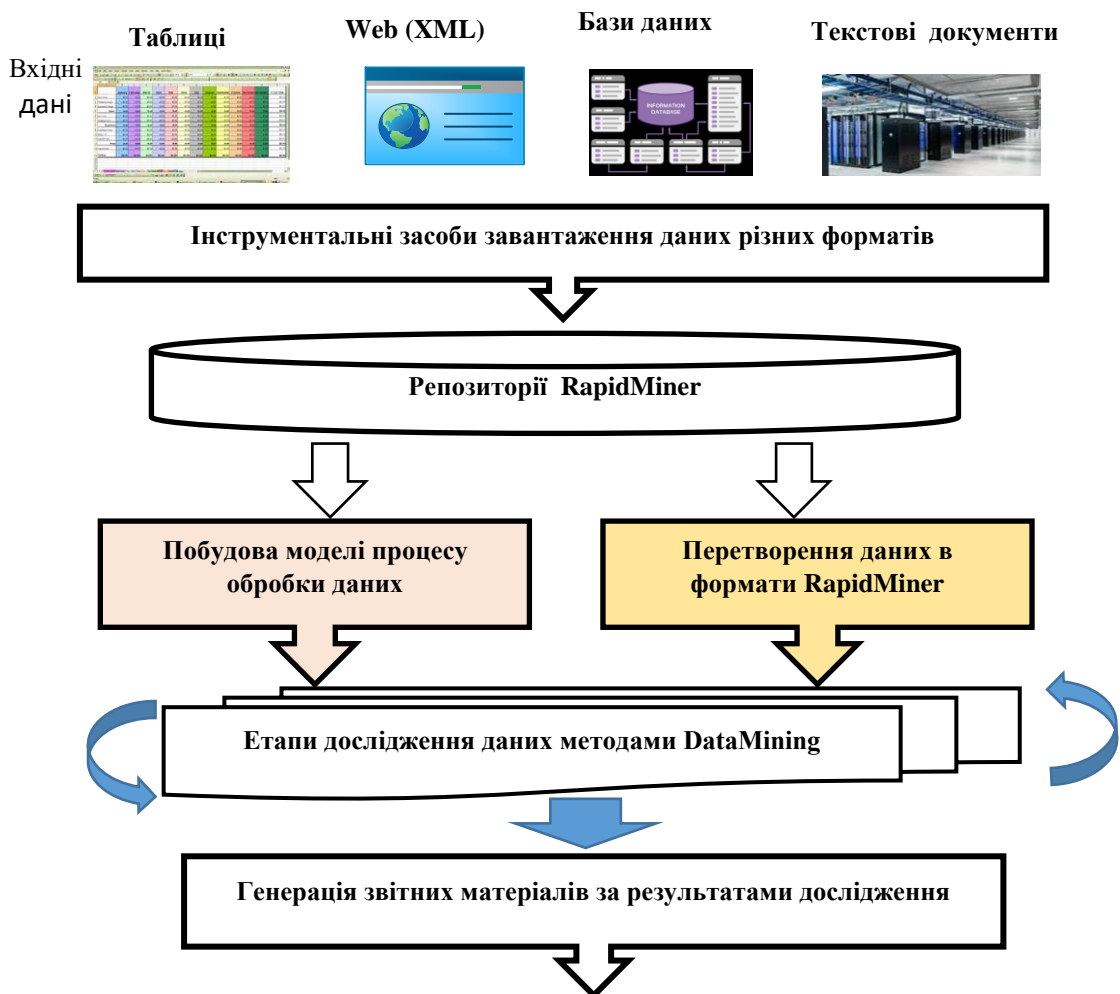


Рисунок Ж.1 – Інтегроване середовище для моделювання процесів аналізу даних – RapidMiner

Методологія побудови моделі обробки даних для формування типових сценаріїв зокрема засобами RapidMiner складається з наступних етапів:

1. Початкова фаза інтелектуального аналізу даних – Problem Definition фокусується на розумінні цілей і вимог, які висуваються до аналізу даних. Після того як визначена проблема інтелектуального аналізу даних, має бути розроблена модель процесу реалізації аналізу даних.

2. Підготовки даних – Data Gathering and Preparation охоплює всі завдання, що виникають при створенні таблиць, які потрібні для побудови моделі даних, а також очищення даних та їх перетворення. RapidMiner надає можливість завантаження даних з БД або хмарних сховищ (Amazon S3, Azure Blob, Dropbox).

3. Побудова моделі процесу обробки (аналізу) даних та її розвиток – Model Building and Evaluation. На цьому етапі здійснюється вибір найбільш приємного методу моделювання та калібрування параметрів оптимальних значень операторів обробки даних, з урахуванням особливостей моделі. Якщо алгоритм вимагає перетворення даних, потрібно зробити крок назад до попереднього етапу для необхідного налаштування даних. Результатом виконання цього етапу є

- Модель процесу аналізу даних відповідності до визначеній проблеми;
- Сценарії навчання моделі аналізу даних на тестовій вибірці даних;
- Тестування моделі аналізу даних та оцінка відповідності цілям аналізу.

4. Еволюція та налаштування моделі обробки даних на аналіз з конкретними параметрами.

Процес аналізу даних представляється у вигляді інтерактивного графа операторів обробки даних. Основним робочим елементом в середовищі RapidMiner є процес – (робоче поле <Process>) В основі роботи процесу лежить налагодження окремих компонентів процесу – операторів <Operators>.

У лівій частині екрана RadpidMiner розташовані панель завантаження даних – <Repository> і панель операторів.

Методологія побудови моделі обробки даних засобами RapidMiner складається з наступних етапів:

1. **Початкова фаза інтелектуального аналізу даних** – Problem Definition фокусується на розумінні цілей і вимог, які висуваються до аналізу даних. Після того як визначена проблема інтелектуального аналізу даних, має бути розроблена модель процесу реалізації аналізу даних.

2. **Підготовки даних** – Data Gathering and Preparation охоплює всі завдання, що виникають при створенні таблиць, які потрібні для побудови моделі даних, а також очищення даних та їх перетворення.

3. **Побудова моделі процесу обробки (аналізу) даних та її розвиток** – Model Building and Evaluation. На цьому етапі здійснюється вибір найбільш приємного методу моделювання та калібрування параметрів оптимальних значень операторів обробки даних, з урахуванням особливостей моделі. Якщо алгоритм вимагає перетворення даних, потрібно зробити крок назад до попереднього етапу для необхідного налаштування даних. Результатом виконання цього етапу є:

- Модель процесу аналізу даних відповідності до визначеній проблеми;

- Сценарії навчання моделі на тестовій вибірці даних;

- Тестування моделі та оцінка відповідності цілям аналізу;

4. **Еволюція та налаштування моделі обробки даних** на аналіз з конкретними параметрами.

Згідно з методикою будівельними блоками моделі процесу аналізу даних є оператори. В RapidMiner оператори розділені на функціональні групи, такі як: Data Access (доступ до даних), Blending (маніпулювання), Cleansing (очищення), Modeling (моделювання), Scoring (оцінювання), Validation (перевірка коректності моделі) та інші.

Процес аналізу даних представляється у вигляді інтерактивного графа операторів обробки даних (Рисунок Ж.2). У лівій частині екрана RapidMiner розташовані панель завантаження даних – <Repository> і панель операторів – <Operators>. RapidMiner надає можливість завантаження даних з БД або хмарних сховищ (Amazon S3, Azure Blob, Dropbox).

Основним робочим елементом в середовищі RapidMiner є процес – (робоче поле <Process>) В основі роботи процесу лежить налагодження окремих компонентів процесу – операторів (вкладка <Operators>).

Сьогодні до програмної платформи RapidMiner включено більш 400 операторів.

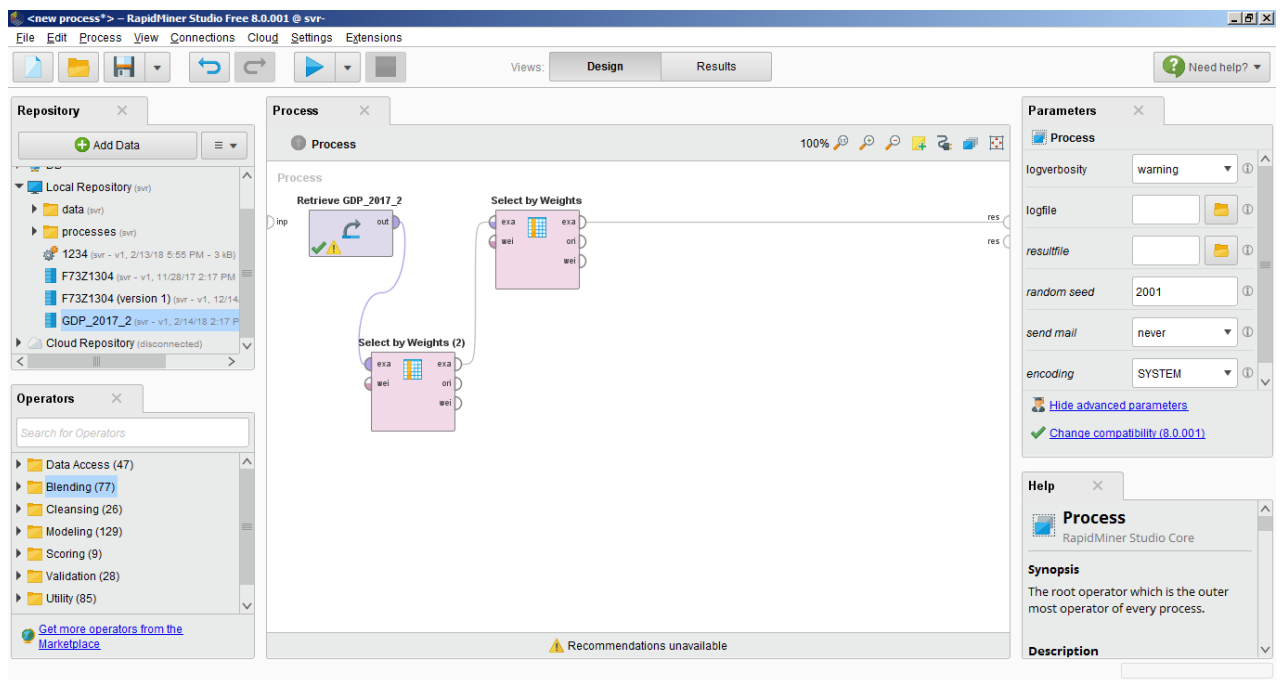


Рисунок Ж.2 – Головна сторінка програмної платформи RapidMiner 8.0.

Оператори розташовані у вкладці <Operators> програмної платформи RapidMiner 8.0. розбиті на функціональні групи – (Рисунок Ж.3). В останній версії RapidMiner 8.0. групи оператори розподілені наступним чином [111]:

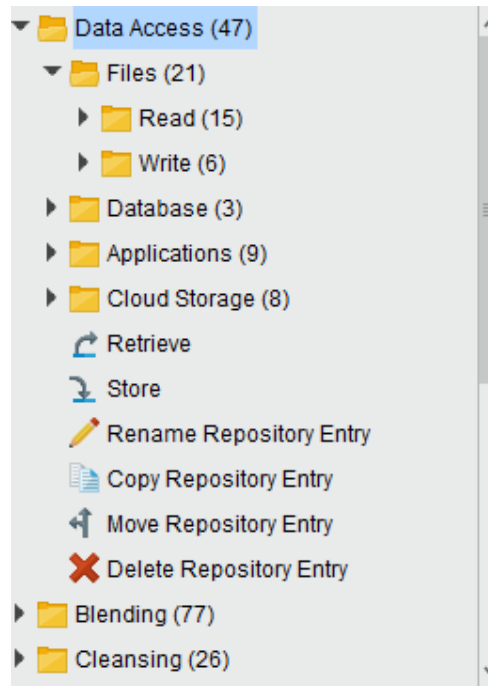


Рисунок Ж.3 – Дерево операторів в RapidMiner 8.1.

➤ **Data Access** (доступ до даних) – складається з множини операторів, за допомогою яких задаються робота з даними читання/запис даних за різними форматами зберігання (CSV, Excel, XML, SAS, Access, AML, ARFF, XRFF, Database, SPSS, Stata, Sparse, dBase, C4.5, BibTeX, DasyLab, URL), включаючи роботу з хмарними сховищами, та управління процесами і групами об'єктів (збереження, вилучення, циклу) у <Repository>;

➤ **Blending** (маніпулювання) – складається з множини операторів, які виконують процедури: перейменування і модифікації ролей, перетворення типу даних, перетворення атрибутів, зміна значень, очищення даних, фільтрація, сортування, агрегація даних, операції над множинами даних (об'єднання, перетин і ін.);


➤ **Cleansing** (очищення) – множина операторів для очищення даних (нормалізація даних, усунення дублікатів даних, зменшення розміру, заміна відсутніх значень, усунення відхилень в даних тощо);

➤ **Modeling** (моделювання) – множина операторів процесів обробки даних:

- прогнозування (модель на вимогу, байєсівська модель, дерева рішень, правила, нейронні мережи, підтримка векторних машин, дискримінантний, аналіз логічна регресія і ін.);
- кластеризації та сегментації (модель k-Means і її модифікації, розбиття по щільності на основі просторової кластеризації, максимізації математичного очікування, випадкового розбиття, кластеризації зверху вниз, агломератної кластеризації та ін.);
- зважування атрибутів;
- асоціації;
- кореляції і обчислення залежностей (перехідна матриця,
- подібності обчислень;
- оптимізація (параметрична, вибір функції, генерація функцій, важливість функції);
- **Scoring** (оцінювання) Конфіденційність;
- **Validation** (перевірка коректності моделі) – складається з множини операторів, які виконують вимір продуктивності, значущості, точності обчислень;
- **Utility** (утиліти) – складається з множини операторів, які використовують для макросів і журналів дій, роботи з файлами, створення наборів даних, додавання анотації та інші;
- **Extensions** (розширення) – множина операторів, які можуть використовуватися для розширення функціонального складу операторів за бажанням дослідника (наприклад, Text Processing).

Кожен оператор зображений піктограмою, з лівого боку якої розташовані вузли входу, а з правого - вузли виходу, через які оператори зв'язуються між собою та з робочою областю для моделювання процесу обробки даних. Виходи одного оператора можна з'єднувати з входами іншого, тоді результат його роботи буде надходити на вхід іншого оператора. Як

правило, кожний оператор має параметри, які задають параметри моделі та налаштовуються у правому полі – <Parameters>.

Процес моделювання запускається кнопкою  Run на верхній панелі. Після закінчення роботи програма попросить дозволу переключитися в режим огляду результатів моделювання <Results>.

RapidMiner може використовуватися і як окремий додаток, і як «інтелектуальний двигок», вбудований в інтегроване середовище для вирішення завдань аналізу великих даних. Отже, завданнями RapidMiner можуть бути як дослідні застосування – для моделювання процесів обробки різноманітних за структурою даних (включаючи й великі дані), так і прикладні – для реального інтелектуального аналізу даних, включаючи аналіз тексту (text mining), аналіз мультимедіа (multimedia mining), аналіз потоків даних (data stream mining), що можуть бути вбудовані.

ДОДАТОК 3. Фрагмент програмної реалізації методу інтелектуальний метод визначення наступних подій сценарію аналітичної діяльності при формуванні наступного кроку сценарію

```
# CART (Classification and Regression Tree) algorithm implementation.
from pprint import pprint
in_data = [ ['x1','y1','z1',18,'A'],
            ['x2','y2','z1',23,'B'],
            ['x3','y1','z1',24,'C'],
            ['x4','y2','z1',23,'C'],
            ['x2','y3','z2',21,'B'],
            ['(empty)','y4','z2',12,'A'],
            ['(empty)','y3','z2',21,'C'],
            ['x2','y1','z2',24,'B'],
            ['x1','y2','z1',19,'A'],
            ['x3','y1','z2',18,'A'],
            ['x2','y3','z2',18,'A'],
            ['x4','y3','z2',19,'A'],
            ['x3','y4','z1',12,'C'],
            ['x1','y3','z2',21,'A'],
            ['x2','y3','z1',18,'C'],
            ['x4','y2','z1',19,'C']
]

class dnode:
    def __init__(self, col=-1, value=None, results=None, tb=None, fb=None):
        self.col=col
        self.value=value
        self.results=results
        self.tb=tb
        self.fb=fb

# Divides a set on a specific column. Can handle numeric
# or nominal values
```

```

def divideset(rows,column,value):
    # Make a function that tells us if a row is in
    # the first group (true) or the second group (false)
    split_func=None
    if isinstance(value,int) or isinstance(value,float):
        split_func=lambda row:row[column]>=value
    else:
        split_func=lambda row:row[column]==value

    # Divide the rows into two sets and return them
    set1=[row for row in rows if split_func(row)]
    set2=[row for row in rows if not split_func(row)]
    return (set1,set2)

# Create counts of possible results (the last column of
# each row is the result)
def uniquecounts(rows):
    results={}
    for row in rows:
        # The result is the last column
        r=row[len(row)-1]
        if r not in results: results[r]=0
        results[r]+=1
    return results

# Probability that a randomly placed item will
# be in the wrong category
def giniimpurity(rows):
    total = len(rows)
    counts = uniquecounts(rows)
    imp = 0
    for t1 in counts:
        p1 = float(counts[t1]) / total
        for t2 in counts:

```

```

    if t1 == t2:
        continue
    p2 = float(counts[t2]) / total
    imp += p1*p2
return imp

# Entropy is the sum of p(x)log(p(x)) across all
# the different possible results
def entropy(rows):
    from math import log
    log2=lambda x:log(x)/log(2)
    results=uniquecounts(rows)
    # Now calculate the entropy
    ent=0.0
    for r in results.keys():
        p=float(results[r])/len(rows)
        ent=ent-p*log2(p)
    return ent

def growtree(rows,scoref=entropy):
    if len(rows) == 0:
        return dnode()
    current_score=scoref(rows)

    # Set up some variables to track the best criteria
    best_gain = 0.0
    best_criteria = None
    best_sets = None

    column_count=len(rows[0])-1
    for col in range(0,column_count):
        # Generate the list of different values in

```



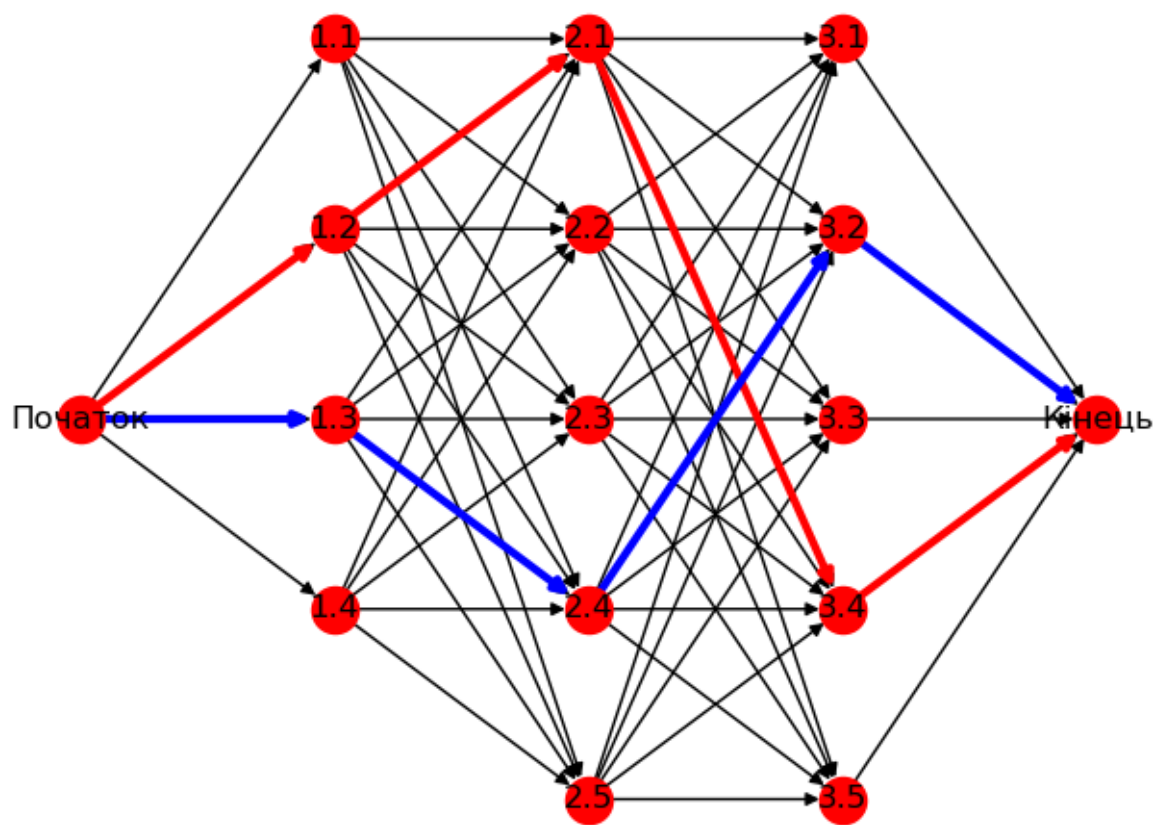
```

# this column
column_values = {}
for row in rows:
    column_values[row[col]]=1
# Now try dividing the rows up for each value
# in this column
for value in column_values.keys():
    (set1,set2)=divideset(rows,col,value)

# Information gain
p = float(len(set1)) / len(rows)
gain = current_score - p * scoref(set1) - (1-p) * scoref(set2)
if gain > best_gain and len(set1) > 0 and len(set2) > 0:
    best_gain = gain
    best_criteria = (col,value)
    best_sets = (set1,set2)
# Create the sub branches
if best_gain > 0:
    trueBranch = growtree(best_sets[0])
    falseBranch = growtree(best_sets[1])
    return dnode(col=best_criteria[0],
                value=best_criteria[1],
                tb=trueBranch,
                fb=falseBranch)
else:
    return dnode(results=uniquecounts(rows))

```

В результаті застосування алгоритму Дійкстри з попереднім розподілом часу з вершин по дугах, а вдруге вартості можна показати, що за різними критеріями обираються різні шляхи у графі:



ДОДАТОК І Акти впровадження

ЗАТВЕРДЖУЮ

Декан ТЕФ НТУУ «КПІ імені Ігоря Сікорського,

д. т. н., професор



Євген ПИСЬМЕННИЙ

«15» січня 2021р.

АКТ

про використання результатів дисертаційного дослідження Ковалю О.В.
«Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності»
в навчальний процес

Комісія у складі:

голови комісії – професора, д. т. н. Гаврилко Є.М.

членів комісії:

- професора, д. т. н. Федорової Н.М.;

- доцента, к. т. н. Ковальчука А.М.,

склала цей акт про те, що результати наукових досліджень, які були проведені Ковалем О.В. в процесі підготовки дисертаційної роботи на тему «Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності», впровадженні в навчальний процес кафедри автоматизації проектування енергетичних процесів та систем теплоенергетичного факультету Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» у вигляді:

1. Теоретико-методологічного та методичного забезпечення навчального процесу при підготовці студентів кафедри за спеціальністю 121 – Інженерія програмного забезпечення за освітньою програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем і веб-технологій», а саме, при викладанні навчальних дисциплін «Інженерія даних та знань», «Технології інформаційного пошуку», «Модельно-орієнтоване проектування програмних систем та при написанні дипломних робіт та магістерських дисертацій. Нові положення стосуються застосування запропонованих:

– теоретичного підходу до побудови та моделювання семантичних моделей прикладної області програмних систем (ПС) та сценаріїв аналітичної діяльності за застосування цих систем, якій забезпечує зменшення помилок при проектуванні ПС на етапі концептуального проектування;

– практичних підходів до розробки алгоритмів та програмних засобів, які базуються на модельно-орієнтованих підходах до проектування ПС и методах моделювання сценаріїв дій користувача-аналітика при застосуванні аналітичних програмних систем, які забезпечують скорочення розробки процесів моделювання та проектування ПС;

– методу процесного моделювання сценаріїв дій користувача та автоматизованого/автоматичного процесу побудови сценаріїв;

– формування моделі предметної області на основі моніторингу інформаційного простору та аналізу семантичних мереж;

– теоретичного та практичного підходу до розробки програмних засобів семантичного пошуку інформації за використанням онтологічної моделі предметної області інформаційного пошуку.

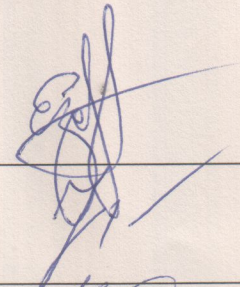
2. Теоретико-методологічного та алгоритмічного і програмного забезпечення моделюючих комплексів та програмно-апаратних дослідних стендів навчально-наукових лабораторій теплоенергетичного факультету, а саме:

– програмно-апаратного комплексу моделювання гідроакустичних процесів;

– програмно-апаратного дослідного стенду оптимізації енергопостачання та енергорозподілу;

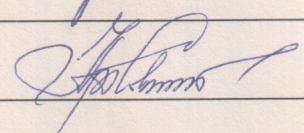
– прототипу програмного комплексу дослідження міжнародної науково-технічної діяльності наукової організації.

Голова комісії:
д. т. н., професор



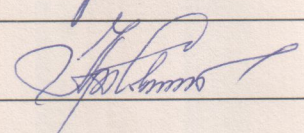
Є.М. Гаврилко

Члени комісії:
д. т. н., професор



Н.М. Федорова

к. т. н., доцент



А.М. Ковальчук



СИМВОЛ

НАУКОВО-ВИРОБНИЧЕ ПІДПРИЄМСТВО

ЄДРПОУ – 18093404, Україна, 01133, м. Київ, вул. М. Бойчука, буд. 6, офіс 103

Тел. 592 95 90, E - mail: svrukr@gmail.com

АКТ

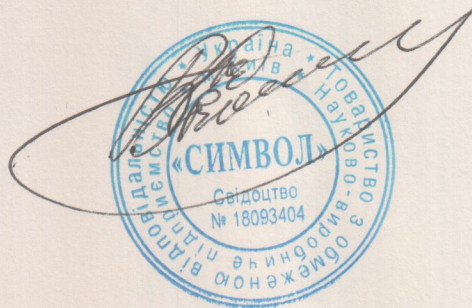
про застосування результатів дисертаційної роботи Ковалю О.В.
«Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності», що подається на здобуття вченого ступеня доктора технічних наук

Отримані автором результати, що представлені в у науково-практичній розробці, яка заснована на результатах отриманих в дисертаційній роботі Ковалю О.В. «Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності», пройшли успішну апробацію та можуть бути рекомендовані для використання при моделюванні функціонування складних технічних програмно-апаратних систем з метою аналізу технічних рішень, які приймаються на етапі проектування цих систем.

Представлені на апробацію методи та програмні засоби комп'ютерного моделювання сценаріїв аналітичної діяльності були використані при виконанні робіт в рамках міжнародного контракту щодо проектування та розробки моделюючого комплексу дослідження взаємодії фізичних полів різної природи. Запропонований підхід до здійснення моделювання функціонування програмної та апаратної частин технічної системи за використанням програмних засобів як побудови типових сценаріїв, які можуть бути задіяні в реальній системі, так й побудови сценаріїв в процесі моделювання експериментів по аналізу функціонування прототипу системи, показав ефективність його застосування та дав можливість скоротити час виконання робіт.

Вважаємо, що представлені на апробацію теоретичні результати та програмні засоби можуть сприяти більш ефективному вирішенню питань, пов'язаних з дослідженням та створенням засобів проектування складних динамічних систем.

Директор
ТОВ НВП «Символ»



В.Р. Сенченко

21-01-2021

АКТ

про впровадження результатів дисертаційної роботи Ковалю О.В. **«Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності»**, що подається на здобуття вченого ступеня доктора технічних наук

Чинний акт складено про те, що в ТОВ «Інженерна логіка» впроваджено окремі результати дисертаційної роботи Ковалю О.В., а саме методи та засоби математичного та комп'ютерного моделювання багаторівневих сценаріїв управління попитом на електричну енергію в реальних умовах функціонування електричних мереж з використанням стимулюючих механізмів енергоринку. Запропоновані підходи до формування сценаріїв управління попитом на енергію з врахуванням варіативності процесів споживання та генерування, ефективно використовуються у базових сценаріях взаємодії механізмів стимулювання кінцевого споживача до активності дозволяють здійснювати ефективне управління енергетичними потоками та енергетичне маневрування потужностями, сприяють розвантаженню та зменшенню пікового навантаження мережі, застосуванню ресурсо-енергозберігаючих режимів експлуатації енергетичного обладнання, зменшенню негативного впливу енергетичної інфраструктури на навколишнє середовище. Уваги заслуговують запропоновані підходи до формування сценаріїв взаємодії елементів енергетичної інфраструктури в умовах інтегрованих та, особливо, автономних мікромереж з обмеженою потужністю та обсягами генерування. Застосування гнучкого сценарного підходу до управління енергетичною інфраструктурою дозволяє максимально гармонізувати процеси генерування та споживання енергії з врахуванням потреб споживача.

Отримані результати використано при виконанні виробничих завдань технологічного циклу ТОВ «Інженерна логіка», причому практичне застосування цих результатів засвідчило ефективність запропонованих методів і засобів математичного моделювання процесів функціонування енергетичних мереж в умовах енергоринку, а також показало перспективність їх подальшого використання для управління попитом на електричну енергію на стороні споживача.

Директор ТОВ «Інженерна логіка»

О.В. Разумовський





N I T HOLDINGS LIMITED

НАЦІОНАЛЬНИЙ ФОНД РОЗВИТКУ

ІНВЕСТИЦІЇ
ТА
ІНОВАЦІЇ

03005, Україна, м. Київ, вул. Сирецька, 33

Тел.: (050) 5810640 Е - mail: nfr@nitholdings.ua

13.11.2020 № 20-Д/20
на № _____

АКТ

про застосування результатів дисертаційної роботи Ковалю О.В. «Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності», що подається на здобуття вченого ступеня доктора технічних наук

Даний акт складений про те, що в Дочірньому підприємстві «НАЦІОНАЛЬНИЙ ФОНД РОЗВИТКУ» застосований запропонований в дисертаційній роботі Ковалю О.В. «Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності» теоретичний та практичний підхід до побудови типових сценаріїв досліджень і розробки моделей сигналів і перешкод, розрахунку параметрів поширення звукових хвиль в багат шарової морському середовищі, зокрема метод перетворення BPMN-моделі в модель уявлення про предметну область сценарію та інтелектуальний метод визначення наступних подій сценарію.

Отримані Ковалем О.В. результати розробки інформаційної технології конвертації BPMN-моделі в OWL-модель сценарію аналізу сигналів та інтелектуальний програмний засіб, що навчається на діях користувача та на основі отриманих в процесі навчання знань представляє дії користувача у вигляді дерева рішень ймовірного сценарію аналізу роботи моделей сигналів і перешкод пройшли апробацію і були використані при виконанні робіт за договором №STE-3-233-Д/К-15 від 01.10.2015 р. та за договором №STE-3-92-Д/К-16 від 25.04.2016 р., що засвідчило практичну значимість запропонованих програмних інструментів при вирішенні задач дослідження та вибору оптимальних за відповідними критеріями алгоритмів аналізу сигналів, і дозволяє в цілому підвищити ефективність проектування складних технічних систем та надійність їх функціонування.

Голова Правління



М.Х. Іл'ясов

ЗАТВЕРДЖУЮ

Проректор з наукової роботи

КПІ імені Ігоря Сікорського

д. т. н., професор



Віталій ПАСІЧНИК

2021р.

про впровадження результатів дисертаційної роботи Ковалю О.В.
«Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності»

Чинний акт складено про те, що в Особливому Конструкторському Бюро «Шторм» (ОКБ «Шторм») Науково-дослідної частини КПІ імені Ігоря Сікорського впроваджено окремі результати наукових досліджень, які були проведені Ковалем О.В. в процесі підготовки дисертаційної роботи на тему «Методи та засоби комп'ютерного моделювання сценаріїв аналітичної діяльності», а саме: процес автоматизованої побудови сценаріїв аналітичної діяльності на основі бібліотеки типових сценаріїв (прецедентів) та трирівнева комп'ютерна моделююча система (інтегроване програмне інструментальне середовище) для тестування і оцінки моделювання сценаріїв аналітичної діяльності.

Отримані результати використано при виконанні виробничих завдань за договорами № USE-16.2-101-D/K-18 від «18» червня 2018 р. та № DLN-20-DP-01 від «15» грудня 2020 р. для дослідження з метою оптимізації роботи вимірювальної станції і методики її лабораторних випробувань та моделювання впливу довкілля на параметри комплексного поля на об'єкти моніторингу. Практичне застосування цих результатів засвідчило ефективність запропонованих засобів аналізу фізичних процесів функціонування вимірювальних систем та об'єктів моніторингу, а крім того показало перспективність їх подальшого використання для якісного відтворення властивостей, характеристик і параметрів технічних систем, які моделюються в процесі їх моделювання та проектування.

Головний інженер
ОКБ «Шторм»

Сергій ПУХА