

ІВАНО-ФРАНКІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
НАФТИ І ГАЗУ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

МЕЛЬНИК ВІТАЛІЙ ДМИТРОВИЧ

УДК 004.89:681.518:622.24

ДИСЕРТАЦІЯ

**«ІНТЕЛІМЕДІЙНА АВТОМАТИЗОВАНА СИСТЕМА ПІДТРИМКИ
ПРИЙНЯТТЯ РІШЕНЬ ПРИ БУРІННІ НАФТОГАЗОВИХ
СВЕРДЛОВИН»**

Спеціальність 05.13.07 – автоматизація процесів керування

Галузь знань 15 – автоматизація та приладобудування

Подається на здобуття наукового ступеня кандидата технічних наук (доктора філософії)

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

_____ /В. Д. Мельник/

Ідентичність всіх примірників дисертації

Засвідчую

Вчений секретар спеціалізованої

вченої ради Д 20.052.03 _____ /Т. В. Гуменюк/

Підпис Т. В. Гуменюка засвідчую:

Вчений секретар університету _____ /В. Р. Процюк/

Науковий керівник: **Шекета Василь Іванович**, доктор технічних наук, професор

Івано-Франківськ, 2021

АНОТАЦІЯ

Мельник В. Д. Інтелімедійна автоматизована система підтримки прийняття рішень при бурінні нафтогазових свердловин. – На правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.07 – «Автоматизація процесів керування». – Івано-Франківський національний технічний університет нафти і газу, Міністерство освіти і науки України, Івано-Франківськ, 2021.

Дисертацію присвячено розробці методу інтелімедійної підтримки прийняття рішень в процесі буріння нафтогазових свердловин на основі методології масштабованих маркерів, що розглядається як основний елемент технології інтелектуального тренінгу з використанням тьюторних тренажерів як виду інтелектуальних автоматизованих систем. При побудові концептуальної предметної моделі інтелімедійної інформаційної системи виконано моделювання основних компонентів автоматизованих тренінгових систем у формі інтелектуальних тренажерів з точки зору особливостей представлення контенту тренінгового матеріалу та методів тестування, що застосовуються, в тому числі в формі емуляції ситуацій підтримки прийняття керуючих технологічних рішень.

Досліджено формальні основи процесу моделювання контенту інтелімедійної автоматизованої системи на основі базової технології представлення та задоволення обмежень шляхом введення технологічних проблем з накладеними обмеженнями як ситуацій інтелектуальної підтримки прийняття рішень засобом реалізованих інтелектуальних функцій інтелімедійної системи. Визначена структура релевантності щодо умов задоволення технологічних обмежень в контексті прийняття рішень, що дозволяє введення метричних характеристик формальних конструкцій в процесі побудови та імплементації абдуктивного виведення знань на основі мультимедійних входжень. Для рішення даної задачі досліджено сутність та види абдуктивних міркувань з введеними ймовірнісними складовими процесу інтерпретації, що дозволило оцінити ефективність абдуктивного підходу в цілому

при вирішенні задач класифікації на множині мультимедійного контенту з метою подальшої імплементації релевантних доменних знань, в тому числі шляхом їх видобування на основі мультимедійних даних з регульованою ймовірнісною оцінкою в формі коефіцієнтів впевненості для заданих формальних конструкцій.

Запропоновано структуру алгоритмічного та програмного забезпечення системи інтелімедійної підтримки прийняття рішень на основі методології масштабованих графічних маркерів з накладеними обмеженнями.

Ключові слова: інтелімедіа, інтелектуальна підтримка прийняття рішень, буріння нафтових і газових свердловин, графічні маркери, масштабованість, правила, бази знань, абдуктивний фреймворк, коефіцієнти впевненості, обмеження, растрові зображення, відеофайли.

ABSTRACT

V. D. Melnyk. The intellimedia automated system for decisions support by the drilling of the oil and gas wells. – The manuscript.

The thesis for a candidate of technical sciences degree on speciality 05.13.07 – Automated systems. – Ivano-Frankivsk National Technical University of Oil and Gas, Ministry of Education and Science of Ukraine, Ivano-Frankivsk, 2021.

The thesis is devoted to the development of method for intellimedia based decision support in the process of drilling of oil and gas wells based on the methodology of scalable markers, treated as an essential part of the intelligent training technology with the use of tutor-simulators as an intelligent automated systems means. By the constructing of the conceptual model of the subject of intellimedia information system, the basic components of the automated training systems was modelled for achieving the form of intelligent simulators in terms of features that represent the content of training materials and the methods of testing, including in the form of decision-support situations emulation control for projected technological solutions.

The basics of simulation process for content of intellimedia automated system were studied based on the core technology of representation and constraint satisfaction theory

by introducing the concept of technological problems with constrains situations as intelligent decision support tool implemented main intelligent features of intellimedia system. The structure meets the relevant conditions regarding technological constrains in the context of decision making and it allows the introduction of metric characteristics of formal structures in the process of building and implementing knowledge abductive output based on multimedia entries. The essence and types of abductive reasoning with imposed probabilistic component of the interpretations were analysed that allowed to evaluate the effectiveness of abductive approach as a whole by solving problems of classification on the set of multimedia content in order to facilitate the implementation of relevant domain knowledge, including their extraction from media with variables probabilistic estimation in the form of confidence factors for scalable markers with imposed constraints tool developing.

The structure of algorithmic and system software for decision-making support based on the methodology of scalable graphic markers with imposed constraints was created for the existing intellimedia environment.

Keywords: intellimedia, intelligent decision support, drilling of oil and gas wells, graphic markers, scalability, rules, knowledgebases, abductive framework, certainty coefficients, constraints, bitmap images, videos.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Мельник В. Д., Демчина М. М, Шекета В. І. Технології інтерпретації документів в інтелектуальних системах дистанційного навчання. *Вісник Кременчуцького нац. ун-ту ім. М. Остроградського*. 2011. № 3 (68). С.18-22.
2. Мельник В. Д., Шекета В. І., Демчина М. М. Виведення значень логічних параметрів в інтелектуальних системах дистанційного навчання. *Вісник Хмельницького нац. ун-ту. Технічні науки : наук. журнал*. Хмельницьк, 2011. № 1. С. 118-122.
3. Вовк Р. Б., Шекета В. І., Мельник В. Д. Система інтелектуальної підтримки прийняття рішень при контролі технологічних параметрів. *Методи та прилади контролю якості = Methods and devices of quality control: науково-технічний журнал / За ред. І. С. Кісіля; ІФНТУНГ*. 2012. № 2 (29). С. 119-129.
4. Вовк Р. Б., Мельник В. Д., Гобир Л. М. Подання та оброблення технологічних знань про процес буріння на основі обмежень. *Науковий вісник ІФНТУНГ*. 2013. №1(34). С. 73-81.
5. Шекета В.І., Демчина М. М., Мельник В. Д. Імплементация інтелектуальної стратегії прийняття рішень у процесі буріння. *Нафтогазова енергетика: Всеукраїнський науково-технічний журнал*. 2013. №2(20). С. 38-50.
6. Мельник В. Д. Абдуктивне виведення знань про процес буріння на основі мультимедійних даних про бурове обладнання. *Науковий вісник Івано-Франківського національного технічного університету нафти і газу*. 2016. №1(40). С.80-91.
7. Шекета В. І., Мельник В.Д., Гобир Л. М. Інтелімедійна інформаційна система підтримки прийняття рішень в процесі буріння. *Проблеми інформаційних технологій : науковий журнал ВАК*. 2016. №01 (019) травня. С.96-116. (Індексується в Index Copernicus).
8. Мельник В. Д. Формальна структура кейсів технологічних проблем в процесі побудови їх рішень на основі обмежень. *Математичні машини та*

системи: науковий журнал Інституту проблем математичних машин та систем НАН України. 2016. №2. 2016. С. 116-127. (Індексується в RSCI-PIHЦ)

9. Мельник В. Д. Використання продукційного підходу для інтелектуалізації діалогів з користувачем в системах дистанційного навчання. *Радиоелектроника и молодежь в XXI веке : материалы XIII Международного молодежного форума* (г., Харьков, 30 марта - 1 апреля 2009). X. : ХНУРЕ, 2009. ч.2. С.168.

10. Мельник В. Д., Шекета В. І., Демчина М. М. Обробка суб'єктивних оцінок групи експертів в інтелектуальній системі дистанційного навчання. *Математичне і програмне забезпечення інтелектуальних систем : матеріали VII Міжнар. наук.-практ. конф.* (м. Дніпропетровськ, 18-20 листопада 2009 р.). Дніпропетровськ, ДНУ, 2009. С.184-185.

11. Мельник В. Д. Структуризація відповідей користувача в системі дистанційного навчання на основі експертних параметрів. *Системний аналіз та інформаційні технології : матеріали XI Міжнародної науково-технічної конференції* (м. Київ, 26-30 травня 2009 р.). К.: НТУУ "КПІ", 2009. С.348.

12. Мельник В. Д., Процюк В. Р., Процюк Г. Я. Представлення документів та їх класів в системах дистанційного навчання. *Сучасні проблеми прикладної математики та інформатики : матеріали XVI Всеукраїнської наукової конференції* (м. Львів, 8-9 жовтня 2009 р.). Львів, 2009. С.143.

13. Мельник В. Д., Шекета В. І., Демчина М. М. Прийняття рішень при модифікації запитів в системі дистанційного навчання. *Теоретичні та прикладні аспекти побудови програмних систем : матеріали шостої Міжнародної конференції* (м. Київ, 8-10 грудня 2009 р.). К., 2009. С. 149-156.

14. Vitaliy Melnyk, Roman Vovk, Mykola Demchyna. Frame Based Approach to Construction of Intelligent System for Student Knowledge Control. 2010 IEEE Xth International Conference Modern Problems Of Radio Engineering, Telecommunications And Computer Science: Proceedings, Lviv-Slavske, February 23-27, 2010. P.287. (Індексується в Scopus, RSCI-PIHЦ).

15. Мельник В. Д. Побудова контенту навчального матеріалу орієнтованого на знання. *Математичне і програмне забезпечення інтелектуальних систем* : матеріали VIII Міжнародної науково-практичної конференції (м. Дніпропетровськ, 10-12 листопада 2010 р.). Дніпропетровськ, 2010. С. 148.

16. Демчина М. М., Мельник В. Д., Гобир Л. М. Застосування функцій доцільності при прийнятті рішень щодо оптимізації технологічних параметрів. *Математичне та імітаційне моделювання систем МОДС 2013* : матеріали восьмої міжн. наук.-практ. конференції: тези доповідей (Чернігів-Жукин, 24-28 черв. 2013 р.). Чернігів, 2013. С. 145-149.

17. Мельник В. Д., Якубів Т. Р. Побудова інтелектуальних навчальних систем на основі мультиагентних технологій. *Інформаційні технології в освіті, науці, техніці та промисловості* : зб. тез доповідей Всеукр.наук.-практ. конференції аспірантів, молодих учених та студентів (Івано-Франківськ, 8-11 жовтня 2013 р.). Івано-Франківськ, 2013. С. 98-100.

18. Мельник В. Д. Особливості представлення навчального контенту в інтелектуальній тьюторній системі на основі обмежень. *Математичне та програмне забезпечення інтелектуальних систем* : тези доповідей XI міжнародної наук.-практ. конф. (м. Дніпропетровськ, 20-22 лист. 2013 р.). Дніпропетровськ, 2013. С. 157-159.

19. Демчина М. М., Мельник В. Д., Бестильний М. Я. Інтелектуальний контроль параметрів буріння. *Методи та засоби неруйнівного контролю промислового обладнання*: збір. тез. доп. 4-ої наук.-практ. конф. (м. Івано-Франківськ, 26-27 лист. 2013 р.). Івано-Франківськ: ІФНТУНГ, 2013. С.41-43.

20. Демчина М. М., Юрчишин В. М., Мельник В. Д. Дослідження функціонування бази знань системи адаптивного тестування. *Сучасні інформаційні технології в дистанційній освіті* : зб. тез доповідей III Всеукраїнської науково-практичного семінару (м. Івано-Франківськ, 22-24 вер. 2014 р.). Івано-Франківськ: вид-тво ІФНТУНГ, 2014. С.97-100.

21. Мельник В. Д., Шекета В. І., Романишин Ю. Л., Гургула О. Б. Застосування інтелімедійних інформаційних технологій в навчальному процесі.

Інформаційні технології та комп'ютерне моделювання: матеріали Міжнародної науково-практичної конференції (м. Івано-Франківськ, 23-28 травн. 2016 р.) / Прикарпатський національний університет імені Василя Стефаника. Івано-Франківськ, 2016. С. 62-64.

22. Sheketa V., Melnyk V., Romanyshyn Yu., Chesanovsky M. The Construction of Technological Problems Cases for the Purpose of Intelligible Control. *Perspective technologies and methods in MEMS design (MemsTech'2016) : the XIIth International Conference (Ukraine, Lviv-Polyana, 20-24 th April). Ukraine, Lviv-Polyana, 2016. IEEE Conference Publications. P.96-99.* (Індексується в Scopus, Web of Science, IEEE Xplore Digital Library).

23. Юрчишин В. М., Чесановський М. С., Стисло Т. Р., Мельник В. Д. Побудова знання-орієнтованого управління технологічними процесами нафтогазової галузі. *Інформаційні технології в освіті, техніці та промисловості* : матеріали III всеукраїнської науково-практичної конференції молодих учених і студентів (м. Івано-Франківськ, 10-13 жовтня 2017р.). Івано-Франківськ: вид-тво ІФНТУНГ, 2017. С. 15-17.

24. Юрчишин В.М., Стисло Т.Р., Стисло О.В., Гобир Л.М., Мельник В.Д. Інтелімедійні методи контролю якості бурового обладнання. *Сучасні прилади, матеріали і технології для неруйнівного контролю і технічної діагностики машинобудівного і нафтогазопромислового обладнання: зб. матеріалів доповідей восьмої міжнар. наук.-техн. конф. пам'яті професора Ігоря Кісіля, м. Івано-Франківськ, 14-16 листопада 2017 р. С. 175-177.*

25. Мельник В. Д., Шекета В. І. Інтелімедійні засоби підтримки прийняття рішень в нафтогазовій справі. *Молодий вчений: науковий журнал.* 2019. № 5 (69). С. 268-271.

26. Мельник В. Д. Структуризація тьюторних складових навчальних систем. *Міжнародний науковий журнал «Інтернаука».* 2020. № 3 (83), 1 т. С. 47-50.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	12
ВСТУП.....	13
РОЗДІЛ 1	
ПРЕДМЕТНА МОДЕЛЬ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	
1.1 Технології технологічних тренінгів. Автоматизовані імітаційні тренажери.....	21
1.2 Моделювання основних компонентів автоматизованих тренажерів..	45
1.3 Представлення контенту технологічного матеріалу.....	61
1.4 Методи тестування в автоматизованих навчальних системах і тренажерах.....	66
1.5 Обґрунтування напряду досліджень.....	73
РОЗДІЛ 2	
ФОРМАЛЬНІ ОСНОВИ МОДЕЛЮВАННЯ КОНТЕНТУ ІНТЕЛІМЕДІЙНОЇ СИСТЕМИ	
2.1 Аналіз досліджень у галузі представлення та задоволення обмежень.....	79
2.2 Реалізація інтелектуальних функцій інтелімедійної системи.....	83
2.3 Введення технологічних проблем.....	92
2.4 Структура релевантності в умовах задоволення технологічних обмежень.....	102
2.5 Метричні характеристики введених формальних конструкцій.....	110
ВИСНОВКИ ДО РОЗДІЛУ 2.....	117
РОЗДІЛ 3	
ПОБУДОВА КОНЦЕПЦІЇ АБДУКТИВНОГО ВИВЕДЕННЯ ЗНАНЬ НА ОСНОВІ МУЛЬТИМЕДІЙНИХ ВХОДЖЕНЬ	
3.1 Сутність і види абдуктивних міркувань у відповідних	120

фреймворках.....	
3.2	Інтерпретація імовірнісної складової..... 125
3.3	Оцінка ефективності абдуктивного підходу..... 129
3.4	Введення задачі класифікації на множині контенту..... 132
3.5	Імплементация релевантних доменних знань..... 138
3.6	Оцінка застосування коефіцієнтів впевненості в абдуктивному фреймворку..... 143
3.7	Обґрунтування процедури видобування мультимедійних даних і знань..... 148
	ВИСНОВКИ ДО РОЗДІЛУ 3..... 161
	РОЗДІЛ 4
	РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛІМЕДІЙНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ У ПРОЦЕСІ БУРІННЯ
4.1	Оцінка ефективності імплементации мультимедійного контенту на рівні оболонки експертної системи..... 163
4.2	Розробка структури та реалізація прототипу інтелімедійної системи як версії інтелектуальної системи з мультимедійним інтерфейсом..... 170
4.3	Оцінка ефективності отриманого прототипу при підтримці прийняття рішень у процесі буріння..... 176
4.4	Тестування отриманого прототипу в режимі інтелектуального тьютора-тренажера..... 183
	ВИСНОВКИ ДО РОЗДІЛУ 4..... 205
	ОСНОВНІ РЕЗУЛЬТАТИ РОБОТИ..... 207
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 210
	ДОДАТОК А – Класифіковані лістинги проектованої моделі інтелімедійної інформаційної системи підтримки прийняття рішень в нафтогазовій справі..... 241
	ДОДАТОК Б - Концептуальна схема реалізованого модуля 267

інтелімедійної інформаційної системи підтримки прийняття рішень в нафтогазовій справі.....	
ДОДАТОК В - Діаграма класів реалізованого модуля інтелімедійної інформаційної системи підтримки прийняття рішень в нафтогазовій справі.....	268
ДОДАТОК Г - Діаграма прецедентів реалізованого модуля інтелімедійної інформаційної системи підтримки прийняття рішень в нафтогазовій справі.....	269
ДОДАТОК Д - Класифікована таблиця назв бурових установок.....	270
ДОДАТОК Е - Зображення уміх класів бурових установок використаних для реалізації модуля інтелімедійної інформаційної системи підтримки прийняття рішень в нафтогазовій справі	277
ДОДАТОК Ж - Схема класифікації БУ.....	278
ДОДАТОК И – Приклад тесту на базі графічного контенту (робота з частинками графічного матеріалу).....	279
ДОДАТОК К – Приклад тесту на базі графічного контенту (робота в тестовому режимі).....	282
ДОДАТОК Л – Приклад тесту на базі графічного контенту (режим підтримки прийняття рішень).....	286
ДОДАТОК М – Акт передачі у промислове використання методики реалізації процедур підтримки прийняття технологічних рішень в процесі буріння на основі інтелімедійних баз знань та баз даних експертного досвіду.....	288
ДОДАТОК Н – Акт впровадження унавчальний процес наукових результатів дисертаційної роботи асистента кафедри документознавства та інформаційної діяльності Мельника В.Д. «Інтелімедійна інформаційна система підтримки прийняття рішень в нафтогазовій справі».....	289

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

АСК – автоматизована система керування;

БЗ – база знань;

ЕС – експертна система;

ППР – підтримка прийняття рішень;

СППР – система інтелектуальної підтримки прийняття рішень;

СППР – система підтримки прийняття рішень;

СУБД – система управління базами даних;

ТП – технологічний процес;

ACL – концепція абдуктивного навчання (для абдуктивних фреймворків);

CSP – проблема задоволення обмежень;

WITS – специфікація для передачі даних з бурової площадки;

WITSML – мова розмітки специфікації для передачі даних з бурової площадки;

\forall – квантор загальності «для всіх», «для будь-яких»;

\exists – квантор існування «існує»;

\in, \notin – “належить”, “не належить” відповідно;

\vee – логічна операція «АБО», диз’юнкція;

\wedge – логічна операція «І», кон’юнкція;

\cup – операція “об’єднання” множин;

\cap – операція “перетину” множин;

\Rightarrow – логічна імплікація, «слідують»;

\Leftrightarrow – логічна еквівалентність, «еквіваленція»;

\subset, \subseteq – підмножина, власна підмножина відповідно;

\supset, \supseteq – надмножина, власна надмножина відповідно;

\times – декартів добуток;

$|=$ – відношення наслідування.

ВСТУП

Прототип інтелімедійної інтелектуальної системи нафтогазової предметної галузі базується на знаннях про галузь, що характеризується нечіткістю, невизначеністю, імовірністю та можливістю. Система проектується як інтелектуальний тренажер, який дозволить студенту адаптивно керувати віртуальним аналогом графічного інтерфейсу пульту бурильника, задаючи значення керованих параметрів технологічного процесу й оцінюючи параметри на виході.

В загальному випадку на сьогоднішній день мультимедійні технології за своєю природою є статичним відображенням певних аспектів технологічних знань. І з даної позиції розбивання суцільного зображення на сітку та встановлення жорстких і динамічних маркерів є, очевидно, єдино можливим рішенням проблеми. Початково було встановлено, що для прототипу системи необхідно розробити форму візуального представлення знань, щоб зробити її, відповідно, більш корисною. Тип візуальних даних оцінюється з точки зору їх важливості. В якості базового типу при виконанні аналізу літературних джерел вибрано растровий тип зображень. Виконане порівняння мультимедійних і немультимедійних форматів зображень даних предметної області підтвердили правильність вибраного рішення.

Для представлення знань із предметної галузі використання відео й анімації є корисним, але, водночас, не є абсолютно необхідним, тому що необхідну демонстрацію можна побудувати, зокрема, і на основі нерухомих зображень. В інтелімедійній системі підтримки прийняття рішень у процесі буріння основним застосуванням визначено режим інтелектуального тренажера як початкового етапу створення прототипу системи SCADA-рівня. Тому, в даному випадку, з точки зору точності, рухомі картини предметної області були б безумовно кращими в плані представлення динаміки предметної області. Водночас, статичне візуальне представлення даних було визначено як основна вимога щодо системи GRID-прив'язаних маркерів. Вимога звукового супроводу отримала низький пріоритет, хоча зрозуміло, що звуковий опис технологічної операції може бути корисним (і реально використовується в промислових застосуваннях). Проте така форма

представлення інформації предметної області в режимі тренажера не є визначальною, а більшою мірою є доповнюючою, що відповідає загальному принципу представлення та сприйняття інформації оператором технологічного процесу (ТП), згідно якого звук є менш корисним, ніж візуальні дані.

Як відомо, основна увага при проектуванні будь-якої експертної системи приділяється тому, щоби комп'ютерне представлення експертних даних максимально відповідало способу представлення, який використовують експерти предметної області. В проектуваному прототипі саме візуальне представлення експертних даних у формі растрових зображень визначає сутність корисності та практичної орієнтованості проектуваної системи з точки базової архітектури експертних систем в цілому. Експертні знання, в будь-якому випадку, не можуть бути спотворені при вибраному способі представлення. Тому використання будь-якого спрощення й узагальнення рівня представлення необхідним чином призводитиме до потенційної втрати важливої інформації про предметну область. Тому, визначивши основним завданням ефективно поєднання технології експертних систем і мультимедійних систем, очевидно, необхідно розглянути всі можливі способи графічного представлення сутностей предметної області.

Актуальність теми. Дане дослідження присвячене розробці та реалізації інтелектуальної системи з мультимедійним інтерфейсом та базою знань з мультимедійними включеннями. При створенні прототипу було розглянуто ряд аргументів, які також можна успішно застосувати для задачі створення інтелектуальної системи, яка оперує з мультимедійним контентом. Основна проблема на даному шляху полягає в досягненні ефективності процесу перенесення експертних знань та досвіду в комп'ютерну програму. Основна особливість полягає саме в тому, що такі знання та досвід опираються на мультимедійну основу. Загально відомо, що використання мультимедійних даних та технологій на даному етапі розвитку інформаційних технологій пов'язано з рядом обмежень, які суттєво впливають на кожен етап життєвого циклу розробки інтелектуальних систем. Тому в якості основної проблеми можна виділити проблему практичної імплементації шляхом інтеграції технології інтелектуальних систем та технології мультимедійних

систем і даних, які, кожна в своєму класі, є, в принципі, достатньо дослідженими, але в питаннях їх інтеграції, яка дозволить збереження саме інтелектуальних функцій, є ряд невирішених аспектів, яким, власне, і присвячене дане дослідження.

Важливим аспектом досліджуваної проблеми є також тестування отриманого прототипу системи з метою оцінки ефективності її функціонування, зокрема, в задачах підтримки прийняття рішень. Як показує досвід, для створення ефективної інтелектуальної системи необхідно вирішити ряд допоміжних проблем. Особливо тоді, коли система поєднує в собі різнопланові технології. Зокрема, в мультимедійній експертній системі важливо визначити, яким найкращим чином представляти експертну інформацію, використовуючи всі доступні мультимедійні можливості. Проте рішення задачі інтеграції мультимедіа в існуюче оточення не може вирішуватися за рахунок порушення цілісності основної архітектури експертної системи. Слід виходити з того, що мета використання мультимедіа полягає в доповненні вже існуючої бази знань таким чином, щоб її можна було застосовувати більш ефективно. Тобто, мультимедійні включення не мають на меті заміну бази знань як основного компоненту, а використовуються як доповнюючий елемент. Те ж стосується інтерфейсу. Ознакою високоякісної експертної системи є тісний зв'язок між інформацією (знаннями) і способом її представлення. В ідеальному випадку, експертна система з певною точністю дублюватиме міркування експерта, презентуючи такі міркування як вид знань. Проте в більшості предметних областей така ідеальна відповідність є неможливою. Тому переважно використовується спеціалізовані методи представлення знань, які споріднені зі специфікою роботи експерта. Однак, змінюючи способи представлення знань, завжди існує небезпека порушення цілісності такого представлення і, як наслідок, порушення цілісності самої системи. Тому верифікація цілісності (**Data Integrity** для баз даних, **Logical Consistency** для баз знань) є необхідним інструментом, що повинен бути інтегрований в ядро системи та складати невід'ємну частину її функціональності. В даному контексті саме мультимедійні засоби є тією доступною технологією, яка дозволяє точну імітацію уявлень експерта з конкретного знання (предметної області). Оскільки такі дані інтегруватимуться через інтерфейс, то,

відповідно, необхідно чітко визначати, які складові інтерфейсу є необхідними, а які, відповідно, доповнюючими. Для цього необхідно встановити черговість представлення форматів даних, щоб експерт зміг визначити найбільш важливі елементи і включити їх як мінімально необхідні.

Зв'язок роботи з науковими програмами, планами, темами. Вибраний напрям досліджень є складовою частиною тематичного плану кафедри інженерії програмного забезпечення Івано-Франківського національного технічного університету нафти і газу (ІФНТУНГ). Тематика роботи є частиною планових науково-дослідних робіт із розвитку нафтогазового комплексу України та базується на результатах виконання науково-дослідних тем: «Розробка теоретичних та прикладних концепцій застосування сучасних інформаційних технологій в нафтогазовій галузі» 2008-2012 рр. (затв. Науковою радою ІФНТУНГ протокол № 3/48 від 08.09.2008р.); «Розробка моделей, методів і алгоритмів прямого і непрямого опрацювання великих потоків даних для підтримки прийняття управлінських рішень у сфері енергозабезпечення держави» (номер державної реєстрації 0118U006795).

У вищеназваних темах НДР автор був безпосереднім виконавцем робіт щодо розробки формальних моделей інтелімедійної системи підтримки прийняття рішень процесів буріння нафтових і газових свердловин на основі мультимедійного контенту.

Мета і задачі дослідження Метою даної роботи є розробка методу та системи інтелімедійної підтримки прийняття рішень при виконанні технологічних операцій в процесі буріння нафтогазових свердловин.

Досягнення вказаної мети забезпечується в дисертаційній роботі шляхом розв'язання таких задач:

- 1) Аналіз предметної моделі інтелімедійної інформаційної системи.
- 2) Формалізація постановки задачі моделювання процесів інтелімедійної підтримки прийняття рішень при виконанні технологічних операцій процесу буріння шляхом вирішення технологічних проблем з накладеними обмеженнями в рамках абдуктивного фреймворку.

3) Обґрунтування та дослідження технології інтелімедійної підтримки прийняття рішень при виконанні технологічних операцій в процесі буріння нафтових і газових свердловин в умовах невизначеності.

4) Побудова формальної структури функціонування системи процесів підтримки прийняття рішень при оперуванні з мультимедійним контентом на кожному етапі життєвого циклу.

5) Розробка формального механізму підтримки прийняття рішень оператором в технологічному процесі буріння нафтових і газових свердловин на основі оперування з класифікованими входженнями мультимедійного контенту предметної області.

6) Реалізація системи інтелімедійної підтримки прийняття рішень при виконанні технологічних операцій в процесі буріння нафтових і газових свердловин.

7) Розробка методики інтелімедійної підтримки прийняття рішень щодо вибору технологічних операцій в процесі буріння на основі систем масштабованих маркерів з накладеними обмеженнями.

Об'єктом дослідження є нестационарний технологічний процес буріння нафтових і газових свердловин, який супроводжується необхідністю прийняття ефективних та раціональних рішень в умовах невизначеності.

Предметом дослідження є системи інтелектуальної підтримки процесів прийняття рішень при виконанні технологічних операцій процесів буріння нафтових і газових свердловин шляхом вирішення технологічних проблем на основі обмежень в абдуктивному фреймворку.

Методи дослідження. Для вирішення поставлених задач у роботі проведені теоретичні дослідження з використанням методів порівняльного аналізу, систематизації та узагальнення (для аналізу автоматизованих систем-тренажерів, структури прийняття рішень в технологічному процесі); системного підходу; формально-логічних досліджень із використанням базового апарату на основі предикатної логіки; методів теорії множин (для моделювання функціональності складових системи); методів теорії представлення та задоволення обмежень і теорії ймовірності; методів математичної статистики.

Наукова новизна одержаних результатів полягає в розвитку та поглибленні методологічних підходів щодо підтримки прийняття рішень при виконанні технологічних операцій процесу буріння нафтових і газових свердловин, що ілюструється мультимедійним контентом та визначається наступними науковими результатами:

- **вперше** представлено модель інтелімедійної автоматизованої системи з виділеними основними компонентами інтелектуальної сутності, що *дозволило* визначити структурні елементи процесу підтримки прийняття рішень при бурінні нафтових і газових свердловин шляхом введення технологічних проблем з накладеними обмеженнями та імплементованою релевантністю щодо умов задоволення та порушення накладених технологічних обмежень;
- **удосконалено** формальні основи моделювання контенту інтелімедійної автоматизованої системи на основі теорії представлення та задоволення обмежень;
- **покращено** формальні засоби абдуктивного виведення знань на основі мультимедійних входжень в рамках введеного інтелімедійного фреймворку з імовірнісною складовою у формі абдуктивних міркувань та введеною задачею класифікації на множині мультимедійного контенту;
- **удосконалено** процедури видобування мультимедійних даних та знань на основі методології коефіцієнтів впевненості для абдуктивного фреймворку, що дозволяє імплементацію релевантних доменних знань про технологічні операції процесу буріння.

Отримали подальший розвиток методи інтелектуалізації прийняття рішень та систем інтелектуальної підтримки прийняття рішень в процесі буріння шляхом реалізації метрично впорядкованих інтелімедійних маркованих складових.

Практичне значення отриманих результатів дисертаційної роботи полягає в тому, що розроблені моделі, методи та методики, а також алгоритмічне та програмне забезпечення, побудоване на їх основі, дають змогу вирішувати задачі прийняття ефективних та раціональних рішень в процесі буріння, що відповідно підтвердило свою ефективність під час тестування отриманої інтелімедійної автоматизованої системи підтримки прийняття рішень на фактичних даних

«Укрнафта Буріння» ПАТ «Укрнафта». Розроблена система та рекомендації з її використання, прийняті до впровадження відділом АСУ ТП підприємства «МІКРОЛ» як прототип модуля візуалізації процесу підтримки прийняття рішень на основі баз знань мультимедійного контенту у «SCADA»-системах (акт від 01.03.2016 р.) і «Укрнафта Буріння» ПАТ «Укрнафта» (акт від 15.03.2016 р.). Результати досліджень упроваджені в навчальний процес кафедри інженерії програмного забезпечення ІФНТУНГ (акт від 15.03.2013 р.) для студентів напряму підготовки 6.050103 – Програмна інженерія в дисципліні «Методологічні основи наукових досліджень в нафтогазовій галузі» та спеціальності 8.05010301 – Програмне забезпечення систем у дисципліні «Математичні методи аналізу алгоритмів».

Особистий внесок здобувача. Основні положення та результати дисертаційної роботи, які виносяться на захист, одержані автором особисто. У роботах, написаних у співавторстві, дисертанту належать: у роботі [1] – спосіб інтерпретації маркованих входжень; у роботі [2] – ідея інтелектуалізації технології дистанційного навчання шляхом введення контрольованих параметрів; у роботі [3] – методологія підтримки прийняття рішень по виділених ключових параметрах; у роботі [4] – ідея представлення знань про технологічний процес буріння; у роботі [5] – фреймово-продукційна методологія побудови інтелектуальних стратегій підтримки прийняття рішень.

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи доповідались на 18-ти міжнародних і всеукраїнських науково-практичних конференціях, у тому числі:

1. XIII Міжнародний молодіжний форум «Радиоэлектроника и молодежь в XXI веке» (30 марта-1 апреля 2009., г. Харьков, Харьковский национальный университет радиоэлектроники).
2. VII, VIII, XI Міжнародні науково-практичні конференції «Математичне та програмне забезпечення інтелектуальних систем MPZIS-2009, 2010, 2013» (м. Дніпропетровськ, Дніпропетровський національний університет ім. Олеса Гончара).

3. XI Міжнародна науково-технічна конференція «Системний аналіз та інформаційні технології» (26-30 травня 2009 р., м. Київ, Національний технічний університет України «Київський політехнічний інститут»).
4. Шоста Міжнародна конференція «Теоретичні та прикладні аспекти побудови програмних систем» (8-10 грудня 2009 р., м. Київ, Національний університет «Києво-Могилянська академія»).
5. Xth International Conference «Modern Problems Of Radio Engineering, Telecommunications And Computer Science TCSET 2010» (February 23-27, 2010, Lviv-Slavske, Ukraine)
6. Восьма міжнародна науково-практична конференція «Математичне та імітаційне моделювання систем МОДС 2013» (24-28 черв. 2013 р. м. Чернігів-Жукин, Чернігівський державний технологічний університет).
7. XIIth International Conference «Perspective technologies and methods in MEMS design MemsTech'2016» (20-24 April 2016, Lviv-Polyana, Ukraine).

Публікації: основні результати дисертаційної роботи викладено в 26 наукових працях, в тому числі 8 статей у наукових фахових виданнях України, 2 публікації у виданнях, включених до Міжнародної науково-метричної бази SCOPUS та 16 публікацій у матеріалах міжнародних та Всеукраїнських науково-технічних конференцій.

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел з 219 найменувань на 31 сторінці й 12 додатків на 49 сторінках. Загальний обсяг роботи складає 289 сторінок. Основний зміст викладено на 209 сторінках. Робота містить 34 рисунки та 1 таблицю.

РОЗДІЛ 1

ПРЕДМЕТНА МОДЕЛЬ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1. Технології технологічних тренінгів. Автоматизовані імітаційні тренажери

Нафтогазова галузь характеризується складними технологічними процесами, аварії під час яких призводять до значних економічних, екологічних і людських втрат. Для роботи з подібними процесами потрібні спеціально підготовлені кваліфіковані оператори, на яких лягає значна відповідальність за наслідки прийнятих рішень з управління процесом виробництва. За деякими оцінками, за період 60-90-х рр. частка аварій з вини операторів серед найбільш великих аварій у світовій нафтохімії і нафтопереробці становить 26%, а середні втрати в результаті однієї великої аварії перевищують 35 млн. доларів.

З бурхливим розвитком комп'ютерних технологій з'явилася можливість моделювати складні технологічні комплекси для підготовки та підвищення кваліфікації фахівців у різних сферах господарської діяльності. З цих умов у багатьох країнах використання комп'ютерних навчальних тренажерів (автоматизованих систем навчання - АСН) для навчання персоналу в сфері нафтогазовидобування, переробки та нафтохімії стає законодавчою нормою.

Зокрема, федеральний стандарт США передбачає проходження обов'язкового комп'ютерного тренінгу для операторів, прийнятих на роботу, а також перепідготовку працівників не рідше одного разу на три роки. Американський нафтовий інститут (API) рекомендує використовувати комп'ютерні тренажери режимів пуску та зупинки для визначення рівня кваліфікації працівників щороку. Внутрішні правила провідних нафтових компаній світу передбачають обов'язковий відновний курс після відпусток, хвороби або тривалої відсутності практики з інших причин.

Використання імітаційних навчальних тренажерних комплексів дозволяє підвищити професійний рівень оперативного та технологічного персоналу нафтогазовидобувної галузі, набути необхідного практичного досвіду - відпрацювання базових навиків роботи з системою управління та навиків дій в аварійних або нештатних ситуаціях (без ризику негативного впливу на хід реального технологічного процесу і не вдаючись до експериментів на реальних об'єктах). При цьому, одним із найефективніших методів навчання підвищення кваліфікації операторів є застосування навчальних комп'ютерних тренажерів реального часу.

Основне завдання таких тренажерів - формування комплексних навиків прийняття рішень при аварійних або нештатних ситуаціях, що базується на моделюванні динамічного відгуку об'єкта й системи управління на довільні дії оператора. Тобто, за своєю суттю, навчальні тренажери є програмними рішеннями для моделювання хіміко-технологічних процесів у динамічному режимі із забезпеченням режиму реального часу.

Основним недоліком комп'ютерних навчальних тренажерів є обмеження інтелектуальної підтримки процесів прийняття рішень при виконанні технологічних операцій – наприклад, процесів буріння нафтових і газових свердловин – та їх висока вартість. Відсутня можливість використання комп'ютерних навчальних тренажерів з фахової діяльності на багатьох підприємствах зумовила зростаючий попит на співпрацю зі спеціалізованими навчальними центрами. Серед них «Тренажерний буровий центр» Івано-Франківського національного технічного університету нафти і газу, який є унікальним у сфері перепідготовки фахівців-буровиків за напрямом “Контроль стану свердловин. Керування свердловиною під час флюїдопроявів” згідно з вимогами Міжнародного Форуму Свердловинного Контролю (International Well Control Forum – IWCF) із урахуванням питань безпеки та уникнення економічних втрат від помилок персоналу.

На жаль, вартість даних навчальних систем обчислюється десятками і сотнями тисяч доларів. Самостійно підприємства не в змозі розробляти настільки складні інформаційні системи та комплекси через відсутність на ринку відповідних сертифікованих комплексних навчальних центрів. Спеціалізовані підприємства, які

розробляють комп'ютерні навчальні тренажери, не продають свої платформи для їх створення. Водночас абсолютно очевидно, що існуючі підприємства не можуть задовольнити потреби всіх підприємств.

В містах Москва, Іркутськ і Тюмень використовуються всього три повномасштабні бурові тренажери. Однак, новий, четвертий, на відміну від попередніх версій Drilling Systems, дозволить відпрацьовувати складні технологічні процеси, наприклад, ліквідацію прихватів бурильної колони.

На сьогодні, за кордоном існує кілька сучасних тренажерних платформ, розроблених і підтримуваних основними світовими виробниками комп'ютерних тренажерних комплексів (КТК) – ABB Simeon, Inc., Honeywell, Inc., CAE Link, Inc. й ін. Ними розроблено теорії для диспетчерських тренажерів із транспорту газу, методику побудови диспетчерських автоматизованих тренажерів, а також створено комплекси математичного, інформаційного та програмного забезпечення для тренажерів диспетчера з вирішення завдань оперативного управління газотранспортною системою [1-6].

Тренажерні програми також розробляються фахівцями ТОВ «Енергоавтоматика». Створено тренажер оператора магістрального продуктопроводу, який імітує функціонування всього продуктопроводу, системи вимірювання, обробки та передачі інформації. Математичний модуль тренажера моделює функціонування продуктопроводу, системи вимірювання та датчиків, алгоритми обробки інформації в контролерах нижнього рівня, процес передачі даних каналами зв'язку та функціонування бази даних верхнього рівня. Цей тренажер функціонує спільно з системою управління на базі MicroSCADA німецької фірми ABB. Крім того, компанія має й інші розробки спільно зі SCADA-системами: MOTOROLA SCADA, MMG SCADA, Infinity SCADA.

Серед розробників тренажерних комплексів слід відзначити також компанії ЗАТ «Транзас» і ЗАТ «Автоматизація Моніторингу Технологій», які володіють тренажерними комплексами, призначеними для визначення остаточного ресурсу магістральних нафтогазопроводів та довговічності трубопроводу за різних режимів експлуатації. Вони допоможуть фахівцям з управління транспортуванням нафти і

газу здійснювати вибір оптимального режиму, який дозволить максимально продовжити термін служби трубопроводів з метою створення запасу часу для проведення профілактичного ремонту і заміни «старих» ділянок.

Зокрема ВАТ «Гіпротрубопровід» використовується програмна продукція з прогнозу ресурсу трубопроводів, що базується на аналітичних розробках щодо оцінки надійності транспорту нафти і газу, створених фахівцями ВАТ «Гіпротрубопровід», РАО «Газпром», ВНІСПТнефть. Як вихідні дані використовуються результати експериментальних досліджень (отриманих при дефектоскопії і при оглядах) зразків металу, що реально працювали на трубопроводі. В інституті проблем транспорту енергоресурсів (ІПТЕР - колишній ВНІСПТнефть) розроблені програми для розрахунку міцності ресурсу й надійності елементів трубопроводів. За словами авторів, програми дозволяють промоделювати раціональні параметри, випробувальний тиск, проаналізувати руйнування.

Нижче наведено приклади різних видів тренажерів для нафтогазової галузі. **Тренажер-імітатор буріння АМТ-231** є ефективним технічним засобом навчання та підвищення кваліфікації інженерного персоналу бурових підрозділів нафтогазовидобувних підприємств. Використовується також для навчання студентів за спеціальностями буріння свердловин, розробки та експлуатації нафтових і газових родовищ. Апаратно-програмний комплекс тренажера складається з пультів і постів управління, обладнання для проведення свердловин, персонального комп'ютера та програмного забезпечення. Тренажер імітує в реальному і прискореному масштабі часу технологічні процеси провідки свердловин: поглиблення, спускопідйому, цементування, ліквідації нафтогазопромислових явищ.

Програмне забезпечення тренажера містить засоби проектування навчальних завдань із будь-якими початковими умовами виконання провідки свердловин: характеристиками продуктивного пласта, конструкцією свердловини, набором устаткування й інструменту, технологій виконання основних операцій, нестандартними ситуаціями. Передбачені і засоби контролю й оцінки тих, хто навчається, ведення персональних журналів проходження навчального процесу, формування протоколу навчання. Викладач має можливість урізноманітнити

встановлений хід виконання навчальних завдань імітацією різних нештатних ситуацій та ускладнень. При імітації технологічних процесів на екран монітора виводяться числові характеристики умов імітованого процесу, графіки найважливіших контрольованих технологічних параметрів, а також анімації, що відображають у реальному часі роботу обладнання, інструменту та стан свердловини.

Імітація роботи обладнання супроводжується звуком. Тренажер дозволяє побачити приховані від прямого спостереження процеси, спостерігати процеси виникнення і розвитку ускладнень та аварійних ситуацій. Це дозволяє набути й удосконалити практичні навички з виконання, контролю та оптимізації основних технологічних процесів, розпізнавання та запобігання ускладнень і аварійних ситуацій, ліквідації нафтогазопроявів і викидів. Крім одиночного виконання, існує виконання навчальним класом, яке дозволяє набути навички колективної роботи в складі бурової бригади.

Базова поставка навчального класу АМТ-231 містить ліцензію на 15 робочих місць і дозволяє здійснювати індивідуальне та групове навчання, обслуговування до декількох десятків робочих місць і забезпечення контролю дій тих, хто навчається, з робочого місця інструктора.



Рисунок 1.1 - Тренажер-імітатор буріння АМТ-231

Викладач у «навчальному класі» має такі можливості: варіювання форми проведення занять за допомогою режимів – окреме завдання для кожного, спільне завдання для всіх і режим навчання «дивись на мене»; повний контроль роботи всіх учнів зі свого робочого місця; призначення завдання і організація колективної

роботи; урізноманітнення встановленого ходу виконання навчальних завдань імітацією різних нештатних ситуацій та ускладнень:

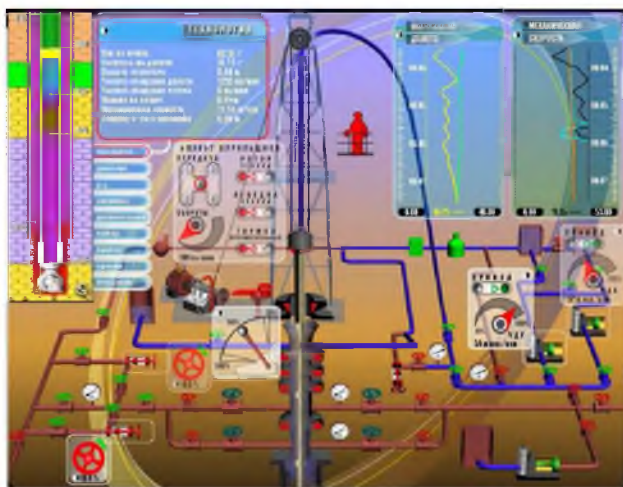


Рисунок 1.2 - Інтерфейс оператора тренажера-імітатора буріння АМТ-231

Тренажер-симулятор буріння DART. «КСА DEUTAG» є розробником тренажера буріння «DART» (Drilling and Advanced Rig Training) - комп'ютерної системи, що дозволяє відпрацьовувати навички буріння та різноманітні інженерні нештатні ситуації в режимі реального часу з застосуванням технології внутрішньо-свердловинного моделювання. Використання 3D-графіки, симуляція в режимі реального часу і звукові спецефекти дозволили тренажеру «DART» отримати визнання як затребуваного навчального тренажера в нафтогазовій індустрії.



Рисунок 1.3 - Тренажер-симулятор буріння DART

Тренажер забезпечує відпрацювання як практичних навичок роботи на бурових установках, так і сценаріїв оптимізації бурових робіт. Система DART

використовується програмне забезпечення, яке ефективно інтегрується і відтворює свердловинні умови. В такий спосіб забезпечується практика з використанням реальних проектних даних щодо свердловин як для початківців, так і для досвідчених працівників. Такий підхід до автоматизованого навчання сприяє часовій оптимізації при запуску нових бурових установок.

Тренажер характеризується такими специфікаціями:

- забезпечення реалістичності практичних рішень у процесі навчання та підготовки бурових бригад до початку роботи на новій буровій установці або початку нового проекту з буріння;
- можливість пробурити віртуальну свердловину за заданими проектними параметрами та виявити потенційні проблеми;
- сприяння підвищенню ефективності комунікацій і згуртованості бурових бригад.

Серед клієнтів, які скористалися навчальними програмами тренажера «DART», компанії «BP», «CNR», «Shell», «ТНК-ВР» і Університет імені Роберта Гордона.

Тренажер-імітатор капітального ремонту свердловин АМТ-411. Тренажер призначений для навчання робочого й інженерного персоналу підрозділів капітального ремонту свердловин нафтогазовидобувних підприємств, а також для студентів спеціальності буріння свердловин, розробка й експлуатація нафтових і газових родовищ. Апаратно-програмний комплекс тренажера складається з пультів і постів управління обладнанням для капітального ремонту свердловин, персонального комп'ютера та програмного забезпечення.



Рисунок 1.4 - Тренажер-імітатор капітального ремонту свердловин АМТ-411

Тренажер імітує в реальному та прискореному масштабі часу технологічні процеси капітального ремонту свердловин: глушіння свердловини, ремонтне цементування, розбурювання, спуск-підйом, ліквідацію нафтогазопроявів при бурінні, обробку привибійної зони, освоєння свердловини свабом, освоєння свердловини компресором, гідророзрив пласта, гідропіскоструменеву перфорацію:



Рисунок 1.5 - Інтерфейс оператора тренажера-імітатора капітального ремонту свердловин АМТ-411

Як для випадку АМТ-231, існує виконання «навчальний клас» з аналогічними характеристиками.

Тренажер-імітатор експлуатації й освоєння свердловин АМТ-601. Навчальний тренажер призначений для інженерного персоналу видобувних підрозділів нафтогазовидобувних підприємств, а також студентів спеціальності

розробка й експлуатація нафтових і газових родовищ. Тренажер імітує в реальному та прискореному масштабі часу такі технологічні операції:

- освоєння і вивід на режим;
- дослідження на встановленому та невстановленому режимі;
- експлуатація в ускладнених умовах.

Тренажер дозволяє проводити навчання для роботи з:

- нафтовими свердловинами;
- фонтанним способом;
- за допомогою відцентрових насосів;
- за допомогою штангових насосів;
- газліфтним методом;
- газовими свердловинами;
- нагнітальними свердловинами;

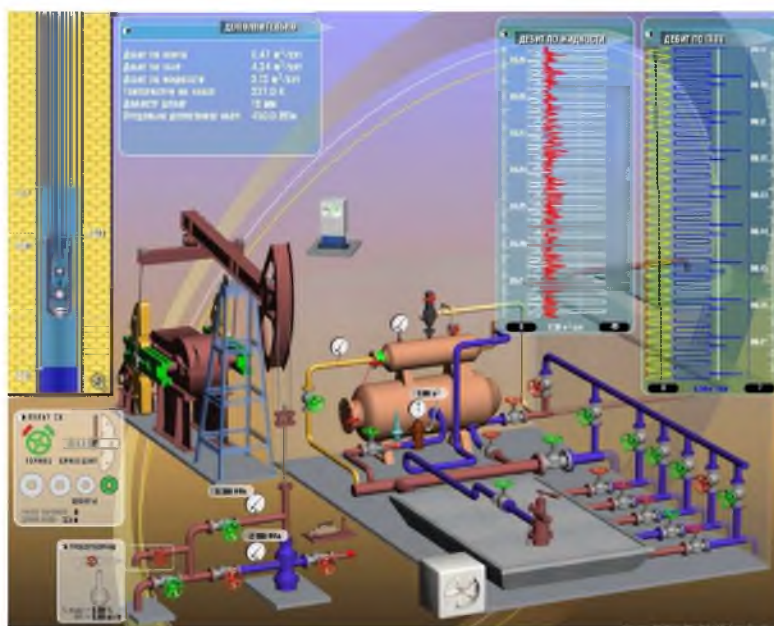


Рисунок 1.6 - Тренажер-імітатор експлуатації та освоєння свердловин АМТ-601

Програмне забезпечення тренажера містить засоби проектування навчальних завдань певними початковими даними свердловин (типом свердловини, характеристиками продуктивного пласта, властивостями флюїду, набором устаткування, нештатними ситуаціями), а також засоби контролю та оцінки дій тих,

хто навчається, ведення персональних журналів перебігу навчального процесу, формування протоколу навчання.

Цей тренажер є ефективним технічним засобом навчання та підвищення кваліфікації працівників видобувних підприємств. дозволяє набути та вдосконалити практичні навички виконання з дотриманням регламенту робіт, дослідження свердловин, розпізнавання та запобігання ускладнень і аварійних ситуацій, отримати навички розуміння фізичних процесів у свердловині. Базова поставка навчального класу АМТ-601 передбачає ліцензію на 15 робочих місць.

Тренажер технологічної платформи «Транзас». Це обладнання, призначене для відпрацювання операцій, безпосередньо пов'язаних із добуванням нафтових запасів, підготовкою нафти для транспортування, зберіганням нафти в танках кесона та періодичним відвантаженням її на танкери.



Рисунок 1.7 - Тренажер технологічної платформи «Транзас»

У тренажері змодельовані такі системи:

- система зберігання та відвантаження нафти;
- система забортної води;
- система сепарації нафти;
- система компримування газу;
- система поглинального газу;

- система підготовки паливного газу;
- система утилізації пластової води;
- система закачування води для підтримки ППД;
- поведінка свердловини;
- система гирла свердловин.

Новий тренажер технологічної платформи входить до складу інтегрованого комплексу «Видобуток-Шельф». Він призначений для підготовки фахівців із освоєння континентального шельфу, містить тренажери установок морського базування, тренажери точок відвантаження і танкерів арктичного класу [8-33].

Тренажер нафтоналивного терміналу. Компанією «Транзас» спільно з «Kuumenlaakso Polytechnic» було розроблено тренажер технологічних операцій багатопільового терміналу, який використовується для відпрацювання принципів завантаження танкерів. Тренажер дозволяє відпрацьовувати взаємодію оператора терміналу та вантажних помічників танкера:



Рисунок 1.8 - Тренажер нафтоналивного терміналу

Тренажер вантажних операцій нафтоналивного терміналу призначений для навчання операторів терміналу – вантажного майстра й інженера паркової зони. Основною метою підготовки є тренінг і відпрацювання навиків з диспетчерського управління для набуття навиків з управління, розподілу ємностей парку; здатності відстежувати та контролювати всі операції, що відбуваються в системах терміналу;

здатності адекватно реагувати на технологічні та погодні зміни. При цьому необхідно враховувати технологічні можливості систем, оперативно оцінювати можливі наслідки своїх дій, виявляти й аналізувати зміни в ланцюгу технологічних операцій із урахуванням нових обмежень та можливостей.

Вправи завантаження / розвантаження реального танкера для терміналу можуть бути об'єднані в один технологічний цикл або використовуватися як окремі вправи для вивчення конкретних операцій із застосування окремих систем або їх елементів. Тренажер дозволяє завантажити вправу, яка описує початковий стан і наближений сценарій підготовки. Інструктор може змінювати в довільному порядку хід виконання вправи і додавати різні несправності й обмеження. Виконання завдання припиняється лише при виведенні з ладу найважливішого устаткування або при досягненні стану «екологічна катастрофа».

Для забезпечення ефективного тренінгу прототипом тренажера було обрано багатоцільовий термінал (місткістю 700 000 м³), здатний обробляти «VLCC, LCC» танкери. Технологічні можливості тренажера повністю охоплюють цикл роботи терміналу. Змодельовані параметри можуть доповнюватися пожежними системами, а також системами станції вагонів розвантажувальних / завантажувальних систем:



Рисунок 1.9 - Інтерфейс тренажера наповнювального терміналу

Тренажер використовує універсальну математичну модель, побудовану на принципі єдності топологічної розрахункової схеми систем терміналу і танкера, що

утворюють єдину трубопровідну систему. Застосування графових алгоритмів дозволяє автоматично розділяти розрахункову область на окремо вирішувані блоки. В основі рівняння моделі – базові закони збереження маси, енергії й рівняння стану, які дозволяють враховувати всі енергетичні ефекти, в тому числі фазові переходи при зміні параметрів навколишнього середовища.

Крім того, застосовується візуалізація, що імітує 32 зовнішні камери відеонагляду, розташовані в найважливіших місцях терміналу (як це робиться на реальних аналогах і нових танкерах). Це наближає навчальне місце тренажера до умов реального посту управління терміналу, адже саме відеоспостереження дозволяє оцінити параметри роботи, неконтрольовані системою управління, наприклад, протікання в трубопроводах і цистернах:



Рисунок 1.10 - Імітація зовнішніх камер відео спостереження

Впровадження зовнішнього відеоспостереження створює нову концепцію навчання, спрямовану на підготовку фахівців для роботи з сучасними системами управління, розрахованими на невеликий штат персоналу.

КТК-М (комп'ютерний тренажерний комплекс багатofункціональний). КТК-М був розроблений раніше існуючою російською компанією «Петроком»,

(сьогодні це частина «Honeywell») – компанії-лідера з упровадження тренажерних комплексів у світі.

Існують три види тренажерних продуктів, запропонованих «Honeywell» на платформі КТК-М, для кожної з яких використовується своя тренажерна модель. Спеціалізована тренажерна модель – це модель ТП, яка повною мірою відображає специфіку конкретної установки – технологічні апарати, обв'язку, систему управління. Така модель достатньо точно відтворює динаміку реального ТП. Вимоги щодо точності повною мірою реалізуються операторським інтерфейсом: тренажер повністю відтворює операторський екран розподіленої системи управління.

Серед переваг таких спеціалізованих тренажерів виділяють такі: використання як випробувального полігону для нових систем управління, застосування для переведення діючої системи управління зі щитової на комп'ютерну.

Окрім того, спеціалізовані моделі дозволяють швидко й ефективно перевірити складні системи блокувань з розгалуженою логікою, виявити і усунути помилки, непомічені під час проектування та реалізації, і, тим самим, прискорити введення в експлуатацію модернізовані установки. Тобто, якісна спеціалізована модель допомагає не лише інженерному, але й технологічному забезпеченню процесу.

На малюнку 1.11 представлений один із операторських екранів моделі «типової» установки каталітичного риформінгу (типу Л-35-11 / 600). Ця модель охоплює блоки гідроочищення, відпарки, риформінгу і стабілізації, системи подачі паливного газу та зворотної води. Складається з 3 колон, 5 реакторів, 2 печей, поршневих і відцентрових компресорів, теплообмінників, повітряних і водяних холодильників, насосів та ін.

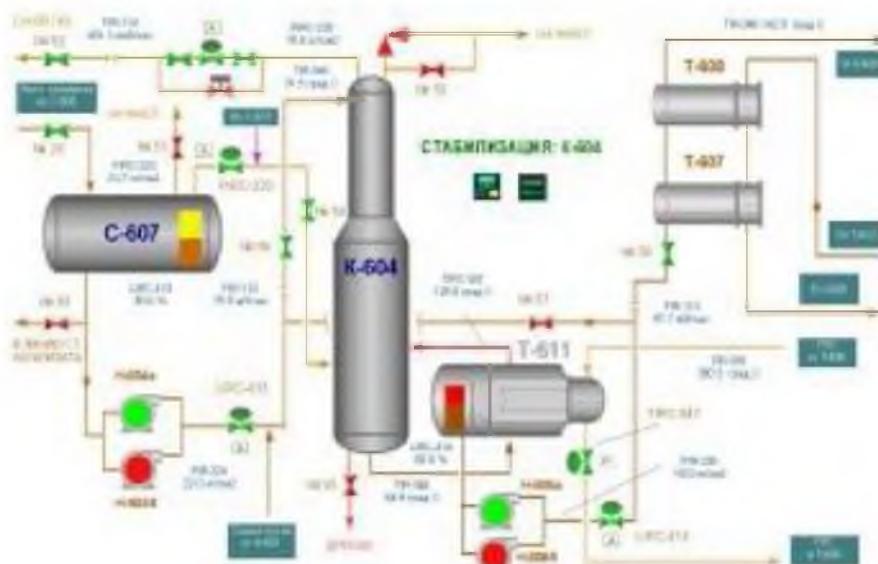


Рисунок 1.11 - Операторський екран моделі «типової» установки каталітичного реформінгу (типу Л-35-11/600)

Ще одним прикладом типової установки первинної переробки нафти є модель АВТ-6, яка охоплює блоки підігріву сировини та знесолення, колону К-1, атмосферну колону зі стріппінгами, атмосферні та вакуумні печі, вакуумну колону з системою для утворення вакууму системою, весь блок стабілізації та поділу бензинів.

За допомогою бібліотеки моделей типових процесів нафтопереробки можна змоделювати такі процеси:

- первинна переробка нафти (ЕЛОУ-АВТ);
- каталітичний риформінг бензинів (ароматичного або паливного профілю);
- гідроочищення дистилатів;
- фракціонування граничних газів (ДФУ);
- вісбрекінг гудрону;
- виробництво водню;
- виробництво сірки;
- пропанова деасфальтизація масел;
- уповільнене коксування (установка з 4-ма коксовими камерами) тощо:

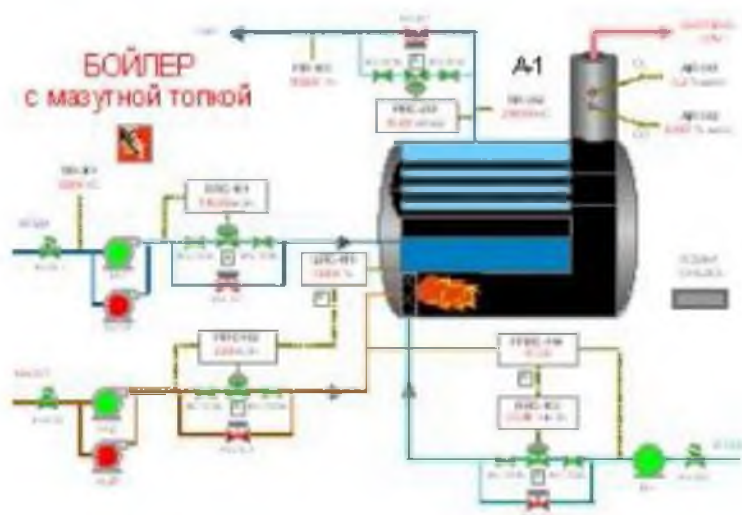


Рисунок 1.12 - Операторський інтерфейс моделі «Бойлер з мазутною топкою»

Для операторів, які пройшли необхідну теоретичну підготовку в навчальних класах, але не набули виробничого досвіду та не навчалися раніше на комп'ютерних тренажерах великих технологічних установок, «Honeywell» пропонує моделі базових технологічних вузлів і апаратів – типові елементи технологічного устаткування та систем управління. КТК-М працює на стандартних IBM PC-сумісних комп'ютерах і не вимагає спеціального обслуговування та інженерного супроводу.

UniSim. Система «UniSim» є розробкою «Honeywell» і є набором уніфікованих засобів моделювання, що дозволяють поліпшити функціонування підприємства протягом всього життєвого циклу - від проектування технологічних процесів із використанням статичних і динамічних моделей, перевірки системи управління та навчання операторів до управління й оптимізації, моніторингу та планування роботи підприємства в режимі реального часу.

«UniSim» забезпечує істотно розширені можливості «off-line» і «on-line» моделювання для проектування, інжинірингу, навчання операторів й оптимізації технологічних процесів. Користувачам системи надано можливості збору та доступу до знань про технологічний процес для забезпечення максимально вигідних умов повернення інвестицій в технології моделювання, що підвищує рентабельність підприємства.

Програмний пакет «UniSim Operations» – це синергія попереднього рішення «Honeywell, Shadow Plant» (його найкращі характеристики) та програмного забезпечення для імітаційного моделювання, розробленого компанією «AspenTech». Програмний пакет використовує розподілену архітектуру, для зняття обмежень, пов'язаних із нестачею обчислювальних ресурсів. При моделюванні в реальному часі це забезпечує: більш високу точність (як при навчанні операторів, так і в інжинірингу); можливість моделювання складних технологічних об'єктів, що складається із кількох установок; можливість відтворення складних додатків (у т.ч. системи управління).

Основними компонентами Тренажера «UniSim» є:

- платформа «UniSim»;
- робоче місце інструктора;
- робочі місця операторів.

Платформа «Unisim» – це базовий механізм, що забезпечує управління та синхронізацію різних завдань і прикладних програм для проведення високоточного імітаційного моделювання великомасштабних процесів у режимі реального часу. Завдання і програмні рішення, зазвичай, розподіляються між кількома ПК.

Модель технологічної установки дозволяє з високою точністю відтворити поведінку розподіленої системи управління (PCU) і системи протиаварійного захисту; в цьому її ключова відмінність від моделі технологічного процесу. Моделі деяких PCU можна емулювати в середовищі «UniSim», в іншому випадку середовище моделювання стимулюється через контролери, надані постачальниками PCU.

Програмне забезпечення «UniSim Design» (нова версія продукту R390 вийшла навесні 2009 року) містить останню версію «HYSYS». Використовувана не як стандартний інтерфейс оператора в складі «Unisim Operations», пропонує статистичне та динамічне моделювання, дизайн, контроль якості роботи, оптимізацію і бізнес-планування для підприємств з видобування і переробки нафти і газу, а також для підприємств хімічної промисловості:

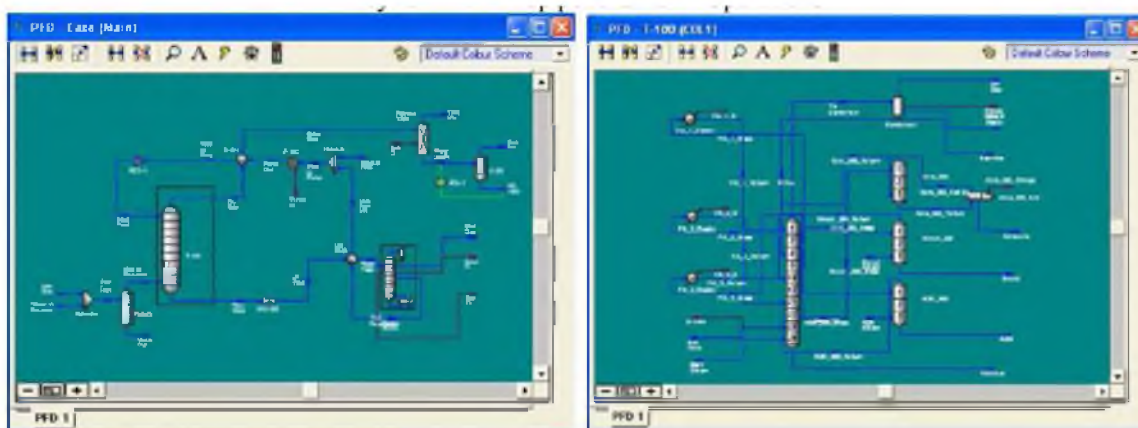


Рисунок 1.13 - Інтерфейс користувача «UniSim Design»

«Бібліотека» тренажера налічує безліч спеціалізованих моделей технологічних установок нафтопереробки, нафтохімії та хімії, в тому числі складні установки первинного розподілу нафти, гідроочищення, каталітичного риформінгу, каталітичного крекінгу й інші процеси різних ліцензіарів. Окрім того, оскільки відомий технологічний ліцензіар - компанія «UOP» - є дочірньою компанією «Honeywell», бібліотека «UniSim Design» має ексклюзивні динамічні моделі ТП розробки «UOP» (Unionfining, Platforming, Tataray, Isomar, Parex, Molex, CycleMAX, Unioncracking, Penex, Alkylolation та ін.).

Також «UniSim Design» підтримується повний набір моделей технологічних апаратів, у тому числі дистиляційні колони, реактори, теплообмінне обладнання, ротаційне обладнання; роботу з твердими речовинами і в змішаних середовищах (вібраційне сито, рукавний фільтр-пилловловлювач, центрифуга, конвеєр, дробарка і ротаційний подрібнювач). Модель дає великий вибір логічних елементів, елементів конфігурації системи управління, що дозволяє моделювати різноманітні технологічні ситуації.

Тренажер «Управління в нафтогазовій галузі». Тренажер є розробкою лабораторії інформаційних управлінських систем «ЛАБІУС» і прикладом тренажера першого рівня - рівня підприємства. Тренажер «Управління в нафтогазовій галузі» імітує процес реалізації проекту з розробки великомасштабного нафтового родовища.



Рисунок 1.14 - Тренажер «Управління в нафтогазовій промисловості»

Процес освоєння родовища в тренажері передбачає вирішення учасниками таких завдань:

- геологорозвідка ділянок родовища;
- управління проектною командою та забезпечення персоналом нафтопромислу;
- буріння та будівництво свердловин;
- організація проектних робіт;
- будівництво об'єктів видобутку, очищення та підготовки нафти, для забезпечення інфраструктури;
- організація логістики та доставки вантажів;
- взаємодія з постачальниками та підрядниками;
- енергетичне забезпечення нафтопромислу;
- будівництво нафтопроводу та транспортування нафти;
- охорона праці, промислова безпека й охорона навколишнього середовища.

При проходженні тренажеру учасники навчання стикаються з факторами, що ускладнюють процес реалізації проекту: природно-кліматичними чинниками, геологічними ризиками, зміною цін на ринках енергоносіїв, негативним впливом ненадійних підрядників, скороченням об'ємів фінансування проекту, зміною державної політики тощо.

Окрім простих тренажерних навчальних комплексів існують й автоматизовані системи навчання. Якість прийнятих оператором рішень безпосередньо залежить від

рівня володіння двома базовими навиками, необхідними для безпечного й ефективного управління технологічним процесом: навиком прогнозування наслідків порушення нормального ходу процесу (методика «Що станеться, якщо ...») і визначення можливих причин порушень (методика «Генерація можливих причин відхилення від норми»).

Обидві методики ґрунтуються на базі знань (БЗ), сформованої узагальненням типових порушень процесу, аналізом цих порушень на реальних технологічних установках, моделюванням ситуацій на комп'ютерному навчальному тренажері та залученням експертних оцінок фахівців.

Прикладом автоматизованої системи навчання в предметній області є автоматизована система навчання (АСН) «АФОН». Система «АФОН» автоматизує тренування та тестування за двома навиками. Залежно від типу навика, якому навчають, пропонується обрана випадковим чином причина або симптоматика. Учасник навчання має оцінити зміну параметрів, або, навпаки, вибрати причини виникнення запропонованих симптомів. Відповіді оцінюються, а на стадії тренування пропонується правильна відповідь. Передбачено налаштування параметрів тестування і детальне протоколювання іспитів.

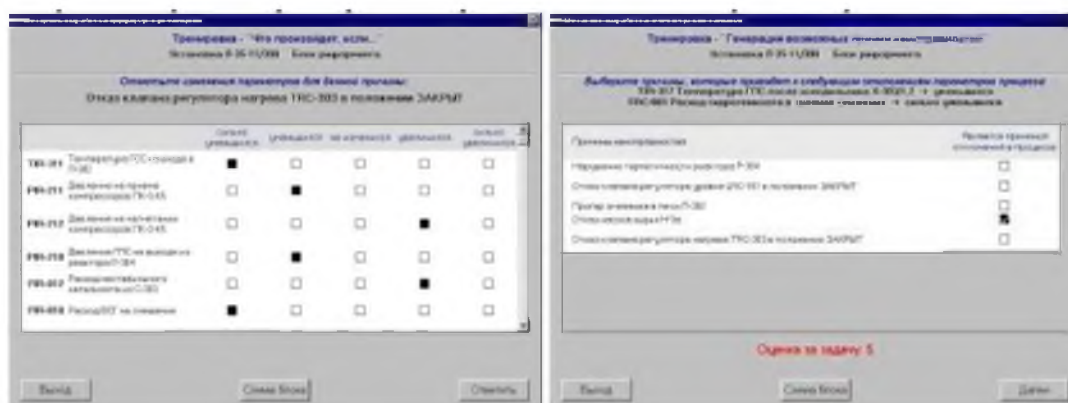


Рисунок 1.15 - АСН«АФОН». Методика «Що станеться, якщо?» і методика «Генерація можливих причин несправностей»

АСН «ДИАГНОСТ». Автоматизована система навчання «ДИАГНОСТ» призначена для вироблення в оператора вміння ефективно та надійно визначати причини несправностей за характером зміни параметрів технологічного процесу.

Навчання реалізується в формі «гри» з оператором, в ході якої «ДИАГНОСТ» «загадує» одну з причин і робить «початкове вкидання»: повідомляє оператору про зміну одного з параметрів процесу внаслідок заданої причини. Завдання оператора - знайти пропоновану причину запитам змін інших параметрів із запропонованого списку, причому зробити це необхідно за мінімальну кількість кроків. Користувач легко доповнює загальний список причин новими, вводячи, таким чином, в базу знань (БЗ) і в предметну область навчання нові змістовні ситуації порушення ходу технологічного процесу. Також легко формуються окремі ігри, сукупності можливих причин і пов'язаних із ними симптомів, що пропонуються оператору. Для кращого розуміння завдання, пропонуються підказки у вигляді технологічних схем, описів процесу, а також оцінок зробленого інформаційного запиту.

АСН «ПЛАС +». Програмний комплекс «ПЛАС +» – це відкрита, розширювана користувачем автоматизована система навчання. До неї входить:

- електронний підручник «PlasView», який містить принципову технологічну схему та план-схему розміщення обладнання, а також комп'ютерну версію блок-карт технологічних блоків;

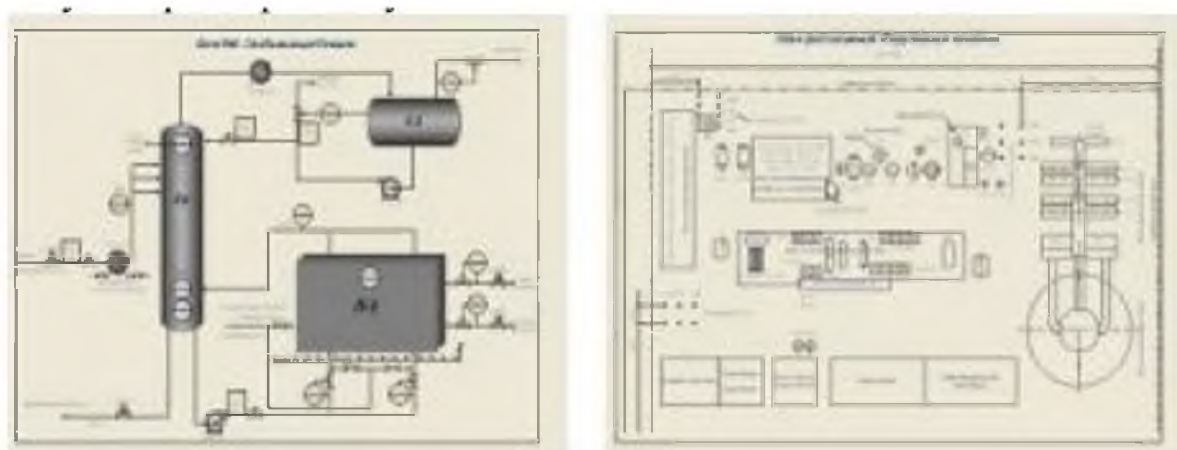


Рисунок 1.16 - АСН «ПЛАС +». Принципова технологічна схема і план-схема розміщення устаткування

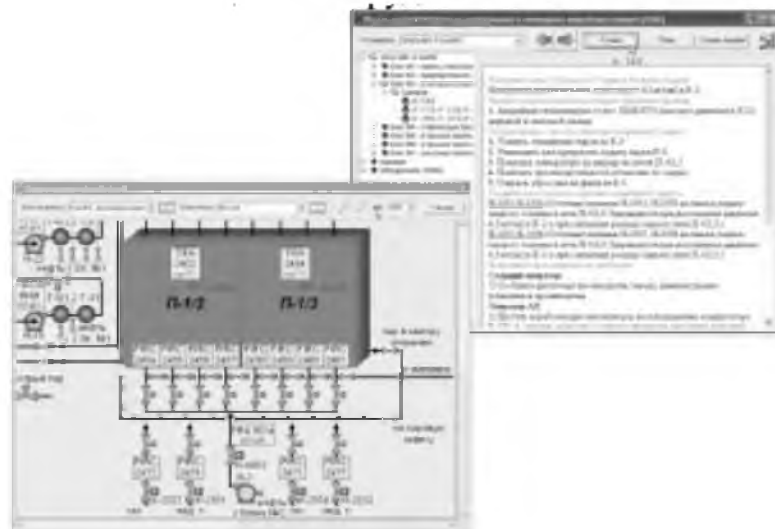


Рисунок 1.17 - АСН «ПЛАС +». Електронний посібник

- модуль перевірки знань «PlasTest». У режимі тренування цей компонент системи дозволяє учаснику навчання практикуватися, вибираючи правильні елементи оперативної частини «ПЛАС» (передумови та розпізнавальні ознаки аварій, оптимальні способи протиаварійного захисту, технічні засоби протиаварійного захисту, виконавці та порядок їх дій) для різних аварійних ситуацій:



Рисунок 1.18 - АСН «ПЛАС +». Перевірка знань з «ПЛАС +».

- модуль налаштування параметрів тестування «PlasOption», для коригування фахівцям служб промислової безпеки та підготовки персоналу системи оцінювання іспиту;
- модуль редагування «PlasEdit», для оновлення всіх структурних елементів блок-карт (аж до створення ПЛАС'ів «з нуля»):



Рисунок 1.19 – АСН «ПЛАС +». Кадри редагування оперативної частини «ПЛАС»

АСО «ТРЕНАРІЗ». Комп'ютерні тренажери, призначені для підготовки оперативного персоналу до виконання таких процесів:

- пуску технологічного процесу;
- роботи в передбачених регламентом режимах;
- планової й аварійної зупинки процесу;
- управління процесом у передаварійних і аварійних режимах;
- виконання дій з локалізації аварій.

Комп'ютерний тренажер – це програмно-апаратний комплекс, який створюється для конкретного технологічного об'єкта на основі аналізу небезпеки. Результати аналізу використовуються розробником для створення баз знань і конструювання тренажерного комплексу за допомогою спеціального програмного засобу «Дизайнер».

Комп'ютерний тренажер створюється для кожного технологічного об'єкта і дозволяє організувати навчання та тренування персоналу технологічної установки в дві стадії:

- навчання та тренування логіки дій з локалізації та ліквідації аварій за технологічними схемами (рисунок 1.20 а);
- навчання, вироблення моторних навичок дій операторів і їх взаємодії при аварії в реальному просторі об'єкта, імітованого комп'ютером (рисунок 1.20 б):

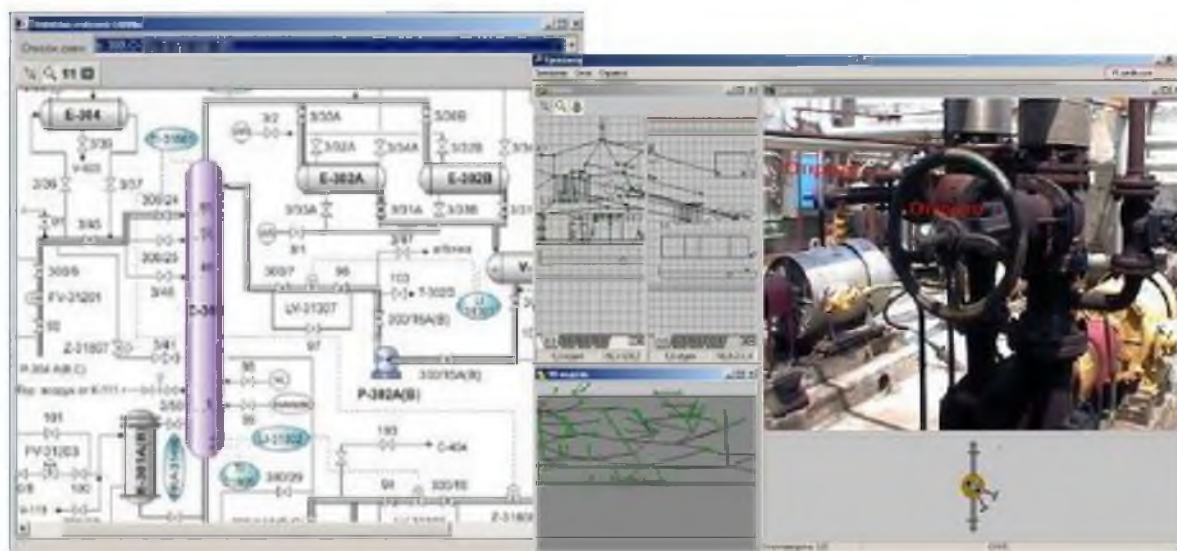


Рисунок 1.20 – Тренажер з технологічними схемами (а)
і в імітованому просторі установки (б)

Підсумовуючи викладений аналіз історії та сучасного стану бурових імітаційних навчальних тренажерів та автоматизованих навчальних систем, можемо впевнено стверджувати про необхідність та ефективність застосування, подальшого розвитку й удосконалення інтелектуальної й апаратної підтримки процесів прийняття рішень при виконанні технологічних операцій процесів буріння та обслуговування нафтових і газових свердловин шляхом ефективною реалізації в навчальному процесі [34-52].

Для дослідження виокремленої наукової задачі наві буде розроблена й обґрунтована предметна модель інтелектуальної інформаційної системи в контексті статистичної та динамічної поведінки. Дослідження здійснено методами систематизації й узагальнення, порівняння автоматизованих систем-тренажерів та

симуляторів, аналізу структури прийняття рішень в технологічних рішеннях, системного підходу, формально-логічних досліджень, методів теорії множин, методів теорії та задоволення обмежень, теорії ймовірності, математичної статистики.

1.2 Моделювання основних компонент автоматизованих тренажерів

Користувачі (об'єкти тренінгу) розподілені автоматизованих навчальних систем (АНС) можуть використовувати режим роботи, який називається віртуальною аудиторією. В даному режимі користувачі працюють за однаковою навчальною стратегією *LearningStrategy*, виконуючи однакові навчальні процедури *LearningRoutines* і при цьому взаємодіючи один з одним. Цей метод реалізовує технологію навчання (тренінгу) шляхом інтерактивної взаємодії *LearningbyInteraction*.

В АНС [53-61] використовуються методи візуалізації описових характеристик технологічних завдань і технологічних проблем, наприклад, показник диференціації рівня складності завдання відносно до рівня знань об'єктів навчання (тренінгу).

Особливості інтелектуальних навчальних систем полягають у використанні методів штучного інтелекту (зокрема, експертних систем), побудованих на основі графових представлень логічних зв'язків *LogicalDependencies* і залежностей між навчальними елементами *LearningElements* (курси тренінгу, технологічні проблеми, теми тренінгу, розділи тренінгу, змістовні модулі, електронні підручники, електронні посібники, електронні енциклопедії, електронні словники). Таке представлення описується матрицею логічних зв'язків *LogicalDependenciesMatrix*

Знання *Knowledges* є результатом навчальної діяльності об'єкта тренінгу *TraineeLearningActivities* у навчальному середовищі *LearningEnvironment*, створеному АНС. Контент тренінгового матеріалу представлений у вигляді парадигми: контентні дані (*ContentData*) – контентні знання (*ContentKnowledges*) – контентні мета знання (*ContentMetaKnowledges*). Крім того дані та знання можна

класифікувати як: дедуктивні, індуктивні, інференційні, фактологічні, процедурні, методичні, концептуальні, описові. З точки зору методики представлення знання можуть містити онтологічні *OntologyDescriptors*, структурні *StructureDescriptors*, синтаксичні *SyntacticalDescriptors*, семантичні *SemanticalDescriptors* і концептуальні *ConceptualDescriptors* дескриптори. Об'єкти даних *DataObjects*, об'єкти знань *KnowledgeObjects*, об'єкти метазнань *MetaknowledgeObjects* представлені у вигляді багаторівневих словників *MultiLevelDictionaries*, *MultiLevelThesaurus*. Це забезпечує навігацію онтологіями понять, термінів і концепцій предметної області *SubjectDomainOntologys*. Під час роботи з АНС (засобами її інтерактивного інтерфейсу) об'єкти навчання (тренінгу) інтерпретують контентні дані, контентні знання та контентні мета знання, отримуючи відповідне представлення (когнітивне відображення *CognitiveReflections*) у формі предметних знань, умінь і навичок.

Таким чином, у даному контексті АНС відіграє роль медіатора між тьютором-людиною (з відповідною множиною знань, умінь і навичок) та об'єктом тренінгу, який у результаті інтерактивної взаємодії з АНС за участю тьютора-викладача одержує відповідні когнітивні представлення. Проте результат навчальної діяльності оператора в режимі тренінгу не є результатом перенесення ідеальних профілів тьютора-викладача у формі когнітивних представлень об'єкта тренінгу. Це зумовлюють такі чинники:

1. Складність формалізації знань, представлених природною мовою експерта *ContentExpert* предметної області (російська, українська, англійська тощо).
2. Залежність знань експерта від досвіду його роботи в предметній області, що визначається такими характеристиками як повнота та достовірність досвіду, які важко формалізувати.

В АНС властивості об'єктів навчання в формі технологічного тренінгу *LearningSubjectsFeatures* співставляються з ідеальними значеннями, і результат співставлення використовується як шаблонна одиниця вимірювання та особливостей навчального процесу *LearningProcess*. Модель компетентності

CompetenceModel визначає ефективність дій об'єктом тренінгу *EffectiveTraineeAction* у виділеній предметній області та базується на основі висококласифікованих знань *HighProfiledKnowledges*, спеціалізованих умінь і навичок. Модель об'єкту тренінгу містить такі основні складові: *LearningActivitiesGoal* - мета та завдання навчання (тренінгу); *TraineeKnowledgesLevel* - рівень знань об'єкта тренінгу при вивченні ним поточного курсу тренінгу, теми, підтеми, змістовного модуля; *SelectedLearningStrategie* - вибрана навчальна стратегія вивчення контенту навчального матеріалу з послідовністю генерації тестових питань, навчальних завдань, навчальних проблем; *TraineeFeedback* – технологія генерації зворотного зв'язку з об'єктом *htysyue* з визначеними методами, засобами та правилами оновлення його системного профілю *TraineeProfile*.

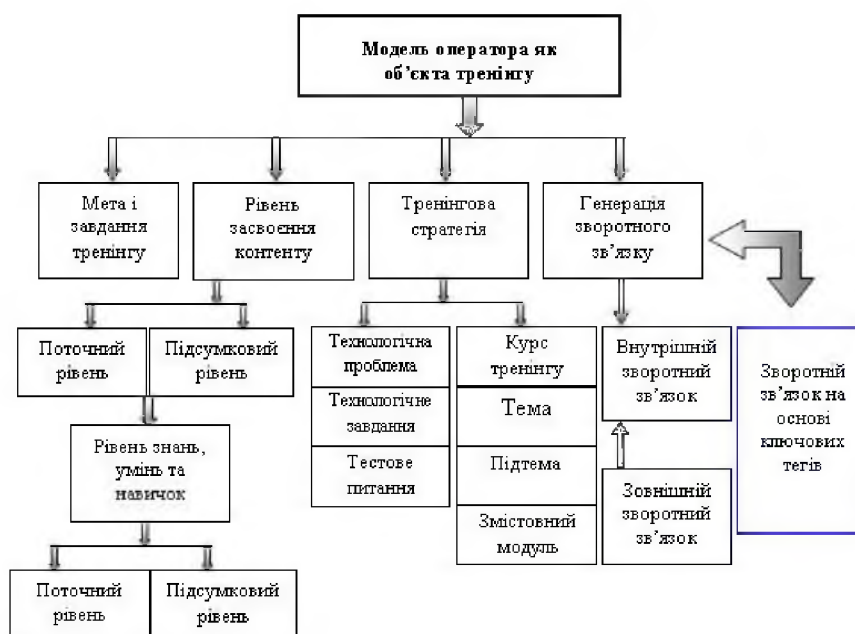


Рисунок 1.21 – Структурна схема моделі об'єкта зворотного зв'язку

В інтелектуальних АНС на основі обмежень модель об'єкта тренінгу містить множини релевантних, порушених і задоволених обмежень. Після реєстрації об'єкту навчання система розпізнає його системний профіль *TraineeProfile* і виконує ініціалізацію мети роботи з системою в поточній сесії *LearningGoal*. Вона має

відповідати цілі стратегії тренінгу *GoalDeducedStrategy*, із визначеною підмножиною контенту навчального (тренінгового) матеріалу *LearningContentSubset* і наперед визначеною послідовністю контрольних вправ, завдань й технологічних проблем. Це відповідає початковим налаштуванням системи *InitialSetting*, встановленими на основі профілю об'єкту тренінгу. Технологія, методи та засоби побудови моделі об'єкта тренінгу залежать від виду навчальної (тренінгової) системи. Доцільно розглянути такі випадки:

1. В класичній навчальній системі модель об'єкта навчання утворюється на основі множини системних параметрів, що дозволяють контролювати ступінь засвоєння контенту навчального матеріалу за виділеною темою, підтемою, навчальним курсом, змістовним модулем і, відповідно, адаптивно вибрати оптимальну навчальну стратегію, контролювати ефективність її застосування на основі параметру поточного та підсумкового рівня знань, умінь і навичок об'єкту навчання;

2. В інтелектуальних навчальних системах, крім множини системних параметрів контролю ступеня засвоєння контенту навчального матеріалу, використовується також множина правил, які дозволяють програмувати та контролювати роботу об'єкта навчання в навчальному середовищі системи. Тобто на основі множини правил здійснюватиметься вибір та імплементація навчальних стратегій, генерація та контроль зворотного зв'язку системи з об'єктом навчання тощо;



Рисунок 1.22 – Контроль моделі об'єкта навчання

3. В інтелектуальних навчальних систем на основі підходу представлення та задоволення обмежень множина правил замінюється множиною обмежень, що забезпечують виконання описаних вище завдань, а також підтримки інтелектуальності системи.

Таким чином, модель об'єкту тренінгу є ефективною з точки зору імплементації якщо вона контролюється множиною правил або множиною обмежень. При цьому така множина правил або обмежень може бути предметно залежною (прив'язаною до певної предметної області) або предметно незалежною. Такий підхід дозволяє адаптивно та динамічно оновлювати поточну модель об'єкта тренінгу на основі аналізу його поточних і підсумкових профілів, що забезпечує адаптивність та інтелектуальність АНС у цілому.

В АНС курс тренінгу є деякою скінченою послідовністю навчальних тем *LearningTopics*, які, відповідно, можуть розбиватися на підтеми, а відповідно сукупності завершених навчальних тем можуть утворювати змістовні модулі, що є особливо важливими в умовах упровадження Болонського процесу. Таким чином, основним питанням ефективної імплементації структури навчального матеріалу *LearningContentStructure* у навчальне середовище АНС є визначення доцільної й ефективною послідовності вивчення введених тем навчального курсу на основі аналізу та дослідження логічних зв'язків між ними та формування на цій основі відповідних навчальних стратегій. Це є необхідною складовою АНС, особливо їх інтелектуальних версій. Технічно послідовність вивчення тем і логічні зв'язки між ними представляються у вигляді моделі засвоєння контенту тренінгового матеріалу *ContentMasteringModel*. Визначення та дослідження логічних зв'язків між темами тренінгового матеріалу може виконуватися і як частина процесу концептуалізації та структуризації предметної області *SubjectDomain* (при використанні інтелектуальних АНС). При визначенні структури й обсягу навчальних тем класичним підходом є представлення навчальної теми як послідовності (сукупності) навчальних елементів *LearningElements*. У класичній теорії навчання навчальний

елемент розглядається як цілісні, структуризовані та завершені фрагменти навчального матеріалу. Використання навчальних елементів у АНС найбільш продуктивно представлене в інтелектуальних версіях, де навчальні елементи можуть бути зведені до рівня концептів предметної області *SubjectDomainConcepts*. Це дозволяє найбільш повно реалізувати характеристики цілісності, неподільності та завершеності. Таким чином, реалізація навчальних елементів в АНС має онтолічний характер і виконується у вигляді ряд кроків:

1. **крок 1** – визначення множини концептів предметної області;
2. **крок 2** – введення навчальних елементів на рівні визначених концептів або шляхом їх поєднання;
3. **крок 3** – визначення логічних зв'язків між введеними навчальними елементами;
4. **крок 4** – візуалізація отриманих логічних зв'язків між навчальними елементами у вигляді графового представлення;
5. **крок 5** – числове представлення логічних зв'язків у формі матриці *LogicalDependenciesMatrix*;
6. **крок 6** – побудова матриці черговості навчальних елементів на основі матриці логічних зв'язків;
7. **крок 7** – формування навчальної стратегії як зв'язаного переліку навчальних елементів, побудованої на основі матриць черговості та матриці логічних зв'язків *RelationPriorityMatrix*.

Модель тестування [62-67] – формально-логічна конструкція, що дозволяє АНС виконувати коректну оцінку знань, вмінь та навиків навчальної діяльності об'єкта у визначеній предметній області. Автоматизовані навчальні системи та їх інтелектуальні версії використовують ряд відомих моделей і їх модифікацій. Перелік моделей за рівнем складності може бути наступним: моделі на основі засобів *FuzzyLogic* (зокрема, моделі з використанням лінгвістичних значень) імовірнісні моделі та моделі на основі коефіцієнтів упевненості; уніфіковані адаптивні моделі; моделі на основі сценаріїв; моделі з диференційованою складністю навчальних завдань і проблем; моделі на основі представлення та

задоволення обмежень; моделі на основі контролю рівнів засвоєння знань, умінь і навичок; моделі на основі часових диференціацій і преференцій. Існують також гібридні та комбіновані моделі, створені на основі перелічених методів, засобів і технологій.

Основною особливістю моделей тестування на основі нечітких множин *FuzzyTestingModel* є можливість використання відповідей об'єкта навчання на навчальні завдання та представлення рішень навчальних проблем у формі лінгвістичних нечітких значень. В даній формі можуть бути представлені результати навчальної діяльності об'єкта: наприклад у формі вводу значень – більше, менше, найбільше, найменше, максимальне, мінімальне, середнє, ймовірно, малоімовірно, швидше за все, в більшості випадків, інколи, частково, переважно тощо. Також – параметри й оцінки, що вводяться самою системою, (особливо якщо АНС характеризується як інтелектуальна) для оцінювання часу, витраченого об'єктом навчання на формування відповіді або побудови коректного рішення; оцінки коректності відповіді або рішення, запропонованого об'єктом навчання; диференціації рівня складності завдання або навчальної проблеми; оцінки поточного рівня знань; оцінки підсумкового рівня знань; оцінки рівня й ефективності навчальної діяльності. Для оцінки параметрів система може використовувати такі лінгвістичні значення, зазначених як: малий, великий, середній, дуже великий, дуже малий, складний, простий, частковий, загальний, підвищеної складності, спрощений, правильний, неправильний, частково правильний, частково неправильний.

Використання нечітких характеристик *FuzzyParams* і нечітких оцінювань *FuzzyEvaluations* в АНС має як переваги, так і недоліки.

Основні переваги:

1. Можливість простої та зрозумілої класифікації складності навчальних завдань і проблем: прості, складні, підвищеної складності тощо.
2. Інтуїтивно зрозуміла для об'єкта навчання шкала оцінювання його знань, умінь і навичок за класичними оцінками: відмінно, добре, задовільно тощо.

Недоліки:

1. Складність переведення нечіткої оцінки (в класичній шкалі) в рейтингову оцінку (в 100-бальній шкалі) при поточному оцінюванні знань, умінь та навичок, а також для представлення результатів тестування.

2. Складне переведення диференційної оцінки рівня складності *ComplexityLevel* тестових завдань і навчальних проблем із нечіткої форми в чисельну шкалу.

Основним компонентом моделі тестування на основі стратегії *StrategyTestingModel* є вибір та формування стратегії тестування *TestingStrategy*, – вибірка навчальних завдань і навчальних проблем. Критерієм формування вибірки є перелік навчальних тем, навчальних курсів, змістовних модулів, рівнів складності, форм представлення навчальних завдань і проблем, встановлених часових проміжків, перелік накладених обмежень і їх систем.

На основі сформованих та активованих сценаріїв кожен об'єкт навчання одержує власну вибірку навчальних завдань та проблем з бази даних, бази знань або сховища даних системи. Це забезпечує властивість паралельності (однакової кількості та сумарної складності тестових наборів). Проте, на відміну від інших, властивість паралельності у даній моделі є детермінованою (наперед визначеною) і закладається в уже сформовані стратегії.

Недоліком даної моделі можна назвати її низьку адаптивність, оскільки неможливо передбачити наперед і закласти у відповідну стратегію поточні та підсумкові рівні знань, вмінь і навичок об'єктів навчання.

Для обраного навчального курсу, предметної області знань, сукупності областей знань, розділів навчального курсу, навчальних тем, змістовних модулів виділяється множина навчальних завдань і множина навчальних проблем, із визначеними рівнями складності для кожного входження. Суть моделі тестування зі зростаючою складністю *IncrementalComplexityModel* полягає у виконанні навчальної процедури, що складається з наступних кроків:

1. **крок 1** – визначення кількості рівнів складності;
2. **крок 2** – необхідність включення в тест завдань, що представляють всі введені рівні складності;

3. **крок 3** – здійснення випадкового вибору підмножини завдань так, щоби кількість вибраних завдань була більшою або рівною кількості введених рівнів складності;
4. **крок 4** – сортування згенерованої множини завдань за рівнями складності в порядку інкременту (зростання складності) ;
5. **крок 5** – перевірка розподілу завдань за рівнями складності та проведення нормалізації даного параметру.

Використання описаної моделі має такі переваги:

1. забезпечення надійності тестування шляхом контролю їх паралельності за градієнтом складності;
2. диференціація сумарного результату тестування на основі включення параметру їх складності.

Для обраного навчального курсу, предметної області знань, сукупності областей знань, розділів навчального курсу, навчальних тем, змістовних модулів виділяється множина навчальних завдань і множина навчальних проблем. Суть моделі тестування з часовими обмеженнями *TimedTestModel* полягає в імплементації кроків:

1. **крок 1** – випадкова вибірка підмножини завдань і навчальних проблем;
2. **крок 2** – вказання часової межі (максимального часу) для введення відповідей на навчальні завдання та коректного рішення навчальних проблем;
3. **крок 3** – генерування (після завершення виділеного часу) системою сумарної оцінки підрахунком вагових коефіцієнтів правильних відповідей і коректних рішень, виконаних об'єктом навчання за встановлений час.

Дана модель є особливо ефективною при врахуванні складності завдань і навчальних проблем та їх диференційованого сортування у міру зростання або спадання складності.

Встановлення часових інтервалів для процедур тестування є важливим фактором поточного та підсумкового контролю рівня знань, умінь та навичок з

точки зору глибини та комплексності розуміння та засвоєння контенту навчального матеріалу. Можна виділити такі аспекти даної проблеми:

1. встановлення мінімально можливого часового проміжку сприятиме розвитку *автоматизму* в засвоєнні контенту навчального матеріалу. Такий підхід є особливо ефективним при фахових тренінгах, наприклад, для сертифікаційних екзаменів із програмних технологій;

2. встановлення часових проміжків середньої та високої розмірності дозволяє розвивати елементи творчого підходу до рішення навчальних проблем і завдань (і, відповідно, не сприятиме розвитку автоматизму);

3. встановлення надмірних часових проміжків, як показує досвід, є мало ефективним, оскільки в результаті надмірні старання об'єкту тренінгу дають зворотній ефект.

Важливою характеристикою інтелектуальних АНС є можливість контролювати рівень засвоєння контенту навчального матеріалу та його ранжування (розподілу за рівнем засвоєння). Для цього множина всіх тестових завдань і тестових технологічних проблем класифікується за рівнями засвоєння контенту тренінгового матеріалу. Тому послідовність виведення (активації) технологічних завдань й проблем розбивається на завершені етапи. При цьому перехід від однієї порції до іншої виконується системою, якщо об'єкт тренінгу показує рівень оволодіння контентом, більший деякого граничного значення *ThresholdValue*, визначеного для кожного рівня.

При вирішенні технологічних завдань й проблем важливим фактором для АНС є час відповіді *AnswerTime*, затрачений об'єктом тренінгу на пошук правильної відповіді або коректного рішення. Тому дискретні значення часу відповідей і часу рішень є однією зі складових поточної оцінки знань, умінь та навичок; сумарні часові значення є складовими підсумкових оцінок. Важливість часового параметру зумовлюється тим, що він дозволяє контролювати:

1. поточний і підсумковий рівень знань, умінь і навичок об'єкта тренінгу;
2. рівень самостійності об'єкту тренінгу при введенні правильних відповідей і пошуку коректних рішень;

3. рівень і швидкість розуміння об'єктом тренінгу виділених концептів предметної області, тем, змістовних модулів і курсів тренінгу у цілому.

Важливим способом підвищення надійності результатів тестування *TestingResultsReliability* поточного *CurrentKnowledgeLevel* і підсумкового *SummarizedKnowledgeLevel* рівня знань об'єкту тренінгу є фіксація часу відповіді *AnswerDuration* на навчальні завдання та часу рішення *SolutionTime* навчальних проблем. Також важливими є коефіцієнти диференціації *DifferentiationDegree* рівня складності *ComplexityLevel* технологічних завдань і проблем із перенесенням часових характеристик і коефіцієнтів рівнів складності у відповідні розділи профілів об'єкту тренінгу і моделей зон його найближчого розвитку.

Модель тестування *TestingModel* визначає порядок формування тестових завдань і технологічних проблем, спосіб формування множин обмежень, послідовність приєднання технологічних завдань та проблем в тренінговій стратегії, спосіб визначення поточного рівня знань об'єкта тренінгу, методи контролю зони найближчого розвитку.

В класичних АНС навчальна стратегія системи *SystemLearningStrategy* відносно до об'єкта тренінгу будується на основі множини його преференцій *TraineePreferencesSet* щодо запропонованої системою послідовності курсів тренінгу, тем і змістовних модулів. В інтелектуальних АНС навчальні стратегії генеруються на основі дослідження співвідношення між поточним профілем об'єкту тренінгу *TraineeCurrentProfile* і моделлю його зони найближчого розвитку *ProximalDevelopmentZone*.

Одним з елементів інтелектуальності та адаптивності АНС є зворотній зв'язок *Feedback*. Зворотній зв'язок дозволяє контролювати процес навчання *LearningProcess* і навчальну діяльність об'єкта тренінгу *TraineeLearningActivities* вибором відповідних навчальних стратегій *LearningStrategies*; контролювати засвоєння об'єктом тренінгу контенту технологічного матеріалу, курсів тренінгу, їх тем і завершених кредитних модулів *LearningContentMastering*; інтерпретувати кількісні та якісні показники результатів тестування *TestResults*; перевіряти

виконання технологічних завдань *TechnologicalTasksSolutions* і рішення технологічних проблем *TechnologicalProblemsSolutions*; оновлювати профілі об'єктів тренінгу *TraineeProfileUpgrading*.

Розрізняють два види зворотного зв'язку: *ExternalFeedback* - зовнішній зворотний зв'язок - і *InternalFeedback* - внутрішній зворотний зв'язок. Зовнішній зворотний зв'язок описує інформаційний обмін між експертом предметної області *ContentExpert* з одного боку і АНС – з іншого. Основними особливостями зовнішнього зворотного зв'язку *ExternalFeedbackFeatures* є:

1. Корекція профілів об'єктів тренінгу *TraineeProfileCorrection*
2. Корекція викладачем навчальної діяльності об'єкта тренінгу *TraineeActionCorrection*.
3. Корекція викладачем навчальних стратегій *LearningStrategiesCorrection*.

Внутрішній зворотний зв'язок описує інформаційний обмін між АНС (*LearningSystem*) й об'єктом тренінгу - *TraineeObject*. Основними особливостями внутрішнього зворотного зв'язку *InternalFeedbackFeatures* є:

1. Представлення відповідей системи *SystemFeedbacks* на дії об'єкту тренінгу *TraineeObjectActions* (*TraineeFeedback*) при вирішенні технологічних проблем *TechnologicalProblemsSolution* і навчальних завдань *TechnologicalTasksSolution*;
2. Представлення результативного внутрішнього зв'язку *ResultingInternalFeedback* (у формі демонстрації правильного результату *CorrectResultVisualisation*, демонстрації правильного рішення *CorrectSolutionVisualisation*, відповіді типу «вірно – невірно» *True – False*, відповіді у формі лінгвістичного значення *LinguisticValue*, відповіді у формі значення з коефіцієнтом впевненості *ConfidencedValues*);
3. Представлення консультативного внутрішнього зворотнього зв'язку *ConsultingInternalFeedback* (допомога *ConsultingHelp*, пояснення *ConsultingExplanation*, підказка *ConsultingHint*);

4. Представлення дій самокорекції об'єктом тренінгу *SelfCorrectionalActions* щодо корекції власної тренінгової діяльності *TraineeActivities* ;

5. Представлення результатів успішності *TraineeActivitySuccessfulness* , коефіцієнтів успішності *TraineeSuccessfulnessDegree* , лінгвістичного значення успішності *SuccessfulnessLinguisticValue* або помилковості *LearningActivitiesErrancy* , коефіцієнту помилковості *LearningErrancyDegree* , лінгвістичного значення помилковості *ErrancyLinguisticValue* тренінгової діяльності;

6. Представлення вихідних даних для виконання оцінки *LearningActivitiesEvaluations* і корекції результатів *LearningActivitiesCorrections* навчальної діяльності [68-74].

Окремим видом АНС є підвид інтелемедійних автоматизованих навчальних систем.

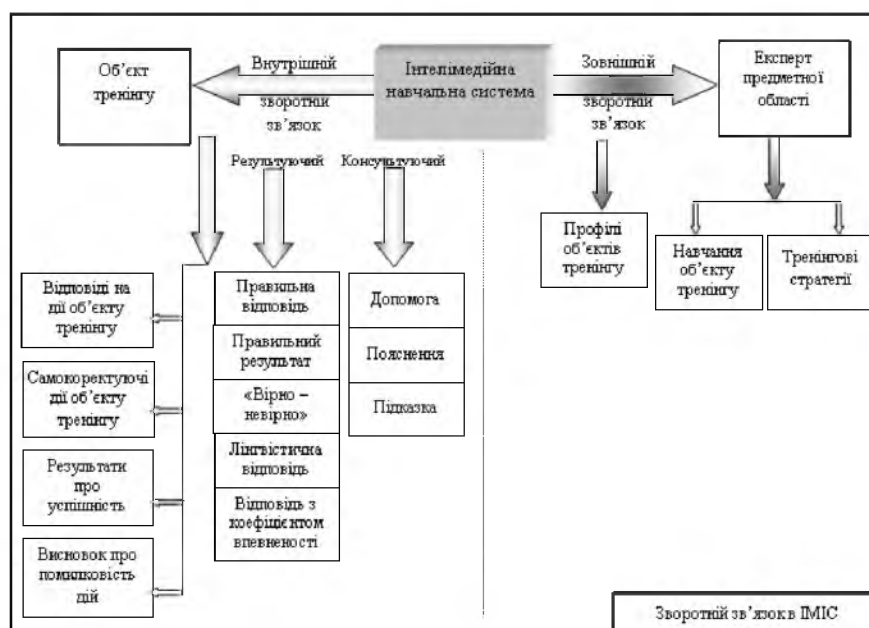


Рисунок 1.23 – Структурна схема зворотного зв'язку в ІМАС

АНС містить ряд традиційних засобів електронного навчання. Найбільш класичним засобом є електронний підручник *E – Book* , який складається з теоретичного матеріалу та прикладів. Разом із тими суттєвими відмінностями АНС

є засоби формування в об'єктів тренінгу знань *LearningObjectsKnowledges*, вмінь *LearningObjectsSkills* і навичок роботи в предметній області *SubjectDomainExpierence*, і програмні інструменти контролю й оцінки знань, вмінь і навичок. Дана задача вирішується імплементуванням у навчальні стратегії *LearningStrategies* послідовностей технологічних проблем *TechnologicalProblemSequences* і послідовностей технологічних завдань *TechnologicalTasksSequences*.

Автоматизована навчальна система містить такі основні складові:

1. блок контентного навчання;
2. тренінговий блок;
3. тестовий блок;
4. блок зворотного зв'язку.

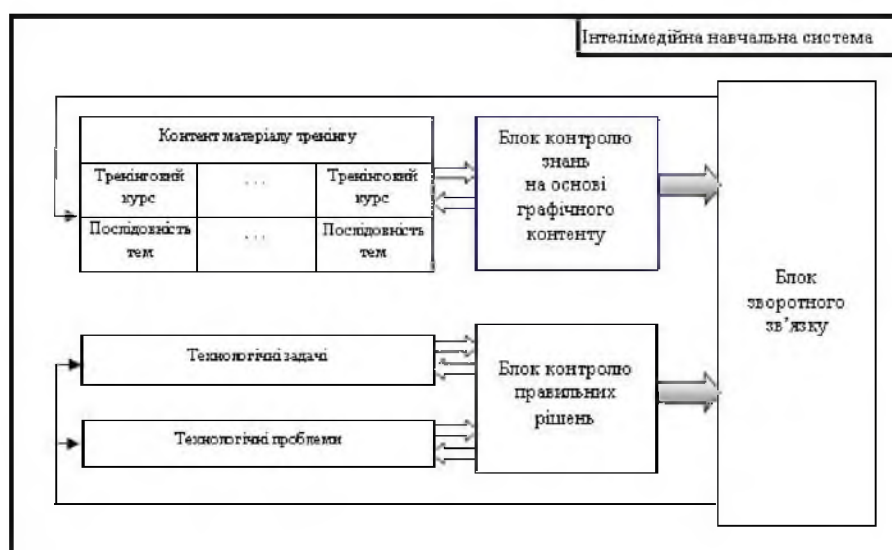


Рисунок 1.24 – Структурна схема процесу навчання засобами ІМАС

АНС на основі інтерфейсних елементів формує навчальне середовище *LearningEnvironment*, яке забезпечує виконання таких основних функцій: *ContentForm* - форма представлення контенту; *ContentTransferMode* - спосіб передачі контенту; *KnowledgePresentationForm* - форма представлення знань; *KnowledgeTransferMode* - спосіб передачі знань; *LearningResultControl* - контроль

результативності навчання; *LearningProcessSequence* - послідовність навчального процесу, *LearningProcessChronology* - часова послідовність навчального процесу. Таким чином, основне завдання навчального середовища, що формується комп'ютерними та програмними засобами АНС, полягає в забезпеченні та підтримці процесів формування знань *KnowledgeForming*, зберіганні знань *KnowledgePreserving* і русі знань у рамках навчального середовища *KnowledgeTurnover*.

АНС, побудовані на основі оверлейної моделі об'єкта тренінгу *TraineeOverlayModel*, розглядають множину знань об'єкта тренінгу *TraineeKnowledgeSet* відносно множини знань системи *SystemKnowledgeSet* як підмножини $TraineeKnowledgeSet \subset SystemKnowledgeSet$. Крім того, структура даних множин є співрозмірною. Кожному входженню навчального контенту *LearningContent* присвоюється певний числовий атрибут *NumericalAttribute*, який відображає степінь розуміння *ComprehensionLevel* об'єктом тренінгу технологічного матеріалу з даного розділу. Значення таких атрибутів верифікується в процесі опитування об'єктів тренінгу *TraineeQuestionings*.

Важливим фактором ефективної роботи АНС є рівень усвідомлення об'єктом тренінгу *TraineeRealizationLevel* й обґрунтування вибору способу тренінгової чи імплементаційної дії *TraineeActionMode*. Можна виділити декілька рівнів даного процесу: **рівень 1** – об'єкт тренінгу може пояснити виконані ним дії *TraineeActions*, відправлені системі відповіді *TraineeAnswers*, виконані вибори *TraineeChoices*, запропоновані рішення технологічних проблем *TechnologicalProblemSolutions* на основі контенту тренінгового матеріалу *TraineeContent* з тренінгового курсу *TraineeCourse* (тобто пояснення об'єкта тренінгу можна розглядати як зумовлені контентом *SubjectSubstantiated*); **рівень 2** – об'єкт обґрунтовує свої дії, відправлені системі відповіді, виконані вибори, запропоновані рішення технологічних проблем на основі контенту тренінгового матеріалу суміжних тренінгових курсів *ProximateTraineeCourses* (тобто, цей процес зумовлюється контентом суміжних

тренінгових курсів); **рівень 3** – пояснення формується на основі контенту різних тренінгових курсів *JoinedCoursesSubstantiated*.

Однією з моделей об'єкта тренінгу *TraineeObjectModel*, що використовується в АНС, є пертурбаційна модель *PerturbationModel*. В її основі дивергенція (розбіжності) знань навчальної системи *LearningSystemKnowledges* і множини знань об'єкта тренінгу *TraineeObjectKnowledges*.

Для досягнення ефективності АНС побудованої на пертурбаційній моделі, необхідно визначити відмінності множини знань об'єкта тренінгу і множини знань системи, що дозволить уникнути невизначеності в моделі об'єкта тренінгу *ModelUncertainty*. Серед причин неспівпадінь даних моделей можуть бути: недостатня кількість знань *KnowledgeLack*; включення хибних знань *IncorrectKnowledge*; некоректне використання знань *IncorrectKnowledgeApplication* або відсутність навичок їх застосування *KnowledgeApplicationSkills*; допущення об'єктами тренінгу неумисних помилок *UnintentionalError*; внесення об'єктами тренінгу довільних рішень навчальних проблем *RandomSolutions* і випадкових відповідей на технологічні завдання *RandomAnswers*, що спричиняє виникнення навмисних помилок *PurposeError*. Ці види помилок можна усунути шляхом повторного уточнюючого тестування об'єктів тренінгу *RepeatTestig* або використанням контрольних послідовностей простих питань *SimpleQuestionsSequences*.

Ефективною моделлю об'єкта тренінгу *TraineeObjectModel* в АНС є різницева модель *DifferentialModel*. Основою різницею моделі об'єкта тренінгу є відмінності між множиною знань об'єкта тренінгу та множиною знань, закладених в тренінговому контенті системи *ContentBasedKnowledges*. Основні особливості даної моделі *ModelFeatures* – це можливість визначення множини знань, відсутньої в об'єкта тренінгу у поточному періоді його тренінгової діяльності *AbsentKnowledges*, ідентифікування ситуації неправильного використання контентнобазованих знань *IncorrectKnowledgeApplication*. Основний механізм функціонування АНС на основі

різничевої моделі – це аналіз відповідей і рішень об'єкта навчання *AnswersSubmitted*, *SolutionsSubmitted* та їх порівняння з еталонними ідеальними знаннями, закладеними в системі *SystemKnowledges*, які відповідають знанням експерта предметної області *SubjectDomainExpert* і є достатніми для вирішення технологічних завдань і проблем даного класу *SampleTaskSolution* і *SampleProblemSolution*.

1.3 Представлення контенту технологічного матеріалу

Новітні ІМАС оперують зі знаннями та даними. Основне завдання полягає в передачі знань експерта закладених в контенті бази знань ІМАС об'єкту тренінгу. При цьому дані, які не є знаннями, слід перетворити в знання застосуванням методу видобування даних і знань (Data Mining and Knowledge Mining). Спосіб передачі знань визначається на основі підходу представлення та задоволення обмежень. Тобто фрагмент знань, який представляє правильне рішення проблеми *CorrectSolution*, описується відповідною множиною задоволених обмежень, ранжованими відповідно значень вагових коефіцієнтів. Тому основною характеристикою коректних знань *CorrectKnowledges* в ІМАС є опис на основі обмежень, що якісно відрізняє ІМАС від інших автоматизованих навчальних систем.

В АНС під надмірністю електронного підручника *E – BookExcessiveness* розуміють наявність контенту, інформативна цінність якого не співставляється з множиною зони найближчого розвитку об'єктів тренінгу *ProximalDevelopmentZones* і, таким чином, не дозволяє одержувати знання, описані множиною цілей тренінгу *TraineeGoals*.

Наповнення технологічного контенту в АНС можна представляти як компактні включення – кейс-набори, які поєднують тестовий, графічний, аудіо-, відео- та мультимедійний контент тренінгового матеріалу, що може бути самостійно опрацьований об'єктом тренінгу.

Важливою особливістю АНС, яка сприяє їх швидкому поширенню, є широкі можливості щодо представлення тренінгового контенту, його підтримки та оновлення. Ефективність представлення контенту оцінюється градієнтом дані – знання. Класичним прикладом тренінгового контенту в формі даних *DataTraineeContent* є вміщений в АНС з вивчення мов програмування довідковий матеріал у формі локального *help*-файлу (*C++ Builder Help* або посилання на відповідний глобальний інтернет-ресурс, наприклад *MSDN – Microsoft Developer Network*). Проте наявність такого контенту має швидше кількісну, ніж якісну характеристику. Тобто, ознайомлення об'єкта тренінгу з даним контентом, не гарантує, що він відразу почне писати успішні програми на відповідній мові програмування. Тому основною перевагою інтелектуальних та інтелектуалізованих АНС є тренінговий контент у формі знань *KnowledgTraineeContent*. Другою важливою характеристикою контенту є вибір способу його групування *ContentOrderingMode*. Основний спосіб групування контенту – представлення експертом предметної області *ContentExpert* у формі індексованого ресурсу зі змістом *IndexedTraineeContent*. Індексований контент дозволяє виконувати пошук за ключовими словами *KeywordsSearch*, за списком ключових слів упорядкованих в алфавітному порядку *OrderedKeywordsSearch*, а також за змістом *ContentIndex* у порядку викладення навчальних тем, підтем, модулів, курсів. Під рухом знань *KnowledgeTurnover* розуміють передачу контенту *ContentTransfer* від носія знань *KnowledgesSources* до одержувача знань *KnowledgesRecipient*.

При роботі з АНС об'єкт тренінгу може отримати знання кількома способами:

1. під час вирішення технологічних задач *TechnologicalTaskSolution* і технологічних проблем *TechnologicalProblemsSolution*, що є частиною самостійної тренінгової діяльності об'єкту тренінгу *TraineeSelfActivity*. Цей тип знань називається неартикулярними знаннями;

2. на основі аналізу прикладів рішень типових технологічних завдань і проблем, тобто шляхом навчання на прикладах *LearningbyExamples*;

3. шляхом прямої передачі від експертного модуля *DirectTranferedKnowledges*.

Ефективним методом АНС є рішення технологічних завдань і проблем на основі раніше виконаних прикладів *PreviouslyDoneExamples*. В основу даної технології навчання *LearningTechnology* покладено принцип аналізу технологічних завдань подібного типу *SampleTasks* і навчальних проблем подібного типу *SampleLearningProblems*, успішно розв'язані об'єктом навчання в попередніх сесіях роботи з системою *LearningSessions* замість виведення тестового пояснення помилок, *ErrorsExplanation* допущених об'єктом тренінгу *TraineeObject*.

Електронні посібники *Learning-E-Manuals* в АНС мають практичне спрямування стосовно застосування навичок, отриманих у результаті оволодіння контентом тренінгового матеріалу *ContentBasedSkills*, і вмінь із застосування знань, отриманих у результаті вивчення певного тренінгового курсу *TraineeCourseKnowledges*.

З точки зору ефективності основним завданням АНС щодо рівня засвоєння тренінгового контенту *ContentMasteringSkills* є побудова максимально придатної *MostUseFull* для об'єкта тренінгу послідовності контенту тренінгового матеріалу *TraineeContentSequence*. Дана задача вирішується за допомогою наступних категорій: вибір курсу тренінгу *TraineeCourseSelection*; вибір розділу курсу тренінгу *CourseChapterSelection*; вибір завершеного модуля тренінгу згідно ECTS (European Community Course Credit Transfer System) *TraineeModuleSelection*; вибір наступної теми *NextTopicSelection*; вибір наступного завдання *NextTaskSelection*; вибір наступної технологічної проблеми *NextProblemSelection*.

Практикум *E – PracticalCuidebook* є інструментом АНС, з практично орієнтованими технологічними завданнями та технологічними проблемами, що

підвищують рівень засвоєння контенту тренінгового матеріалу *ContentMasteringSkills*.

На рівні адаптації до поточного рівня знань об'єкта тренінгу *CurrentKnowledgeLevel* ефективними є генерація та використання ситуатійних технологічних завдань *SituationalTasks* і ситуатійних технологічних проблем *SituationalTechnologicalProblem*, для перевірки знання *TraineeObjectKnowledges* і вміння *TraineeObjectSkills* з метою практичного застосування *ContentApplication*.

Для виконання задачі класифікації контенту тренінгового матеріалу *TraineeContent*, формування послідовності технологічних задач *TechologicalTasksSequence* і технологічних проблем *TechnologicalProblemsSequence* доцільним є введення тематичних завдань *SubjectTask* і тематичних технологічних проблем *SubjectTraineeProblem*. Їх основним представленням є ланцюгове представлення задач *ChainedTasks* і проблем *ChainedTraineeProblems* комплексної перевірки знань *KnowledgeTesting* для виокремленої тематичної області контенту тренінгового матеріалу *SelectedTopic*.

Представлення тренінгового матеріалу *TraineeContentPresentation* в ІМАС володіє певним ступенем абстрагованості *AbstractionLevel*. Використовується такі категорії абстрагованих представлень: *DescriptiveLevelCategory* – категорія описових представлень засобами природної мови користувача, що дозволяє описати факти, явища та процеси предметної області й здійснити їх класифікацію; *AnaliticalLevelCategory* – категорія представлень логічними засобами теоретичних основ формування та функціонування фактів, явищ і процесів предметної області та їх аналізу на якісному рівні; *FormalLevelCategory* – категорія представлень формально-логічними засобами фактів і явищ предметної області, створення їх формально-логічних моделей і виконання аналізу на кількісному рівні; *AxiomaticLevelCategory* – категорія аксіоматичних описів й обґрунтування фактів, явищ і процесів предметної області.

В ІМАС використовуються адаптивні *AdaptiveTraineeCourses*, дистанційні *DistanceTraineeCourses*, гіпермедійні *HypermedialTraineeCourses* навчальні курси, які є об'єднанням навчальних систем *LearningSystem* за спорідненими предметами *ConnectedSubjects*. Практична реалізація ІМАС може проектуватися для визначеної предметної області *SelectedSubjectDomain* або для кількох споріднених предметних областей *ConnectedSubjectDomains*. Відповідно, в другому випадку система включатиме в себе деяку підмножину навчальних систем зі споріднених предметів тренінгу *ConnectedTraineeSubjects*. Об'єкт тренінгу одержуватиме технологічні проблеми з одного або кількох предметів тренінгу, які на практиці представлені у вигляді курсів тренінгу *TraineeCourses*. Об'єкт тренінгу може самостійно вибирати предмети тренінгу зі списку *ListedSubjects* і технологічні проблеми, якщо задовольняються обмеження високого рівня, що описують його область найближчого розвитку.

Новітні застосування ІМАС дозволяють поєднувати тестову, графічну та мультимедійну інформацію для підвищення якості процесу тренінгу. З точки зору укладачів баз даних, баз знань і контенту тренінгового матеріалу в ІМАС це відкриває широкі можливості з використання класичних технологій, таких як: електронні довідники *E-References*, електронні словники *E-Dictionaries*, електронні енциклопедії *E-Wikis*, електронні путівники *E-Guidance*, електронні підручники *E-Book*. Перевага мультимедійних засобів ІМАС в легкому та зрозумілому наочному представленні контенту тренінгового матеріалу, з описом складних способів вирішення технологічних проблем або контролю ходу технологічних процесів. Проте, в даному випадку складно організувати зворотній зв'язок (наприклад, при використанні аудіо- чи відео лекцій), оскільки процес є одностороннім.

Як і в інших автоматизованих навчальних системах, тренінгова діяльність в ІМАС організовується у вигляді сценаріїв (стратегій) *TraineeStrategies*. В кожній стратегії використовуються попередньо підготовлені знання *PreparedKnowledge* у вигляді структурованого контенту *StructuredContent*, напівструктурованого

контенту *Semistrukturovanogo* і слабоструктурованого контенту *Weaklystrukturovanogo*. В ІМАС реалізується класична стратегія тренінгу на прикладах *TraineebyExamples*. Проте звичайні автоматизовані навчальні системи реалізують дану стратегію демонстрацією технологічних проблем та їх готових рішень, то в ІМАС реалізається супроводжується візуалізацією ієрархії релевантних обмежень.

В автоматизованих навчальних системах система гіперпосилань організована так, щоб об'єкт тренінгу, переходячи за гіперпосиланнями з однієї технологічної проблеми до іншої, завжди міг зорієнтуватися в загальній структурі гіперресурсів і вибирати коректну послідовність роботи з конвентом тренінгового матеріалу. Це реалізується шляхом розміщення посилань на попередні рівні та головну сторінку разом зі змістом контенту тренінгового матеріалу. Хоча основна ідея, закладена в ІМАС, полягає в роботі з обмеженнями, проте, як показує досвід, в експлуатації таких систем найбільш доцільним є використання обмежень як лише внутрішнього системного механізму, оскільки розуміння природи обмежень та вміння оперувати ними для опису технологічних проблем є надто складним для об'єктів тренінгу.

1.4 Методи тестування в автоматизованих навчальних системах і тренажерах

Оскільки процедури тестування є важливим елементом функціональності АНС, то верифікація їх валідності, відповідно, виділяється в окрему задачу. Вона залежить від таких параметрів, як: структуризація об'єктів тренінгу за групами або відповідно до рівня знань *InitialKnowledgeLevel*; спосіб та метод інтерпретації результатів тестування; метод відбору та генерації тестових завдань, технологічних завдань, технологічних проблем та їх послідовностей; оптимізація кількості тестових завдань, вибір оптимальної кількості відповідно до курсу тренінгу, теми, змістовного модуля; характеристика повноти контенту *ContentCompleteness* відносно тренінгової теми, курсу, змістовного модуля; розподіл тестових завдань за градієнтом складності (інкрементальний, декрементальний); якісні характеристики

тестових завдань щодо верифікації рівня знань об'єкта тренінгу. Важливою характеристикою тестових завдань є їх варіативність *TaskVariety*, тобто можливість заміни складових компонентів на іншими релевантними, таким чином, щоби відповіді об'єктів тренінгу відрізнялися, а вони вирішували фактично відмінні завдання в рамках однієї теми тренінгу виділеного курсу. Тест вважається ефективним, якщо його завдання є максимально варіативними.

Призначення тестів в АНС – контроль рівня знань, умінь і навичок. Проте робоча область тестування *TestingWorkSpace* може ранжуватися: технологічна проблема, тема тренінгу, сукупність тренінгових тем, сукупність тренінгових підтем, сукупність суміжних курсів тренінгу, сукупність несуміжних курсів тренінгу. Така ранжованість робочих областей тестування *WorkSpaceRanging* відповідає характеристиці тестів, яка називається гетерогенністю тестів *TestHeterogeneity*. Зворотня до гетерогенності тесту – гомогенність – використовується для контролю рівня знань, умінь і навичок за однією виділеною тренінговою дисципліною, за виділеним курсом тренінгу, розділом курсу тренінгу, змістовним модулем.

Тести в АНС складаються з технологічних завдань *TechnologicalTasks* і технологічних проблем *TechnologicalProblems*. Тестові входження *TestEntries* мають такі основні характеристики: диференціація *TestDifferentiation*, дисперсія *TestDispersion*, кореляція *TestCorrelation*. Об'єкти тренінгу вирішують технологічні завдання і технологічні проблеми одним зі способів: вибір одного із запропонованих варіантів відповіді *AnswerChoice* – найбільш правильна, найбільш неправильна, найбільш імовірна, найменш імовірна, всі правильні, всі неправильні, одна правильна відповідь; внесення власного варіанту відповіді до запропонованого технологічного завдання або власного способу рішення технологічного проблеми *TechologicalProblemSolution*; визначення правильної послідовності рішення *CorrectSolutionSequence*, що складається з кількох кроків (необхідно впорядкувати та ранжувати елементи відповіді або рішення в порядку їх релевантності); виконання лінгвістичного оцінювання *LinguisticEvaluation*; введення відповіді на

технологічного завдання та рішення технологічної проблеми з певним коефіцієнтом впевненості *CertaintyFactor*; визначення відповідності *CorrespondenceDefining* – об’єкт тренінгу отримує списки параметрів та списки значень і співвідносить кожен із параметрів до відповідних їм значень; визначення асоціативності, аналогічності, виключення надлишковості.

Якість тесту *TestQuality* – характеристика, яка визначається як сумарне значення надійності *TestingReliability* і валідності *TestingValidity* тесту, причому значення надійності та валідності співвідносяться з результатами тестування *TestingResults*. Тому якість тесту визначається саме якістю результатів тестування об’єкту тренінгу *TraineeTestingResults*.

В класичній моделі тестування *ClassicModelTesting* для виділеної області знань *SelectedKnowledgeDomain*, предметної області *SubjectDomain*, виділеного курсу тренінгу *SelectedTraineeCourse*, виділеної теми *SelectadTraineeTopic* або виділеної підтеми *SelectadTraineeSubtopic* визначається m технологічних завдань *TechnologicalTask* і технологічних проблем *TechologicalProblem*, на основі яких здійснюється тестовий контроль знань, умінь і навичок. Із множини завдань *TaskSet* і множини проблем *ProblemSet* випадковим чином вибирають $l, l \leq m$ технологічних завдань і проблем, представлених у формі тесту з вибором правильної відповіді зі списку або введення правильної відповіді з клавіатури. Результати відповідей на технологічні завдання та запропоновані (вибрані) рішення технологічних проблем оцінюються значеннями “правильно” (*Correct*), “неправильно” (*Incorrect*). Загальна оцінка за тест обчислюється в процентному відношенні правильних відповідей та рішень до загальної кількості. Незважаючи на простоту реалізації, дана модель базується на випадкових виборах і не дозволяє контролювати процес вибірки завдань за їх складності. Крім того результуюча оцінка має кількісно-статистичний характер і, відповідно, не містить якісної складової, що представляє рівень складності *ComplexityLevel* кожного окремого технологічного завдання та

технологічної проблеми. Тому, виходячи з вищеописаного, запропонована модель має низький рівень об'єктивності *TestingObjectivity* і надійності *TestingReliability*.

В класичній моделі тестування з урахуванням складності *ComplexityLevelModel* для виділеної області знань *SelectedKnowledgeDomain*, предметної області *SubjectDomain*, виділеного курсу тренінгу *SelectedTraineeCourse*, виділеної теми *SelectadTraineeTopic* або виділеної підтеми *SelectadTraineeSubtopic* визначається m технологічних завдань *TechnologicalTask* і технологічних проблем *TechnologicalProblem* кожна з яких має певний рівень складності *ComplexityLevel*, $CL_i, i = 1, m$. На їх основі виконується тестовий контроль знань, умінь та навичок. Із множини завдань *TaskSet* і множини проблем *ProblemSet* випадковим чином вибирають $l, l \leq m$ технологічних завдань і проблем, представлених у формі тесту з вибором правильної відповіді зі списку або введенням правильної відповіді з клавіатури. При обчисленні результуючої оцінки враховуються не тільки правильні відповіді та правильні рішення, але й релевантні рівні складності завдань і проблем. Таким чином, врахування складності завдань підвищує об'єктивність тесту *TestingObjectivity*. Проте надійність тестування *TestingReliability* залишається низькою через випадковість вибору технологічних завдань *RandomTasksSelection* і вибору технологічних проблем *RandomProblemsSelection*, а також неможливість контролювати сумарну складність тестових завдань і проблем, які отримує об'єкт тренінгу в процесі тестування.

Важливим для функціональності й імплементаційної стійкості АНС є питання вводу даних об'єктом тренінгу при рішенні технологічних завдань і технологічних проблем. Основними типами вводу є:

1. введення числового значення *NumberValue* в одному з дозволених числових типів;
2. введення логічних значень *LogicalValues* як інтерпретації відповіді на запитання системи типу «Так-Ні»;
3. введення категоризованих значень *CategoricalValues* як результату вибору зі списку варіантів, запропонованих системою;

4. вільне введення значень у співрозмірному текстовому полі або текстовому об'єкті.

Рекомендовано використовувати види вводу 1 – 3. Значення 4-го типу вимагають додаткових затрат на написання процедур семантичної та синтаксичної обробки.

При роботі з користувачем АНС надає йому можливості для виконання різних видів оцінювання *UserEvaluations*. Водночас, АНС очікує від користувача виконання цих видів, а також його *UserEstimations* щодо згенерованих системою питань *SystemRaisedQuestion*. Ці дані користувач може вводити в якісній або кількісній формі:

1. оцінювання й міркування, виражені в числовій формі *NumberedEvaluations* і *NumberedEstimations*;

2. оцінювання та міркування, виражені в логічній формі *LogicalEvaluations* і *LogicalEstimations*;

3. оцінювання та міркування, виражені в категорійній формі *CategoricalEvaluations* і *CategoricalEstimations*;

4. оцінювання та міркування, виражені в числовій формі з певним коефіцієнтом впевненості *CertaintyMarkedEvaluations* і *CertaintyMarkedEstimations*;

5. оцінювання та міркування, виражені в термінах лінгвістичних значень (методи нечіткої логіки) *LinguisticEvaluations* і *LinguisticEstimation*.

Показник ефективності тесту, що вказує на розбіжності результатів тестування *TestResultsDivergency*.

Контент технологічного матеріалу АНС може містити технологічні завдання та технологічні проблеми з різною контентною інтерпретацією, але з однаковими статистичними показниками *StatisticalDescription*. Відповідно, тести, що складаються з елементів паралельних технологічних завдань *ParallelTechnologicalTasks* і паралельних технологічних проблем *ParallelTechnologicalProblems*, називаються паралельними тестами *ParallelTests*.

Новітні АНС використовують технологію псевдотестових технологічних завдань *SemiTestTasks* і псевдотестових технологічних проблем *SemiTestProblems*. Використання цієї технології дозволяє підвищити контекстну залежність технологічного контенту, оскільки в даному випадку правильна відповідь *CorrectAnswer* на одне з ренінгових, і, відповідно, правильне рішення *CorrectSolution* виділеної технологічної проблеми залежить від правильності рішення деякої іншої проблеми або множини технологічних проблем *TechologicalProblemsSet*. Статистичні характеристики залежних завдань *DependedTasks* і залежних технологічних проблем *DependedTechnologicalProblems* будуть, відповідно, взаємозалежними і взаємообумовленими. Найчастіше використовуються псевдотестові технологічні завдання *SemiTestTasks* і псевдотестові технологічні проблеми *SemiTestProblems* таких видів: ситуаційні, тематичні, ланцюгові, однорангової обмеженості та багаторангової обмеженості.

АНС використовують системи тестових завдань *TestTasksCollection* для контролю та визначення рівня знань *KnowledgeLevelVerification*. Тестові завдання *TestTasks* можуть бути розміщені кількома способами: в довільному порядку *RandomTest*, визначеному порядку *DefinedTestOrder* або на основі моделі тестування *TestingModel*. Ефективним способом упорядкуванням тестових завдань є розміщення їх в порядку зростаючої складності *IncrementalComplexity*.

Структура та призначення тестових завдань і технологічних проблем визначається статистичними характеристиками *StatisticalDescription*, такими як: диференціація, дисперсія, кореляція, які є взаємнонезалежними.

В автоматизованих системах навчання один тест вважається більш ефективним, ніж інший, якщо він дозволяє визначити необхідні характеристики об'єкта тренінгу *TraineeObjectFeatures* із використанням меншої кількості завдань (*TaskNumber*), що підвищує якість та швидкість тесту. Таким чином, ефективність тесту *TestEfficiency* визначається на основі двох параметрів: кількість завдань у тесті *TasksNumber* і рівень знань об'єктів тренінгу *KnowledgeLevel*.

В ІМАС рівень знань об'єктів тренінгу визначається областю найближчого розвитку. Кількість завдань у тесті співвідноситься з загальною кількістю обмежень, що описують даний тест *TotalConstraintsNumber*. Це сумарне значення обмежень, присвоєних кожному тестовому завданню, що представляється у вигляді послідовності технологічних проблем *TechologicalProblems*. Множина обмежень щодо певної технологічної проблеми називається множиною релевантних обмежень *RelevantConstraints*. У процесі виконання тесту об'єкт тренінгу намагається вирішити послідовність технологічних проблем, визначених стратегією тренінгу *TraineeStrategy*. У результаті, в кожній множині релевантних обмежень система виділить підмножини задоволених обмежень *SatisfiedConstraints* і порушених обмежень *ViolatedConstraints*.

Таким чином, для оптимізації роботи всієї системи доцільно оптимізувати множини тестів, вибираючи з них найбільш ефективні. Тобто завдання зводиться до оптимального підбору тестових завдань у вигляді послідовності технологічних проблем, які визначають рівень складності тесту *TestComplexityLevel*. Для ІМАС такий процес називається процесом підбору або вибірки на існуючій множині послідовності технологічних проблем, які релевантно визначають рівень складності тесту. Очевидно, що на етапі формування бази знань ІМАС проблема підбору релевантних технологічних проблем зводиться до проблеми підбору і накладання систем релевантних обмежень.

Згідно класичного підходу, що використовується в АНС, найефективнішим вважається той тест, який за складністю завдань відповідає рівню знань об'єктів тренінгу. В ІМАС це визначення формулюється так: складність тесту не повинна виходити за область найближчого розвитку об'єктів тренінгу. При груповому навчанні інтелектуальність й адаптивність ІМАС проявлятиметься в індивідуальному підборі рівня складності тестів відповідно до профілів об'єктів тренінгу. Форма представлення технологічних проблем і тестових завдань теж впливатиме на ефективність тесту. Тому, коли йдеться про загальну ефективність тесту або визначення найбільш ефективного тесту з даної множини, слід визначати ефективність форми представлення тестового контенту *TestContentForm* через

ефективність представлення контенту *ContentRepresentationEfficiency*, що безпосередньо визначає рівень розуміння контенту *ContentComprehensionLevel*.

Ефективність тесту визначатиметься також і способом підбору технологічних проблем і тестових завдань із урахуванням діапазону зміни складності *ComplexityRange*. Залежність полягає в співвідношенні рівня знань об'єктів тренінгу *KnowledgeLevel*, середнього рівня знань *AverageKnowledgeLevel*, складності тесту *TestComplexity*, середнього рівня складності тестів *AverageTestComplexity*. Тому завдання необхідності покриття тестовими завданнями всього діапазону зміни рівнів складності знижуватиме ефективність тесту в кожному окремому проміжку загального діапазону. В протилежному випадку, якщо потрібно ефективно визначати рівень знань об'єктів тренінгу із заданим показником, то необхідна більша кількість технологічних проблем і тестових завдань заданого рівня складності. Тому при використанні АНС в загальному говорять про диференційовану ефективність тестів *DifferentialTestEfficiency*. Це означає, що кожен конкретний тест не може бути ефективним на всьому діапазоні рівнів знань *KnowledgeLevelRange*: тобто він буде більш ефективний в одних проміжках і менш ефективним – в інших. В ІМАС диференціація ефективності тестів визначається відповідними обмеженнями, їх ваговими коефіцієнтами та ранжуванням. Тому структура диференціації розглядається як така, що зумовлена обмеженнями *DifferentialEfficiencybyConstraints*.

1.5 Обґрунтування напряму досліджень

Координація ефективної мультимедійної інтеграції даних у концепцію експертних систем містить ряд аспектів. Пам'ять, дисковий простір, периферійне обладнання – все це разом накладає обмеження на реалізацію проекту. Однією з цілей дослідження є досягнення оптимального співвідношення між інтелектуальністю та мультимедійністю системи (як з точки зору бази знань, так і з точки зору інтерфейсу). Важливо розуміти, що в багатьох реалізація експертних

систем, існують обмеження, які необхідним чином впливають на складність інтерфейсу проектованої реалізації. Тому потрібно чітко розрізняти необхідні та додаткові компоненти експертної системи і першочерговість необхідних реалізацій. Після цього необхідно визначити, чи можливо виконати необхідні дії належним чином. В іншому випадку реалізація системи буде неефективною від початку. Якщо такі дії можуть бути реалізовані, тоді слід також вибрати, відповідно, найкращий спосіб реалізації. *Щодо проектованого прототипу ІМАС, то початково дана система проектувалася як система, заснована на знаннях про предметну область (у нашому випадку – буріння свердловин).*

Серед основних видів мультимедіа, доступних предметній області, перевага надається растровим зображенням: це зумовлюється запропонованим способом прив'язки знань до графічних входжень. Водночас, рухомі зображення та звуковий супровід виконують доповнюючу роль (важливо, що в даному підході рухомі зображення є анімацією певної послідовності статичних зображень, які змінюють одне одного). Загалом, рішення апаратного та програмного забезпечення повинні відображати пріоритети, що передаються різними формами представлення інформації, які система повинна відображати та використовувати. В більшості програмних проектів виконується оцінка програмного та апаратного забезпечення при побудові нової системи прототипу є, як правило, поверхневою, оскільки характеристики програмних пакетів і апаратні компоненти приймаються на основі заявлених характеристик виробниками. Разом із тим, виробники програмного забезпечення вказують тільки мінімальні вимоги для запуску програмного забезпечення, не зазначаючи вимоги для досягнення прийнятної продуктивності (особливо при узгодженні програмного забезпечення в багатозадачному середовищі). Таким чином, обчислювальна потужність наявного обладнання має бути детально вивчена з урахуванням потреб проектованого прототипу. Тому, як правило, не складно визначити мінімальні вимоги для використання, наприклад, відеофайлів, але реальні вимоги для використання усіх форм відео не завжди очевидні. Адже, наприклад, теоретично частота кадрів відеофайлу може встановлюватись програмістом і відео демонструватиметься в зменшеному розмірі.

Водночас, високошвидкісні комп'ютери дозволятимуть перегляд кадрів з високою частотою. Тому для досягнення оптимально ефективного відтворення технологічного процесу (для відео файлів) необхідно зробити оптимальний вибір між швидкістю та точністю. Адже відеокарті, пропущені для досягнення більшої швидкості, можуть містити інформацію, важливу для предметної області. Необхідна швидкість відтворення відеофайлу залежить також від технологічної швидкості перебігу процесів предметної області та відповідної кількості операцій за одиницю часу. З точки зору інтелектуальності система-прототип повинна консолідувати увагу користувача саме на всіх складних деталях при виконанні технологічного процесу. Звуковий супровід, таким чином, більшою мірою виконуватиме задачі запобігання аварійним нештатним ситуаціям і виконуватиме роль попередження щодо некоректно вибраного режиму перебігу технологічного процесу.

Початковим завданням дослідження було розроблення системи, що складається з двох компонентів: власне, сама експертна система й інтерфейс користувача для доступу до системи (в тому числі, і до баз знань у режимі окремого редактора – **Knowledge Base Editor**). Однак, крім цього, необхідно було б виконати включення всіх форм експертних знань (у тому числі відео та растрові зображення) до експертної системи, відокремлених від інтерфейсу.

Важливо відзначити, що знання, представлені у вигляді відео та звуку, будуть, безумовно, частиною бази знань, але не впливатимуть на логічний висновок. Тобто, їх можна розглядати лише як медійні складові інтелімедійної системи. Крім того, одна й та ж експертна система може використовуватися з різними інтерфейсами залежно від задач, які вона має вирішувати. Тому в даному контексті необхідно визначити де, власне, закінчується інтерфейс користувача та починається сутність експертних систем. Адже експертні знання містять інформацію предметної області, яка інтерпретується як взаємодія різних методів (об'єктів і методів об'єктної взаємодії). Разом із тим, меню можна розглядати як загальні інструкції з використання системи, яка є частиною користувацького інтерфейсу. Це зумовлює необхідність вирішення задачі повного відокремлення інтерфейсу від системи (бази знань). Важливо також, що продуктивність системи в цілому залежатиме від місця

зберігання мультимедійних входжень. Зокрема, для досягнення високої швидкодії всі растрові зображення використовуються як частина інтерфейсу, що приховується від користувача доти, поки система не подає на них запит. Таким чином, експертна система покладається на користувацький інтерфейс для отримання растрових зображень. Така взаємозалежність не є ідеальною, оскільки будь-яке оновлення бази знань потребуватиме оновлення користувацького інтерфейсу. Це вноситиме потенційно можливі суперечності, що недопустимо з точки зору ідеології експертних систем, при розробці яких необхідно зберегти всі взаємозалежності між складовими документа (файлу, база знань) для уникнення перехресних взаємозалежностей. Таким чином, виділення двох типів взаємозалежностей, які оновлюються, є необхідними складовими успішної роботи системи (для багатьох користувачів або крос-документних залежностей) коли при оновленні кожної частини, тим не менше, зберігається початкова цілісність системи.

При створенні прототипу системи, доступної певній цільовій аудиторії (в нашому випадку це студенти й обслуговуючий персонал технологічного процесу), необхідно розглянути ряд питань, а саме: програмне забезпечення й апаратні засоби треба підібрати таким чином, щоб вони були доступними для потенційних користувачів. Зрозуміло, що система не може використовуватись, якщо вона не буде доступною в плані початкового оволодіння чи технічного обслуговування. Як показує досвід розробки та впровадження інтелектуальних систем типу «Колектор», «Пласт», «Поклад», «Приплив», «Нафта», співвідношення вартості та корисності таких систем є різним, залежно від рівня впровадження, що варіюється від рівня інтелектуального тренажера до рівня інтелектуального «SCADA»-застосування.

Для досягнення практичного рівня реалізації інтелімедійної технології важливим фактором є розробка системи відповідно до рівня знань цільової аудиторії. Зокрема, для експертної системи важливими є як інтерфейс користувача, так і знання, що містяться в системі. Потенційним користувачам не завжди легко користуватись інтерфейсом такої системи, і вони можуть розчаруватись у можливостях системи. І навіть якщо інтерфейс відповідатиме можливостям користувача, важливим є відповідний рівень представлення знань. Це означає, що

самі знання не можуть бути елементарними, проте вони також не можуть бути й занадто складними. Все це накладає особливі вимоги до цільової аудиторії, структура якої має бути правильно визначеною щодо рівня знань (наявних та потенційних). Водночас, занадто великий ріст рівня знань цільової аудиторії теж суттєво ускладнює проектування системи: з практичної точки зору різні рівні цільової аудиторії вимагатимуть імплементації релевантних рівнів баз знань.

Для побудови завершеного прототипу важливо включити різні підходи до оцінки експертної системи з мультимедіа. Як показує досвід, цей тип системи можна краще оцінити протягом тривалого періоду в плані ефективності використання користувачем знань, отриманих з мультимедіа-версії системи, порівняно з немультимедійними версіями. При використанні конкретно цього прототипу доцільно проводити опитування кожного клієнта системи щодо застосування ним мультимедійної версії системи. Задача полягає у відображенні реального прогресу, що відбувається на основі системи навчання, і, в цьому випадку, йдеться саме про інтелімедійну систему підтримки прийняття рішень у процесі буріння, що базується на знаннях. В загальному, не конектно стверджувати, що така інтелімедійна інформаційна знання-система не є корисною без мультимедійної інформації. Тобто, за відсутності мультимедіа, система працює, в основному, на основі експертного досвіду інтерпретації описів природною мовою, закладених експертом в базу знань, без відповідного підтвердження в зображеннях чи відео. Як показує досвід, невізуалізована інформація запам'ятовується складніше, тобто, фактично, йдеться про неналежний рівень представлення експертних знань. Водночас його можна підвищити застосуванням мультимедійних включень. Більше того, виконаний в монографії (Юрчишин В. М.) [75]. Огляд практичних досліджень показує, що в нафтогазовій предметній області існують форми експертних знань, які взагалі не можуть бути ефективно передані без використання технологій мультимедіа. Тому важливим висновком, отриманим в даному дослідженні, є те, що опора користувача на експертні знання в предметній області стає більш важливою, то використання мультимедійних методів також має збільшуватися пропорційно. В проектованій

системі-прототипі необхідно також усунути протиріччя як на рівні «чистих текстових знань», так і на рівні знань, що опираються на мультимедійні включення.

Представлене дослідження містило проектування, розробку й оцінку прототипу мультимедійної експертної системи (інтелімедійної інформаційної системи). Проектування дизайну системи передбачає багато труднощів. Цифрове представлення ілюстративних та відеоданих про процес буріння ускладнюється неточністю процедури сканування, тому необхідні затрати на їх налаштування. Перетворення та стиснення оцифрованого відео теж нерозривно пов'язане з втратою інформації. Як зазначалось, комп'ютерні експертні знання повинні бути представлені таким чином, щоби відповідати представленню знань, які використовуються екпертом. Як правило, експертні знання, які традиційно вимагають візуалізації, найкращим чином можуть бути представлені за допомогою оцифрованих фотографій або рисунків. Знання, які необхідно презентувати, потребують певної форми анімації або відео.

Таким чином, загальна оцінка отриманого прототипу ІМАС показала, що мультимедійні інтерфейси можуть значно покращити якість подання інформації для нафтогазової предметної області в експертних й інтелектуальних системах відповідно.

РОЗДІЛ 2

ФОРМАЛЬНІ ОСНОВИ МОДЕЛЮВАННЯ КОНТЕНТУ ІНТЕЛІМЕДІЙНОЇ СИСТЕМИ

2.1 Аналіз досліджень в галузі представлення та задоволення обмежень

Класичним підходам до проблеми представлення та задоволення обмежень (*CSP - Constraint satisfaction problems*), інформаційно-пошукових задач на основі обмежень присвячено ряд досліджень [76]. Особливому поширенню даної технології сприяла розробка уніфікованих алгоритмів логічного програмування, що дозволило побудувати розширення класичної теорії логічного програмування в формі логічного програмування в обмеженнях (*CLP - Constraint logic programming*) [77]. Суть підходу CSP/CLP полягає у виконанні декларативного опису проблеми на основі множини (системи, ієрархії обмежень) і пошуку її рішення шляхом використання спеціалізованих алгоритмів, що дозволяють виконувати оцінювання рішення в термінах задоволення множин (систем, ієрархій обмежень). Більшість виконаних застосувань оперує із змінними з скінченними доменами. Технологія програмування в обмеженнях використовується для рішення ряду прикладних задач, таких як: складання часових планів, розкладів, виконання задач конфігурації, перевірки стану технічних засобів та систем тощо.

Особливість підходу CSP/CLP полягає також в можливості опису неточних, нечітких, ймовірнісних, можливісних значень, а також значень з ваговими коефіцієнтами. Проте в загальному випадку твердження та судження представлені людиною-експертом не можливо описати тільки на основі жорстких обмежень. Тому використовується також м'які обмеження. Самі пошукові проблеми в залежності від кількості та якості обмежень поділяються на надобмежені, для яких не існує рішень, щоб задовольняли всі накладені обмеження і недообмеженні, для яких існує ряд невпорядкованих рішень. Багато пошукових проблем дозволяють виконувати пошук найкращого (оптимального) розв'язку на основі наперед заданих

критеріїв оптимізації. Як показує досвід, невизначеність, закладена в формулюванні пошукової проблеми, буде переноситись в знайдені розв'язки. Тому особливо ефективними пошуковими задачами на основі обмежень є ті, які дозволяють використовувати систему преференцій у декларативних описах та алгоритмах рішень, що утворює окремий клас пошукових задач з преференціями.

Перспективним напрямом дослідження є формальна теорія задоволення нечітких обмежень [78-79]. Ця теорія є розширенням класичних теорій ймовірності, теорії можливості та теорії нечіткої логіки.

Дослідження цього напрямку виконується в рамках теорії нечітких обмежень і нечітких відношень. З точки зору побудови ефективних інтелектуальних систем існує питання побудови нечіткого чіткого висновку, висновку на основі обмежень та представлення предметної області засобами нечітких множин [80-81].

Щодо ефективності контролю процесу задоволення обмежень важливе значення має процедура оціночного задоволення обмежень [82-84], яка розглядається як спосіб ведення оцінювання на множині обмежень, та як метод представлення преференцій. Такий підхід може бути розширений введенням оціночних значень для змінних, розглядаючи множини преференцій над множиною змінних. Введення вагових значень для змінних дозволяє розглядати множину присвоєнь, виконувати оцінку процесу задоволення обмежень і контролювати значення вагових коефіцієнтів і їх сум у процесі виконання присвоєння та на множині присвоєнь.

Окремим напрямом дослідження можна виділити технологію представлення та задоволення обмежень, на основі введення формальних мета-структур для вираження преференції, наприклад напівкільця [85-87]. Представлення формальної мета-структури дозволяє вводити набори значень для кортежів присвоєнь, а також використовувати формальні операції та мета-структури для побудови проєкцій множин значень й обмежень.

Ефективним методом роботи з обмеженнями є введення систем та ієрархій обмежень [88-89]. Введення ієрархії обмежень дозволяє оперувати з преференціями на рівні систем обмежень, тобто обмеження можна інтерпретувати як жорсткі

(обов'язкові) і м'які (преференційні). Такий підхід визначає послідовність й спосіб задоволення обмежень у процесі пошуку рішення, його уточнення, пошуку оптимального рішення.

Теоретичним і практичним аспектам побудови ієрархії обмежень присвячено ряд досліджень. Основні ідеї цих підходів полягають у логічній інтерпретації м'яких і жорстких ієрархічних обмежень, введенні відношень впорядкування, представленні нечітких (фазифікованих) ієрархій обмежень, введенні відношення часткового впорядкування ієрархії та систем обмежень, означенні ефективних компараторів, інтерпретації ефективності формальних структур, побудові способу оперування з моделями ієрархії та систем обмежень, оцінці ефективності ієрархії та систем обмежень і їх оптимізації.

Оперування рішеннями пошукових задач на основі обмежень виконується на основі концепції простору рішень [90-92]. Відповідно, виконується пошук певного фіксованого рішення, всіх можливих рішень, найкращого (оптимального) рішення згідно заданого критерію (системи критеріїв).

Класично виділяють 3 основні методи рішення пошукових задач на основі обмежень: метод звуження, метод пошуку, метод синтезу. Найбільш поширеним є метод пошуку на основі кортежів значень, який зводиться до технології бектрекінгу. Основною відмінністю пошукових задач на основі обмежень від звичайних пошукових задач є можливість звуження / розширення початкової задачі, та можливість застосування ефективних технік і методик пошуку.

Для рішення пошукових задач на основі обмежень використовуються також методи сумісності та методи поширення обмежень [93]. Суть методу полягає у виключенні несумісних елементів із доменів змінних. Цей метод розширюється при сумісності вершин, сумісності дуг і при n -арній сумісності. Ефективною є техніка «заглядання наперед» (*lookahead algorithms*) [94], суть якої полягає у виконанні присвоєння тільки після перевірки його впливу на майбутні кортежі рішень.

Для існуючих алгоритмів методик і технік рішення задач задоволення обмежень важливо вибрати початковий набір змінних як основного елементу контролю ефективності бектрекінгу та уникнути надглибокого бектрекінгу (як

надвеликої кількості переприсвоєнь і відмін присвоєнь). Ця проблема вирішується використанням евристик упорядкування змінних. Така евристична стратегія може залежати від кількості важливих значень у домені, від кількості задоволених обмежень, від кількості незадоволених обмежень. Використовують статистичне і динамічне впорядкування змінних [95-96]. Ефективним є метод визначення порядку присвоєння значень змінним на основі деякої евристики впорядкування значень.

Більшість алгоритмів базується на деревах пошуку [97]. Використовуються також структурні алгоритми на основі графового представлення та локальні методи пошуку, наприклад, генетичні алгоритми.

Виконання пошуку оптимального рішення, як правило, базується на алгоритмі гілок і границь із введенням евристичної функції для присвоєнь для обчислення оціночного значення виконаних присвоєнь [98-99]. Використання евристичної функції дозволяє оперувати з піддеревами пошуку та виконувати часткове присвоєння.

Метод введення преференцій для обмежень є ефективним інструментом рішення для надобмежених пошукових задач на основі обмежень [100-102]. Розширенням цього методу є задання преференцій змінним, яке дозволяє переходити із заданим способом часткового впорядкування змінних від надобмежених проблем до оптимізаційних [103-104]. Таким чином, отримують нову інтерпретацію обмежень: обмеження зі змінними з мітками, де мітки виконують роль преференцій [105-107]. Цінність даного підходу в тому, що введені мітки носять локальний характер відносно обмежень зі змінними. В якості міток можуть використовуватись також нечіткі значення. Використання обмежень зі змінними з мітками дозволяє обчислювати вагові значення обмежень як ступінь преференції в процедурах прийняття рішення та в процедурах оптимізації. Тому введення міток дозволяє ефективно застосувати пошукові задачі з нечіткими, можливісними і ймовірнісними обмеженнями.

Важливим питанням дослідження в області пошукових задач на основі обмежень є вибір програмної парадигми для реалізації концепції представлення та задоволення обмежень. Більшість дослідників зосереджуються на парадигмі

логічного програмування [108-110], оскільки вона дозволяє ефективно контролювати та розділяти змінні, використовувати декларативні представлення, механізми бектрекінгу й уніфікації. Проекти цього класу, такі як: ECLiPSe [111], CHIP, CLP, PROLOG III [112] – дозволяють будувати ефективні розширення класичного логічного програмування додаванням концепції задоволення обмежень замість парадигми уніфікації. Проте реалізації ефективні, в більшості випадків, тільки для числових обмежень. Особливістю цих проектів є використання розширених джерел пошуку, введення символічних змінних і новітніх евристик і пошукових стратегій. Також варто звернути увагу на подібні реалізації засобами C++, такі як: ILOG [113] Oz [114] – реалізацію мовою Java (проект Jess [115]).

Оскільки одним із найбільш важливих застосувань технології CSP/CLP є задачі планування, то виділяють окремі класи диз'юнктивних і кумулятивних обмежень [116-118].

У пропонованому дослідженні ми прагнемо виконати побудову ІМАС, що забезпечує підтримку рішення технологічних проблем, базуючись на теорії Олсона, основним елементом якої є виділення помилок [119-121]. Основна ідея підходу полягає в тому, що об'єкт ППР допускає помилки при виконанні технологічних задач, рішенні технологічних проблем. Як показує досвід, помилки виникатимуть навіть тоді коли в попередніх сеансах вивчалися правильні (коректні) способи дій і рішень. За цією теорією, джерелом виникнення помилок є сам об'єкт ППР, що допускає помилки, оскільки засвоєнні ним знання є декларативними, а практичне рішення технологічних задач і проблем вимагає процедурних (алгоритмічних) знань.

2.2 Реалізація інтелектуальних функцій інтелімедійної системи

Таким чином, чим більше декларативних знань буде перетворено в процедурні, тим менше виникатиме помилок. Дії ІМАС, пов'язані з обробкою помилок, поділяють на три категорії: індикація (виявлення), розпізнавання (класифікація), виправлення (корекція) помилок. Розпізнавання помилок виконується на основі наявних декларативних (описових) знань в об'єкта ППР.

$$\text{Match}(\text{SubmittedSolution}, \text{ConstrHrch}[\text{ConstrSyst}[\text{ConstrSet}]])$$

Реалізація інтелектуальних функцій ІМАС полягає в такому способі організації взаємодії з об'єктом ППР, коли виконується активація релевантних доменних знань для об'єкту ППР відповідно до профілю його зони найближчого розвитку (*PDZ - ProximalDevelopmentZone*) $PDZ \subset \text{DomainModel}$. Тому ефективність організації зворотного зв'язку в ІМАС можна оцінити за такими критеріями:

1. зазначення місця виникнення помилки (явне або неявне);
2. ідентифікація та класифікація помилки, співвіднесення її з певною ієрархією помилок, виділення складових помилки і її можливих рівнів;
3. виділення релевантних концепцій предметної області як основи побудови коректного рішення:

$$[\text{DomainConception}^{\text{set}}]^R \rightarrow_{\text{ConstrSet}} \text{CorrectSolution}^{\text{set}}$$

Інформаційно-пошукові задачі на основі обмежень формуються шляхом накладання множини (системи, ієрархії) обмежень на деяку скінчену множину змінних. Для кожної змінної вводиться її домен, з множиною можливих значень. Інтерпретація обмеження полягає в контролі кортежів значень для відповідних наборів змінних. Обмеження задаються явно або неявно, всі множини змінних, доменів та обмежень, що розглядаються, є скінченними.

Присвоєнням на множині технологічних проблем (набором помилок запропонованих рішень технологічної проблеми – *SubmittedSolution*) $TP^{\text{set}} = \{TP_j\}_{j=1..j_{\text{max}}}$ будемо вважати відображення Ψ^{TP_j} з множини змінних $Err_1 \subseteq Err_{TP_j}^{\text{Hrch.}}$ в їх відповідні домени, тобто $\Psi^{TP_j}(err_m) \in D_m$ для $err_m \in Err_1$, $\{D_1 \times D_2 \times \dots \times D_m\} \subset \{TPS_k\}_{k=1..k_{\text{max}}}$. Множину всіх можливих присвоєнь для Err_1 будемо позначати $\Psi_{Err_1}^{TP_j}$.

Застосування присвоєння Ψ^{TP_j} в домені Err_1 для кортежів $(err_1, err_2, \dots, err_{k_1})$ дозволить отримати присвоєння $\{(err_1, d_{err_1}), \dots, (err_{k_1}, d_{err_{k_1}})\}$ і отримати кортеж

значень $\Psi^{TP_j} = (d_{err_1}, \dots, d_{err_{k_1}})$, тобто кортеж значень помилок з інтерпретацією $d_{err_p} \rightarrow \{fb_p, R_p\}_{p \in [1..k_1]}$.

Присвоєння Ψ^{TP_j} вважатимемо повним, якщо відповідні йому домени значень $D_{m_1}, m_1 \in [1..m]$ не є порожніми для всіх станів технологічної проблеми TPS_k . В іншому випадку, присвоєння будемо вважати частковим.

Множину всіх присвоєнь домену $Err_{TP_j}^{Hrch.} \Psi_{Err_{TP_j}^{Hrch.}}^{TP_j} \equiv D_1 \times \dots \times D_{n_1}$ будемо називати простором інтерпретації помилок рішень технологічної проблеми TP_j .

Розглядатимемо присвоєння виду $\Psi^{TP_j} = (err_1, err_2, \dots, err_{k_1})$. Нехай щодо обмеження $c_m \in ConstrHrch_{TP_j}$ цей набір помилок має ранжування виду $(err_1 : rd_1^{c_m}, err_2 : rd_2^{c_m}, \dots, err_{k_1} : rd_{k_1}^{c_m})$. Обчислимо середнє значення релевантності множини помилок з присвоєння обмеженню c_m :

$$AvgErr^{c_m} = (rd_1^{c_m} + rd_2^{c_m} + \dots + rd_{k_1}^{c_m}) / k_1$$

Будемо вважати, що обмеження c_m задовольняється присвоєнням Ψ^{TP_j} (позначимо $\Psi^{TP_j} \in \check{E}^{SD^{c_m}}$) зі ступенем $sd^{c_m} = 1 - AvgErr^{c_m}$. Відповідно, обмеження c_m порушується присвоєнням Ψ^{TP_j} зі ступенем $vd = AvgErr^{c_m}$.

Часткове присвоєння Ψ^{TP_j} будемо вважати послідовним, якщо всі релевантні йому обмеження порушуються.

Попередній виклад показує, що формальний опис технологічних проблем слід здійснювати введенням відповідних множин, систем та ієрархій обмежень. Також можна вводити якісні та кількісні характеристики процесу, такі як: ступінь задоволення обмежень, концепція рішення технологічної проблеми на основі помилок – а також представляти й обчислювати ступінь по зв'язаності та ступінь послідовності. Введене нами означення технологічної проблеми можна розглядати як розширення класичної ідеї інформаційно-пошукових задач на основі обмежень. Суть такого розширення полягає в наступному:

1. введенні впорядкування обмежень у формі систем і ієрархій;

2. введенні оціночних значень для обмежень, що дозволяє виражати такі якісні та кількісні характеристики як релевантність, вагове значення, імовірнісне значення, можливісне значення та преференції;

3. структуризації технологічних проблем за їх станами та впорядкуванні множини технологічних проблем у вигляді технологічного процесу;

4. інтерпретації введених станів технологічних проблем відносно представлення доменів значень множини змінних;

5. введенні помилок як множини змінних;

6. введенні оціночних значень для помилок, що дозволяє виражати вагу помилок і релевантність обмежень щодо поточної помилки;

7. паралельному представленні обмежень у вигляді правил для реалізації інтелектуальної функції ІМАС на їх основі.

Основне завдання, яке вирішувалося при формальному обґрунтуванні побудованого розширення, полягає в дослідженні можливості реалізації інтелектуальних функцій ІМАС, можливості реалізації інтелектуальної системи на основі обмежень у вигляді системи на основі правил. Тому отримане формальне представлення слід класифікувати як представлення гібридної інтелектуальної системи на основі обмежень і правил. Перевагою використання правил є можливість привнесення в ІМАС класичних методів інтелектуалізації. Відповідно, перевагою використання обмежень є можливість організації бази знань у формі представлень із природномовною інтерпретацією.

Введення означення ступеня задоволення (*sd – satisfaction degree*) для технологічних проблем виконується з тією ж метою, що і для класичних інформаційно-пошукових задач: введення можливості порівняння й оцінки виділених присвоєнь Ψ - класу. Проте в цьому випадку самі присвоєння інтерпретуються більш широко: від класичного накладення обмежень до аналізування порушень обмежень і помилок (виокремлених помилок, множин помилок, систем помилок, ієрархій помилок). Також у запропонованому підході розширено інтерпретацію рішення для технологічних проблем. Адже в даному випадку завдання ІМАС полягає не стільки в пошуку рішення технологічної

проблеми (зокрема, на основі методології *CSP*), скільки в забезпеченні контролю за процесом формування рішення на основі виділення й аналізу помилок об'єкта ППР.

Для класичних *CSP* концепція рішення вводиться як інтерпретація виконаного присвоєння щодо його достатності або оптимальності. В нашому випадку, інтерпретація рішення присвоєнням Ψ - класу виконується на основі центрального елементу – помилок, що виникають у процесі рішення. Тому інтерпретація присвоєння з центральним елементом у вигляді помилок – це спосіб реалізації інтелектуальності ІМАС, що базується на прив'язування правил до обмежень. Тому в досліджуваному випадку заслуговує на увагу введення ступеня зв'язаності (послідовності) рішення технологічної проблеми у вигляді певного способу оцінювання простору оптимальних присвоєнь із накладеними обмеженнями:

$$Ev([\Psi]_{class}^{opt} | =).$$

На сьогодні в дослідженнях теорії та практики методів та засобів представлення та задоволення обмежень розвинуто ряд ефективних методик і технік. Найбільш важливими серед них є: 1) метод вагових коефіцієнтів; 2) метод вартісних коефіцієнтів; 3) методи на основі оптимізаційних технік [122].

З точки зору базового формалізму задач *CSP* у вигляді кортежу (V, D, C) дані методики працюють шляхом звуження відповідних доменів, максимізації або мінімізації значень для кортежів присвоєнь у введених множинах, системах і ієрархіях обмежень. Також деякі методики дозволяють контролювати кількість задоволених і порушених обмежень.

Застосування ймовірнісного підходу [123-125] для введеного формального представлення технологічних проблем на основі обмежень, дозволяє кількісної оцінки релевантності виділених обмежень у поточній технологічній проблемі. Такий підхід характеризуватиметься введенням невизначеності нового типу відносно до невизначеності, яка описувалась формальним підходом на основі вагових коефіцієнтів. Це дозволяє моделювати процес порушення та задоволення обмежень на рівні множин, систем та ієрархій.

Введена формальна інтерпретація функції успішності дозволяє виконувати обчислення рішень з мітками, а саме: успішність множини, системи та ієрархії обмежень буде визначатися безпосередньо важливістю комплексною мітки як функції від її складових. Хоча введення комплексних міток виглядає складним з точки зору практичної реалізації функції кількох параметрів, проте в реальних застосуваннях частину входжень в правій частині формули $l_i^{set} = \{rd_i, cw_i, cpr_i, cps_i, pfc_i, nd_i\}$ заміняють граничними значеннями 0,1. Це дозволяє отримати задачу інтерпретації міток задовільно рівня. При складності використання лінгвістичних значень для представлення міток застосовується процедура дефазифікації, що переводить лінгвістичні значення в значення з інтервалу $[0..1]$, який використовується для представлення функції належності. Оскільки рішення технологічних проблем на основі обмежень, незалежно від вибраного способу індексації міток, зводиться до обчислення певної послідовності максимальних і мінімальних значень, то, фактично, з'єднувальні функції $\oplus_{TP_j}, \oplus_{TP_j.TPS_k}$ відіграватимуть роль функції обчислення середнього значення. Тому при практичній реалізації ІМАС йдеться про оцінку співвідношення кількості та якості міток, що повинно відповідати перевагам і стратегіям експертів предметної області. Оскільки введена формалізація базується на концепції обчислення глобальної мітки, то, очевидно, результуюча мітка може бути як сумаризацією великої кількості міток з невисоким ваговим значенням так і представленням лише кількох важливих міток високої ваги. Пошук рішень полягає в переборі присвоєнь і, відповідно, кожне з таких присвоєнь активуватиме власну систему переваг обмежень. Тому процес контролю за пошуком рішення, очевидно, зводиться до роботи з обчислювальними результатами функції успішності. Щодо практичної реалізації доцільно використовувати два підходи до інтерпретації функції успішності: тривіальний і нетривіальний. В першому випадку функція успішності повертає значення "1", якщо обмеження задовольняється, і "0", якщо обмеження порушується. В другому випадку, який реалізовується в ІМАС, функція успішності

повертає значення в діапазоні $[0..1]$, що дозволяє розширити адаптивність і гнучкість системи введенням методів невизначеності.

В ряді досліджень представлено формалізацію підходу часткового задоволення обмежень. Метою даних підходів є забезпечення пошуку оптимального рішення шляхом контролю метричного ранжування описаних технологічних проблем за рівнями «недообмежені – надобмежені». Дані дослідження показують, що імплементація метрики такого ранжування неможлива в рамках єдиної, строго визначеної формальної моделі, тому моделлю ефективного рівня, в даному випадку, вважається метамодель.

Послаблення (посилення) задач виконується шляхом:

1. збільшення (звуження) доменів змінних;
2. збільшення (звуження) доменів обмежень;
3. видалення (додавання) змінних;
4. видалення (додавання) обмежень.

Основна ідея побудови рішення технологічної проблеми на основі обмежень, викладена вище в різних інтерпретаціях, полягає в тому, що кожна множина релевантних обмежень розпадається на дві підмножини, що не перетинаються: множину задоволених обмежень і множину порушених обмежень. Тому побудову оцінки рішення можна виконати на основі двох факторів впливу:

1. кожне задоволене обмеження повертає значення «1», кожне порушене обмеження – значення «0»;

2. мітки обмежень виконуватимуть роль мультиплікативних факторів, що дозволяє враховувати вагу обмежень, преференцію обмежень тощо.

Згідно введеної ідеї простору технологічних проблем і використовуючи отримані функції, можна сформулювати наступні означення.

Під технологічною проблемою на основі обмежень з *частковими рішеннями* будемо розуміти кортеж $TP_j^{partial} = \langle TP, (STP, \leq), Mt, Sol^{ideal}, Sat(), Viol() \rangle$, де: TP – початкова задача на основі обмежень; STP – її простір; Mt – метрика, введена на даному просторі; Sol^{ideal} – ідеальне рішення для проблеми; $Sat()$ – функція

обчислення вагових значень за задоволеними обмеженнями; $Viol()$ – функція обчислення вагових значень за порушеними обмеженнями. За допомогою ідеальної функції та функції обчислення вагових значень можна виконувати контроль базової задачі та її простору, що на фактичному рівні STP буде містити послаблення та посилення початкової проблеми:

$$STP = \left\langle weak^i(TP_j), strong^i(TP_j) \right\rangle_i.$$

Як необхідне часткове рішення технологічної проблеми на основі обмежень $TP_j^{partial}$ слід розглядати задачу TP' із простору проблем STP , разом з її рішенням $Sol'(TP')$, де метрична відстань

$$Mt^\varphi(TP', TP) < dist^\varphi(|[Viol(TP')] - [Viol(TP)]|).$$

Часткове рішення будемо вважати *достатнім*, якщо

$$Mt^\varphi(TP', TP) < dist^\varphi(|[Sat(TP')] - [Sat(TP)]|).$$

Часткове рішення будемо вважати *оптимальним*, якщо метрична відстань між проблемами мінімальна:

$$Sol_{optimal}^{partial}(TP_j^{partial}) = \min_{STP} Mt(TP', TP)$$

Крім того, матимемо, що $SD(Sol^{partial}) = Mt(TP', TP)$, $CD(Sol^{partial}) = SD(Sol_{optimal}^{partial}) = \min_{STP} Mt(TP', TP)$. Тобто ступенем задоволення задачі $TP_j^{partial}$ вважати значення функції метричної відстані, а ступенем послідовності – ступінь задоволення оптимального рішення.

Обмеженням з оцінкою будемо вважати кортеж $(c_i, evf(c_i))$, де c_i – обмеження з ієрархії $ConstrHrch_{TP_j}^R$, а evf – функція оцінки обмежень ($evf : ConstrHrch_{TP_j}^R \rightarrow EvSet = ExpertEv \cup UserEv$, де $EvSet$ – загальна множина оцінок, $ExpertEv$ – множина оцінок експерта предметної області, $UserEv$ – множина оцінок користувача ІМАС).

Технологічну проблему з накладеними обмеженнями з оцінками TP_j^{Ev} визначимо як формальну структуру

$$TP_j^{Ev} = (Err_{TP_j}^{Hrch.}, Err_{TP_j}^{Domain}, ConstrHrch_{TP_j}^R, (EvSet, \otimes, \succ), evf),$$

де \otimes – операція поєднання оцінок обмежень, \succ – операція впорядкування на множині оцінювань $EvSet$.

Для заданої технологічної проблеми з накладеними обмеженнями з оцінками TP_j^{Ev} і деякого присвоєння $\psi \in \Psi_{Err_{TP_j}^{SubHrch.}}$, де $Err_{TP_j}^{SubHrch.} \subseteq Err_{TP_j}^{Hrch.}$, оцінювання присвоєння ψ відносно до TP_j^{Ev} визначимо як:

$$Ev_{TP_j^{Ev}}(\psi) = \left[\begin{array}{c} \otimes \\ (c_i^s, c_i^v \in ConstrHrch^R) \wedge (c_i^{var} \subseteq Err_{TP_j}^{SubHrch.}) \wedge [\psi \in c_i^v \vee \psi \notin c_i^s] \wedge ConstrHrch^{unR} = \emptyset \end{array} \right]_{i \in [1..k_{max}, 1..n_k]} evf(c_i^r),$$

де c_i^{var} – множина змінних обмеження, $ConstrHrch^{unR}$ – ієрархія нерелевантних обмежень.

Таким чином, оцінювання присвоєння здійснюється шляхом поєднання оцінок окремих обмежень, що дозволяє відбирати рішення на основі оцінки ступеня задоволення присвоєнь

$$Ev(Sol^{potential}) \forall [SD(\psi) | EV_{TP_j^{Ev}}(\psi)]$$

Присвоєння $\psi^* \in Err_{TP_j}^{Hrch.}$ будемо вважати рішенням технологічної проблеми з накладеними обмеженнями з оцінками TP_j^{Ev} якщо воно має мінімальне оціночне значення відносно операції впорядкування оцінювань \succ :

$$Sol(TP_j^{Ev}) = \psi^* | \min_{\succ} Ev_{TP_j^{Ev}}(\psi^*)$$

Оцінка технологічної проблеми TP_j^{Ev} дорівнює ступеню її послідовності та відповідає мініимальному оціночному значенню:

$$Ev(TP_j^{Ev}) = CD(TP_j^{Ev}) = \min_{\succ} Ev_{TP_j^{Ev}}(\psi), \psi \in Err_{TP_j}^{Hrch.}.$$

Спосіб виконання пошуку оціночних значень дозволяє визначити верхній і нижній елемент на множині оцінювання таким чином:

$$\min_{\succ} Ev_{TP_j^{Ev}}(\psi) \Rightarrow \left[\begin{array}{l} \perp \rightarrow \otimes_{i \in [1..k_{max}, 1..n_k]} c_i^s, \psi \in \forall c_i, \psi \in c_i^s \in ConstrHrch_{TP_j}^S \\ \top \rightarrow \otimes_{i \in [1..k_{max}, 1..n_k]} c_i^v, \psi \notin \exists c_i, \psi \in c_i^v \in ConstrHrch_{TP_j}^V \end{array} \right]$$

Нехай TP_j^{Ev} - технологічна проблема з накладеними обмеженнями з оцінками.

Послабленням технологічної проблеми LP_j^{Ev} будемо вважати технологічну проблему

TP_j , за умови, що $ConstrHrch_{TP_j}^S \subset ConstrHrch_{TP_j^{Ev}}^S$ і $ConstrHrch_{TP_j}^V \subset ConstrHrch_{TP_j^{Ev}}^V$,

$TP_j = weak(TP_j^{Ev})$. Для технологічної проблеми з накладеними обмеженнями з

оцінками TP_j^{Ev} і деякого її послаблення TP_j^1 оцінку значення послаблення будемо

виконувати так:

$$Ev_{TP_j^{Ev}}(TP_j^1) = \left[\begin{array}{c} \textcircled{*} \\ \forall c_i \in ([ConstrHrch_{TP_j^{Ev}}^R][ConstrHrch_{TP_j^1}^R \cup ConstrHrch_{TP_j^1}^{unR}]) \\ evf(c_i) \end{array} \right]_{i \in [1..k_{max}, 1..n_k]}$$

Для заданої технологічної проблеми з накладеними обмеженнями з оцінками

TP_j^{Ev} і двох її послаблень (TP_j^1 та TP_j^2) матимемо, що:

$$[ConstrHrch_{TP_j^1}^R \cup ConstrHrch_{TP_j^1}^{unR}] \subset [ConstrHrch_{TP_j^2}^R \cup ConstrHrch_{TP_j^2}^{unR}] \Rightarrow$$

$$\Rightarrow Ev_{TP_j^{Ev}}(TP_j^1) \succeq Ev_{TP_j^{Ev}}(TP_j^2)$$

Присвоєння $\psi \in \Psi_{Err_{TP_j}^{Hrch}}$ вважатимемо розв'язком технологічної проблеми з

накладеними обмеженнями з оцінками TP_j^{Ev} , якщо

$$[\psi \rightarrow Sol(TP_j^{Ev})] \in [\psi \rightarrow Sol(TP_j^1)] \mid \min Ev_{TP_j^{Ev}}(TP_j^1),$$

де $TP_j = weak^{consist}(TP_j^{Ev})$.

2.3 Введення технологічних проблем

Таким чином, технологічну проблему з накладеними обмеженнями з оцінками

TP_j^{Ev} можна розглядати як технологічну проблему з частковими рішеннями $TP_j^{partial}$,

якщо виконати такі дії:

1. для кожного обмеження $c_i^r \in ConstrHrch_{TP_j}^R$ ввести деякий домен, що

відповідатиме його простору $SpaceDomain^{c_i^r}$;

2. побудувати метричну функцію для просторів обмежень Mt^{Space} ,
 $Mt^{Space} \rightarrow \bigoplus_{k=1..k_{max}}^{Space} \{SpaceDomain^{c_i}\}_{i=1..n_k}$, де \bigoplus^{Space} – з'єднувальна функція просторів
 обмежень;

3. ввести структуру послаблення (посилення) як метричну різницю між відповідними ієрархіями обмежень.

Нехай ми маємо множину всіх можливих послаблень $Weak_{TP_j^{Ev}}^{Set}$ і множину всіх можливих посилень $Strong_{TP_j^{Ev}}^{Set}$ для технологічної проблеми TP_j^{Ev} . На їх основі можемо отримати впорядковані множини:

$$\left(Weak_{TP_j^{Ev}}^{Set}, \{ConstrSet_{i_1}^{del}\}_{i_1=1,2,\dots}, \succeq\right), \left(Strong_{TP_j^{Ev}}^{Set}, \{ConstrSet_{i_2}^{add}\}_{i_2=1,2,\dots}, \succeq\right).$$

Виконаємо побудову послідовного послаблення з максимальною множиною обмежень

$$TP_j^1 = weak^{consist}(TP_j^{Ev}) | ConstrSet_{TP_j^1}^{max}, \quad \text{де}$$

$$ConstrSet_{TP_j^1}^{max} = max[ConstrHrch_{TP_j^{Ev}}^R \setminus ConstrHrch_{TP_j^1}^R].$$

Можна порівняти відмінності між послабленою (посиленою) та початковою проблемами відносно множин їх рішень, а саме:

$$dist_{min}^{Mt^{LP}}(TP_j^{Ev}, TP_j^1) \Rightarrow dist_{min}^{Mt^{Sol}}(Sol(TP_j^{Ev}), Sol(TP_j^1)).$$

Крім того, можна перепорядкувати множини послаблень і посилень на основі обчислення ступеня задоволення початкових оцінок:

$$\left(Weak_{TP_j^{Ev}}^{Set}, \{Ev_{TP_j^{Ev}}(TP_{j_1}^{i_1})\}_{i_1, j_1=1,2,\dots}, \succeq\right), \left(Strong_{TP_j^{Ev}}^{Set}, \{Ev_{TP_j^{Ev}}(TP_{j_2}^{i_2})\}_{i_2, j_2=1,2,\dots}, \succeq\right),$$

$$\text{де } TP_{j_1}^{i_1} \in Weak_{TP_j^{Ev}}^{Set}, TP_{j_2}^{i_2} \in Strong_{TP_j^{Ev}}^{Set}.$$

Для дослідження повноти та комплексності технологічних проблем на основі обмежень як окремого підкласу CSP доцільною є оцінка відображень класу:

$$1. ConstrHrch^{fuzzy} \rightarrow TP^{pos};$$

$$2. ConstrHrch^{fuzzy} \rightarrow TP^{fuzzy},$$

А саме побудови проєкції ієрархії обмежень (відповідно, систем і множин обмежень) із лінгвістичними мітками на класи можливісних і нечітких технологічних проблем.

Визначимо можливісну задачу TP_j^μ як кортеж

$TP_j^\mu = (Err_{TP_j}^{Hrch.}, Err_{TP_j}^{Domain}, \{c_i, lg_i^{Constr}(c_i) \mid c_i \in ConstrHrch_{TP_j^\mu}^R\})$, де ієрархія обмежень містить глобальні мітки, що визначаються на основі оцінок необхідності та їх ступенів преференцій: $lg_i^{Constr}(c_i) \in c_i[bd_i : pfc_i]$.

Для деякого присвоєння $\psi \in \Psi_{Err_{TP_j^\mu}^{Hrch.}}$ визначимо, що ψ є оптимальним присвоєнням для TP_j^μ , якщо ψ є рішенням для ієрархії $ConstrHrch_{TP_j^\mu}^R$ із максимальною функцією успішності:

$$[\psi^{optimal} \rightarrow TP_j^\mu] \in (\psi \in Sol(ConstrHrch_{TP_j^\mu}^R) \mid \max_{SF} [SF(ConstrHrch_{TP_j^\mu}^R \psi)]).$$

Функцію успішності можна розглядати як ступінь послідовності для TP_j^μ .

Визначимо технологічну проблему з нечіткими обмеженнями TP_j^{fuzzy} як кортеж $(Err_{TP_j}^{Hrch.}, Err_{TP_j}^{Domain}(c_{L_{TP_j^{fuzzy}}^1}^1, \dots, c_{L_{TP_j^{fuzzy}}^m}^m))$, де $C_{L_{TP_j^{fuzzy}}}^{Set}$ - множина нечітких обмежень. Відповідно, кожне нечітке обмеження $c_{L_{TP_j^{fuzzy}}}^i$ і звичайне обмеження c^i визначимо над деякою скінченною множиною змінною $\{e_{i_1}, \dots, e_{i_k}\}$. Введемо нечітке відношення виду:

$$mf_{c_{L_{TP_j^{fuzzy}}}^i} \{e_{i_1}^{dom}, \dots, e_{i_k}^{dom}\} = SF(c_i^s(e_{i_1}^{dom}, \dots, e_{i_k}^{dom})) \oplus_{TP_j^{fuzzy}} lg^{Constr}(c_i^s),$$

де $e_{i_j}^{dom} \leftarrow e_{i_j}$ - відображення змінних на домени їх значень.

Присвоєння $\psi \in \Psi_{Err_{TP_j^{fuzzy}}^{LP_j^{fuzzy}}}$ буде рішенням технологічної проблеми TP_j^{fuzzy} виду

$$Sol \rightarrow \bigwedge_{i=1..m} \left[\bigoplus_{TP_j^{fuzzy}} c_i \in C_{L_{TP_j^{fuzzy}}}^{Set} \right] \text{ якщо } \psi \text{ є рішенням для } [TP_{L_{TP_j^{fuzzy}}}^{Hrch.}]^\mu, \psi = Sol([TP_{L_{TP_j^{fuzzy}}}^{Hrch.}]^\mu)$$

та $SF(C_{L_{TP_j^{fuzzy}}}^{Set} \psi) = CD(TP_j^{fuzzy})$.

Технологічною проблемою з обмеженнями з ієрархічними мітками будемо вважати формальну структуру:

$$TP_j^{Hrch} = (Err_{TP_j}^{Hrch.}, Err_{TP_j}^{Domain}, ConstrHrch_{TP_j}^R, ([0..1] \subseteq \square, \geq, \bigoplus_{Hrch}^{set}), l_{Hrch}^{set}),$$

де $l_{Hrch.}^{set}$ – ієрархічна мітка, $\oplus_{l_{Hrch.}^{set}}$ – з'єднувальна функція для ієрархічних міток.

Верхня границя значень міток може обмежуватись значенням “1” на рівнях ієрархії або, що найбільш зручно, значеннями $\langle 0, r \rangle, r \in \mathbb{R}$ для всієї ієрархії.

Розглянемо випадок, коли ієрархія міток має n -рівнів. Відповідно, кожен із рівнів представляється системами обмежень, що не перетинаються:

$$ConstrHrch = (ConstrSyst_1 = (ConstrSet_1, \geq_R), \dots, ConstrSyst_n = (ConstrSet_n, \geq_R))$$

$$ConstrHrch = ConstrSyst_1 \cup, \dots, \cup ConstrSyst_n$$

$$ConstrSyst_i \cap ConstrSyst_j = \emptyset, \forall i, j \in [1..n]$$

Розміщення обмежень за рівнями ієрархії відповідно їх вагових коефіцієнтів (міток) може бути довільним або впорядкованим, коли обмеження з більш високими ваговими мітками знаходяться, наприклад, на початкових або кінцевих рівнях.

Таким чином, можна ввести впорядкування ієрархії обмежень за індексами її рівнів:

$$ConstrHrch = \{ConstrSyst_i, \prec_{index}\};$$

$$ConstrSyst_i \prec_{index} ConstrSyst_j \Leftrightarrow i < j; i, j \in [1..n].$$

Водночас, варто враховувати вищевведений спосіб обчислення вагових коефіцієнтів. Тому природним буде також введення впорядкування за ваговими коефіцієнтами:

$$ConstrHrch = \{ConstrSyst_i, \prec_{weight}\}$$

$$ConstrSyst_i \prec_{weight} ConstrSyst_j \Leftrightarrow Weight(ConstrSyst_i) < Weight(ConstrSyst_j),$$

для $Weight(ConstrSyst_i) = \sum_{k=1}^{k_{max}} cw_k$, де k_i - кількість обмежень в системі на i -тому

рівні, $ConstrSyst_i$ - система обмежень i - того рівня.

Нехай задано присвоєння $\psi \in \Psi_{Err_{TP}^{Hrch}}$. Його оцінку можна виконати так:

$$SD(\psi) = \sum_{i=1}^{k_{max}} \left[\sum_{k=1}^{k_i} SD(c_k) \right]_i = \sum_{i=1}^{k_{max}} \sum_{k=1}^{k_i} [SF(c_k)]_i. \text{ Тобто, розглядаючи рівень задоволення}$$

присвоєння сумаризацією рівнів задоволення обмежень на основі функції

успішності,

отримати,

що

$$\{SF(c_k) \cup Ig^{Constr}(c_k)\}_{k=1..k_i} \dot{\in} \{SF(ConstrSyst_i)\}_{i=1..k_{max}} \dot{\in} SD(\psi).$$

Успішність системи обмежень $ConstrSyst_i^1$ відносно присвоєння $\psi \in \Psi_{Err_{TP_j}^{Hrch}}$ будемо розглядати як відображення:

$$SF(ConstrSyst_i^1 \subset ConstrSyst_i) \Big|_{\psi \in \Psi_{Err_{TP_j}^{Hrch}}} : \Psi_{Err_{TP_j}^{Hrch}} \rightarrow (W_1 \leq_w).$$

Для технологічної проблеми з обмеженнями з ієрархічними мітками TP_j^{Hrch} і ієрархії обмежень із мітками

$$\left[ConstrHrch_{TP_j^{Hrch}} \right]_{set}^{Hrch} = ConstrSyst_1^{iHrch} \cup \dots \cup ConstrSyst_n^{iHrch}, \quad \text{успішність ієрархії}$$

відносно присвоєння ψ будемо визначати як:

$$SF(ConstrHrch_{TP_j^{Hrch}} \psi) = [SF(ConstrSyst_1^{iHrch} \psi) \oplus_{SF} \dots \oplus_{SF} SF(ConstrSyst_n^{iHrch} \psi)], \quad \text{де}$$

\oplus_{SF} – з'єднувальна функція для значень функції успішності.

Рішенням для технологічної проблеми з ієрархічними мітками TP_j^{Hrch} будемо вважати таке присвоєння ψ , що значення кожної функції успішності $SF^i(ConstrSyst_i \psi)_{i=1..k_{max}}$ є максимальним:

$$\forall \psi^1 \in \Psi_{Err_{TP_j}^{Hrch}} \exists i_1$$

таким, що

$$\forall i < i_1 : [SF(ConstrSyst_i \psi) \leq_w SF(ConstrSyst_i \psi^1)] \wedge \\ \wedge [SF(ConstrSyst_{i_1} \psi) <_w SF(ConstrSyst_{i_1} \psi^1)], i, i_1 = 1..k_{max}$$

Оскільки, згідно нашого припущення, рівень $ConstrSyst_1$ є найбільш важливим, то, відповідно, його ступінь задоволення теж буде максимальним:

$$SD(ConstrSyst_1) \rightarrow [\forall c_k, c_k \in ConstrSyst_1, \psi \dot{\in} c_k, \mathbf{b} = 1..i_k]$$

Таким чином, успішність присвоєнь можна оцінювати за рівнем ієрархії:

$$SF(ConstrSyst_{TP_j^{Hrch}}^{Hrch} \psi) = \oplus_{SF} SF[ConstrSyst_i \psi]_{i=1..k_{max}}$$

З цього випливає, що навіть не надто високі результати функції успішності на більш важливому рівні визначатимуть краще присвоєння, ніж більш високі значення функції успішності на менш важливих рівнях.

Відповідно, функція успішності може мати додаткові інтерпретації, а саме:

1. успішність за ваговими сумами:

$$SF(ConstrHrch_{TP_j^{Hrch}}^1, \psi) = \sum_{c_i \in ConstrHrch_{TP_j^{Hrch}}^1} [lg^{Constr}(c_i) SF(c_i \psi)]_{i=1..i_{max}}$$

2. успішність у найбільш негативному випадку:

$$SF(ConstrHrch_{TP_j^{Hrch}}^1, \psi) = \min_{c_i \in ConstrHrch_{TP_j^{Hrch}}^1} [lg^{Constr}(c_i) SF(c_i \psi)]_{i=1..i_{max}}.$$

Виконаємо оцінку успішності окремих (виділених) обмежень на основі введеної ідеї глобальних міток $(lg^{Constr}(c), SF(c\psi))$:

$$SF(ConstrHrch_{TP_j^{Hrch}}^1, \psi) = [(lg^{Constr}(c_1) SF(c_1 \psi)), \dots, (lg^{Constr}(c_i) SF(c_i \psi))] \text{ для}$$

$$ConstrHrch_{TP_j^{Hrch}}^1 = \{c_1, \dots, c_i\}_{i=1..k_{max}} \subseteq ConstrHrch_{TP_j^{Hrch}}.$$

Наступним кроком є введення рішення технологічної проблеми на основі обмежень із ієрархічними мітками обчисленням функції успішності для локальних обмежень:

$$\begin{aligned} Sol(TP_j^{Hrch}) &= [(l_1^{set}, err_1), \dots, (l_k^{set}, err_k)] <_{set} [((l_1^{set'}, err_1'), \dots, (l_{k'}^{set'}, err_{k'}'))] \equiv \\ &\equiv (\exists i_1 : err_{i_1} < err_{i_1}') \wedge (\forall i_2 : l_{i_2}^{set} \geq_{\mathbb{R}} l_{i_2}^{set'} \Rightarrow err_{i_2} \leq err_{i_2}') \end{aligned}$$

Таким чином, контролювання пар значень $(lg^{Constr}(c_i), SF(c_i \psi))$ дозволяє також контролювати задоволені обмеження присвоєнням ψ :

$$[lg^{Constr}(c_i), SF(c_i \psi) | SF(c_i \psi) = 0]_{\forall i, i=1..k_{max}} \vdash [\psi \in ConstrSyst_{TP_j}^i]$$

За аналогією з дослідженням відповідності між множинами нечітких міток та технологічними проблемами на основі нечітких обмежень – $L_{set}^{fuzzy} \xleftarrow{def} [TP_j^{fuzzy}]^{set}$ – можна описати відповідність між ієрархічними мітками й ієрархіями обмежень для технологічних проблем $L_{set}^{Hrch} \xleftarrow{def} ConstrHrch_{TP_j}$.

Для технологічної проблеми з ієрархічними мітками $TP_j^{Hrch} = (Err_{TP_j}^{Hrch.}, Err_{TP_j}^{Domain}, ConstrHrch_{TP_j}^R, ([0..1] \subseteq \square, \geq, \oplus_{l_{Hrch}^{set}}), l_{Hrch}^{set})$ і заданою ієрархією обмежень $ConstrHrch_{TP_j} = \bigcup_{i=1}^{k_{max}} ConstrSyst_i$, присвоєння $\psi \in \Psi_{Err_{TP_j}^{Hrch.}}$ з функцією $lg^{Constr}(c_i)$ як функції обчислення вагових значень обмежень, можна розглядати як рішення технологічної проблеми TP_j^{Hrch} для таких класів рішень: 1) з функцією успішності для вагових сум; 2) як краще рішення за ваговими сумами.

Нехай задано технологічну проблему з невпорядкованою системою міток $TP_j^{L^{set}} = (Err_{TP_j}^{USyst}, Err_{TP_j}^{Domain}, ConstrSyst_{TP_j}^U, (L^{set}, \prec, \oplus_{c_i, c_j}^{i \neq j}), l)$, де $Err_{TP_j}^{USyst}$ і $ConstrSyst_{TP_j}^U$ – невпорядковані домени змінних й обмежень, $l \in L^{set}$ – представлення міток, $\oplus_{c_i, c_j}^{i \neq j}$ – з'єднувальна функція для обмежень із невпорядкованими мітками. Введемо динамічну глобальну мітку змінної $dlg_W^{Constr.}(err)$, яка виконуватиме з'єднання міток змінної в усіх обмеженнях $ConstrSyst_{TP_j}^U \psi$, де вона зустрічається:

$$dlg_W^{Constr.}(err_{i_1}) : \psi_W \rightarrow L^{set}, \quad dlg_W^{Constr.}(err_{i_1} \psi) = \bigoplus_{(c_i \in ConstrSyst_{TP_j}^U) \wedge (err_{i_1} \in W \psi_{[err_{i_1}]})} l(c_i, err_{i_1}).$$

У процесі пошуку рішення для присвоєння $\psi : (e_1, e_2, \dots, e_n)$ на кожному кроці множина змінних є впорядкованою згідно виконаних присвоєнь $\overline{\text{sat.}}(e_1, e_2, \dots, e_k, e_{k+1}, \dots, e_n)$. Тому саме використання динамічних міток дозволяє впорядковувати решту змінних, яким ще не були присвоєні значення. Наприклад, для виділеної змінної err_{i_1} перед присвоєнням їй значення, її глобальна мітка повинна мати максимальний ваговий коефіцієнт серед інших змінних.

Означення 2.1. В якості мітки (*label*) [126-128] розглядається пара змінна-значення $\langle v, z \rangle$, яка присвоює змінній певне значення з її домену. Це означає, що змінній v , відповідно до даної мітки, присвоюється значення z .

Обмеження може містити змінні. Для обмеження c його множину змінних позначимо як V_c . Обмеження називається *бінарним*, якщо його множина V_c містить тільки 2 елементи, і *унарним*, якщо тільки один. Усі інші обмеження називаються

небінарними. Задача задоволення обмежень розглядається як *бінарна* якщо, якщо всі її обмеження є бінарними або унарними. За класичним підходом, будь-яку задачу задоволення обмежень можна перетворити в бінарну. Це полегшить її реалізацію засобами комп'ютерної техніки, проте також зумовить збільшення кількості змінних із великими доменами, що зробить початкову задачу більш складною.

Для графічного представлення структури задачі задоволення обмежень використовується *граф обмежень*. При бінарній *CSP* вершини такого графа відповідатимуть змінним, які будуть з'єднуватися дугами тоді, якщо існує відповідне обмеження. *Простір розв'язків* задачі задоволення обмежень утворює множина присвоєнь. *Присвоєнням* вважають відображення ψ множини доменних змінних $V_1 \subseteq V$ на відповідні домени таким чином, що $\psi(v_i) \in D_i$ для $v_i \in V_1$. Присвоєння можна розглядати як множину міток, тобто як спосіб визначення значень змінних. Повним вважають присвоєння, заданим на всій множині доменних змінних; в іншому випадку присвоєння розглядається як часткове. Простір рішень пошукової задачі на основі обмежень розглядають як множину всіх можливих присвоєнь $\Psi_V \equiv D_1 \times \dots \times D_n$.

Обмеження c вважається *задоволеним* у певному присвоєнні ($\psi \in c$), якщо всі змінні з множини V_c даного обмеження в цьому присвоєнні мають такі значення, що відповідний кортеж значень змінних належить c . В іншому випадку обмеження вважається *незадоволеним*.

Для кожного обмеження $c_i(v_{i_1}, \dots, v_{i_m})$, існує зворотне обмеження $\neg c_i(v_{i_1}, \dots, v_{i_m})$, яке не задовольняється, якщо задовольняється c . Кожне часткове присвоєння ψ вважається *сумісним*, якщо всі обмеження, що стосуються змінних, визначених у ψ , задовольняються. Рішенням задачі задоволення обмежень вважають повне сумісне присвоєння, що задовольняє всі обмеження з множини C .

Задача на основі *CSP* вважається *надобмеженою* або *несумісною*, якщо вона не має жодного розв'язку. Задача задоволення обмежень вважається *недодобмеженою*, якщо вона має багато розв'язків. Обмеження з доданими

елементами у відношення вважається *послабленим*, а обмеження з видаленими елементами з відношення – *посиленим*.

Структура оцінювання обмежень технологічних проблем визначається кортежем $(Ev^{set}, \otimes_{TP}, \succ_c, T, \perp)$, де Ev^{set} - множина, елементи якої називаються оцінками; \otimes_{TP} - комутативна, асоціативна бінарна операція над Ev^{set} , яка задовольняє властивостям ідентичності та монотонності; \succ_c - відношення загального порядку над Ev^{set} ; T і \perp - максимальний і мінімальний елемент із Ev^{set} , що задається відношенням \succ_c .

Введення оцінювань для змінних поряд із оцінюванням обмежень дозволяє вводити множини преференцій і досліджувати їх вплив на процес задоволення та порушення обмежень в ході вирішення технологічних проблем. Крім того, введення вагових значень змінних дозволяє будувати часткові присвоєння з введеними мінімаксними функціями для сумарних вагових значень задоволених і порушених обмежень. Це, відповідно, дозволить сформулювати формальну структуру рішення з можливими помилками, які будуть оцінюватись системою.

Для формального опису процесу задоволення обмежень з преференціями при виникненні технологічних проблем введемо представлення деякої формальної метаструктури. В багатьох підходах в якості ініціалізуючої метаструктури використовується ідея напівкільця. Вибір конкретного виду ініціалізації залежить від складності технологічної проблеми та вибраного способу структуризації множини (системи, ієрархії) обмежень. У загальному випадку, введення формальної метаструктури є необхідним для вибору певного способу специфікації значень, які повинні асоціюватися з кортежами значень у доменах змінних при виконанні присвоєнь. Крім того, для вибраної метаструктури слід ввести формальні операції, що дозволятимуть оперувати з обмеженнями.

Як формальну мета структуру розглядається кортеж $(MS, \oplus, \otimes, 0, 1)$, де MS - множина метаструктури; $0, 1 \in MS$; \oplus, \otimes - введені операції з властивостями комутативності, асоціативності та дистрибутивності відповідно. Системою

обмежень вважається кортеж $CS = (MS, D, V)$, де MS – формальна метаструктура, D – домен змінних, V – множина змінних.

Для заданої формальної метаструктури $FMS = (MS, \oplus, \otimes, 0, 1)$ і системи обмежень $CS^{MS} = (MS, D, V)$ в якості обмеження розглядається пара (RC, SC) , де $SC \subseteq V$ і $RC : D^{SC} \rightarrow MS$.

Інтерпретація кортежу (RC, SC) відповідає парам значень $\langle \{умова_релевантності_i\}_i, \{умова_задоволення_i\}_i \rangle$. Обмеження визначатиме множини змінних у sc і присвоюватиме кожному кортежу значень елементи з формальної метаструктури.

Технологічною проблемою на основі м'яких обмежень TP^{soft} вважається пара $(ConstrSet, SC)$ над системою обмежень $CS^{MS} = (MS, D, V)$, де $SC \subseteq V$, а $ConstrSet$ – множина обмежень для TP^{soft} . Позначатимемо результат проєкції значень кортежу κ визначених на множині V^1 на значення кортежу над змінними в множині V^2 через $\kappa \downarrow_{V^2}^{V^1}$.

Наприклад, для заданих обмежень $c_1 = (rc_1, sc_1)$ і $c_2 = (rc_2, sc_2)$ їх сполученням $c_1 \otimes c_2$ буде обмеження (rc^1, sc^1) , в якому множина sc^1 представляє об'єднання множин sc_1 і sc_2 : $sc^1 = sc_1 \cup sc_2$, а rc^1 буде визначатися таким чином:

$$rc^1(\kappa) = rc_1(\kappa \downarrow_{sc_1}^{sc^1}) \times rc_2(\kappa \downarrow_{sc_2}^{sc^1}).$$

Для множини обмежень $ConstrSet = \{c_1, \dots, c_n\}$ введемо скорочення $\bigotimes_{i=1}^n c_i = \bigotimes ConstrSet$ для $c_1 \otimes \dots \otimes c_n$.

Згідно введеного означення сполучення двох обмежень означає побудову нового обмеження. Це включає всі змінні початкових обмежень із прив'язкою до кожного кортежу значень над такими змінними певного елемента метаструктури MS , що отримується шляхом множення елементів, прив'язаних початковими обмеженнями відповідних субкортежів.

Для заданої множини обмежень $ConstrSet = (RC, SC)$ і підмножини $V_1 \subseteq V$, проекцією $ConstrSet$ на V_1 (позначатимемо як $ConstrSet \downarrow V_1$), будемо вважати обмеження (RC', SC') , де $SC' = SC \cap V_1$, і

$$RC'(\kappa') = \sum_{\kappa / [\kappa \downarrow_{SC'}^{SC}] = \kappa'} RC(\kappa).$$

У загальному випадку, операція проекції полягає в виключенні певних змінних, наприклад, при послабленні надобмеженої проблеми TP^{over} . Це виконується шляхом прив'язки до кожного кортежу над множиною залишкових змінних елемента метаструктури, який є сумою елементів, прив'язаних початковим обмеженням до всіх розширень цього кортежу над множиною видалених змінних.

2.4 Структура релевантності за умов задоволення технологічних обмежень

Розглянемо два обмеження виду $c_1 = (rc, sc_1)$ і $c_2 = (rc, sc_2)$ над (MS, D, V) . Введемо впорядкування обмежень \sqsubseteq_{MS} на основі відношення часткового порядку. Будемо вважати, що $c_1 \sqsubseteq_{MS} c_2$, якщо $rc_1(\kappa) \leq_{MS} rc_2(\kappa)$ має місце для всіх кортежів κ , $\kappa \in D$. Також, якщо $c_1 \sqsubseteq_{MS} c_2$ та $c_1 \supseteq_{MS} c_2$, тоді $c_1 = c_2$.

При існуванні двох технологічних проблем на основі м'яких обмежень $TP_1^{soft} = (ConstrSet_1, SC)$ і $TP_2^{soft} = (ConstrSet_2, SC)$, відповідно, введемо відношення перепорядкування технологічних проблем $\sqsubseteq_{TP^{soft}}$ таким чином, що $Sol(TP_1^{soft}) \sqsubseteq_{MS} Sol(TP_2^{soft})$, якщо $TP_1^{soft} \sqsubseteq_{TP^{soft}} TP_2^{soft}$. Якщо $TP_1^{soft} \sqsubseteq_{TP^{soft}} TP_2^{soft}$ і $TP_2^{soft} \sqsubseteq_{TP^{soft}} TP_1^{soft}$, тоді вони мають однакову множину рішень Sol^{Set} . В такому випадку, можемо стверджувати, що TP_1^{soft} і TP_2^{soft} є еквівалентними проблемами і позначати як $TP_1^{soft} \equiv TP_2^{soft}$.

Розглянемо дві системи обмежень $CS^{MS} = (MS, D, V)$ і $CS^{MS'} = (MS', D, V)$, а також технологічні проблеми на основі м'яких обмежень $TP^{soft} = (ConstrSet, SC)$ і

$TP'^{soft} = (ConstrSet', SC)$, що їм відповідають, і мають, відповідно, рішення $Sol(TP^{soft}) = (RC, SC)$ і $Sol(TP'^{soft}) = (RC', SC)$. Технологічну проблему на основі м'яких обмежень TP^{soft} будемо вважати *уточненням* технологічної проблеми TP'^{soft} , якщо для довільної пари кортежів $\kappa^1 = (\kappa_1^1, \dots, \kappa_{|sc|}^1)$, $\kappa^2 = (\kappa_1^2, \dots, \kappa_{|sc|}^2)$, де $\kappa_i^1, \kappa_i^2 \in D$ так, що наявність твердження $RC'(\kappa^1) <_{MS'} RC'(\kappa^2)$ означатиме наявність відношення повного впорядкування $RC(\kappa^1) <_{MS} RC(\kappa^2)$.

Припустимо, що технологічна проблема на основі м'яких обмежень TP^{soft} є *уточненням* TP'^{soft} , і розглянемо також пару кортежів $\kappa^1 = (\kappa_1^1, \dots, \kappa_{|sc|}^1)$ і $\kappa^2 = (\kappa_1^2, \dots, \kappa_{|sc|}^2)$, де $\kappa_i^1, \kappa_i^2 \in D$. Тоді, якщо $RC(\kappa^1) \diamond_{MS} RC(\kappa^2)$, то матиме також місце $RC'(\kappa^1) \diamond_{MS'} RC'(\kappa^2)$

Дві технологічні проблеми на основі м'яких обмежень TP^{soft} і TP'^{soft} будемо вважати *сильно еквівалентними*, якщо кожен з них можна розглядати *сильним уточненням* іншої. Еквівалентні технологічні проблеми на основі м'яких обмежень визначають одне й те ж впорядкування над кортежами значень у множині V і мають однакові відповідні множини оптимальних кортежів, а задача знаходження оптимальних кортежів буде еквівалентною для обох формулювань. Також для двох еквівалентних технологічних проблем на основі м'яких обмежень не вимагається наявність однакових значень у кортежах, а лише однаковий спосіб їх впорядкування. З попередніх тверджень випливає: якщо в одній з еквівалентних проблем існують не порівнювані елементи, то вони також повинні бути і в зв'язаній проблемі.

Щоб мати змогу порівняти класи технологічних проблем, які базуються на різних формальних метаструктурах, введемо поняття *уточнення* для метаструктур.

Для двох заданих метаструктур - MS і MS' - *уточненням* із MS у MS' будемо вважати функцію σ – таку, що трансформує кожен технологічну проблему на основі м'яких обмежень $TP'^{soft} = (ConstrSet', SC)$ з множиною обмежень $ConstrSet' = \bigcup_i (rc'_i, sc_i)$ над системою обмежень (MS', D, SC) у технологічну

проблему $TP^{soft} = (ConstrSet, SC)$, де $ConstrSet = \bigcup_i (rc_i, sc_i)$ над множиною (MS, D, SC) (де $rc_i = \sigma \circ rc'_i$ є істинним для всіх i таких, що TP^{soft} є уточненням TP'^{soft}).

Якщо існує уточнення з MS' у MS , тоді кожна технологічна проблема на основі м'яких обмежень TP'^{soft} над MS' може бути вирішена шляхом застосування уточнення до TP^{soft} і подальшого вирішення результуючої проблеми над MS . Тобто проблеми, визначені над MS' , не є складнішими за проблеми, визначені над MS .

У загальному випадку, через преференції кортежів можна виражати преференції обмежень. У даному випадку, найкраще з можливих значень преференцій присвоюється всім кортежам, що задовольняють задане обмеження. Решту кортежів отримують значення преференції обмеження, що є доповненням до початкового.

Протилежне перетворення дозволяє замінити обмеження $D_1 \times \dots \times D_k \rightarrow PD^{set}$, відповідно, множиною обмежень c_1, \dots, c_k , де k - потужність рангової множини для c , тобто:

$$k = \text{card} \left\{ pd \in PD^{set} \mid \exists \psi \in D_1 \times \dots \times D_k : c\psi = pd \right\}.$$

Однак це перетворення може бути використане тільки тоді, якщо k - скінченне для всіх обмежень. Достатньою умовою отримання скінченного значення k розгляд тільки скінчених доменів змінних. У такий спосіб ми отримаємо скінченну кількість присвоєнь (тобто випадок, коли верхня границя для k відповідатиме кількості присвоєнь).

Виконаємо аналіз можливих способів побудови алгоритмів для рішення ієрархії обмежень технологічних проблем. Послідовність $SQ = \langle c_1, \dots, c_{k_{max}} \rangle$ будемо вважати ієрархічним впорядкуванням для ієрархії $ConstrHrch$, з k_{max} обмеженнями, якщо всі обмеження для SQ відсортовані за рівнями ієрархії ($c_{i_1} \in ConstrSyst_{k_1}$, $c_{i_2} \in ConstrSyst_{k_2}$, $k_1 < k_2 \Rightarrow i_1 < i_2$), і відповідно впорядкування $\leq_{W^{set}}$ (для

$c_{i_1}, c_{i_2} \in ConstrSyst_{k_1}$ таких, що $cw(c_{i_1}) <_{W^{set}} cw(c_{i_2}) \Rightarrow i_1 < i_2$). Послідовність $\langle c_1, c_2, \dots, c_i \rangle$ позначається як SQ_i для $i \leq k_{max}$, $SQ_i \subset SQ$.

Нехай $SQ = \langle c_1, \dots, c_{k_{max}} \rangle$ – ієрархічне впорядкування для ієрархії $ConstrHrch$.

Рекурсивно визначену множину $OS = OS_{k_{max}}$ вважатимемо множиною впорядкування розв'язків для ієрархічного впорядкування SQ :

$$OS_1 = \{\psi^1 \mid \psi^1 \text{ є присвоєнням для } SQ\},$$

$$OS_i = \{\psi^1 \mid \psi^1 \in OS_{i-1} \wedge SF(c_i \psi^1) = \max_{os \in OS_{i-1}} SF(c_i os)\} \text{ для } i \in 2 \dots k_{max}.$$

Якщо OS – множина впорядкування розв'язків для SQ , тоді OS є множиною “краще впорядкованих” рішень. Відповідно, процедуру пошуку рішення для ієрархії обмежень (далі ІО) з компаратором “краще впорядкований” можна розділити на два кроки:

1. Сортування обмежень на кожному рівні $ConstrSyst_i$ для ієрархії $\langle OS_1, OS_2, \dots, OS_{k_{max}} \rangle$ із використанням вагових коефіцієнтів, що задаються функцією cw та їхнім впорядкуванням $\leq_{W^{set}}$ для отримання послідовності обмежень OS_i ;

2. Застосування впорядкування вхідних обмежень згідно послідовності $\langle OS_1, OS_2, \dots, OS_{k_{max}} \rangle$.

Введення глобальних компараторів дозволяє класифікувати технологічні проблеми згідно оціночних значень обмежень (вагових коефіцієнтів) і виконувати ініціалізацію компараторів «кращий для сум вагових коефіцієнтів», «кращий за кількістю незадоволених» та «лексикографічно кращий».

При компараторі «кращий для сум вагових коефіцієнтів» розглянемо обмеження з міткою $c:l$ з ІО $ConstrHrch$ із ваговим коефіцієнтом $cw(c)$, визначене над множиною змінних (v_1, \dots, v_k) . Тоді c відповідатиме м'якому обмеженню (rc, sc) – такому, що $sc = (v_1, \dots, v_k)$ і:

$$\forall \psi \equiv [v_1 = c_1^2, \dots, v_k = c_k^2] : rc(c_1^2, \dots, c_k^2) = (\dots, cw(c^1) \times SF(c^1 \psi), \dots)$$

Компаратор «кращий за кількістю незадоволених» можна розглядати як частковий випадок компаратора «кращий за сумами вагових коефіцієнтів», у якому вагові коефіцієнти всіх обмежень рівні 1. Відповідний клас технологічних проблем на основі м'яких обмежень з ваговими коефіцієнтами спрощує початкове формулювання м'якого обмеження до виду $rc(c_1^2, \dots, c_k^2) = (\dots, SF(c^l \psi), \dots)$, де $SF(c^l \psi) \in \{0, 1\}$ на l -тій позиції для $c : l$.

Вихідною точкою для побудови класу технологічних проблем з м'якими обмеженнями для компаратора «кращий у найгіршому випадку» є виконання підстановки для випадку мультиплікативної операції: для кожного рівня ІО виконується пошук найменш задоволеного обмеження.

Розглянемо класичний випадок побудови ієрархії обмежень із трьох обмежень. $ConstrHrch = ConstrHrch_1 \cup ConstrHrch_2 = \{c_1, c_2\} \cup \{c_3\}$ для якого справедлива нерівність $cw(c_1)SF(c_1 \psi) \leq cw(c_2)SF(c_2 \psi)$, що існує для кожного присвоєння ψ . Відповідний клас технологічних проблем на основі м'яких обмежень означимо як $(ConstrSet, SC) = (\{(rc_i, sc_i) | c_i(rc_i, sc_i), i = 1 \dots 3\}, sc_1 \cup sc_2 \cup sc_3)$. Такий клас технологічних проблем на основі м'яких обмежень повинен задовольняти нерівність щодо c_1 і c_2 . Оскільки обидва обмеження належать до того ж рівня ієрархії CH і c_2 є більш важливим ніж c_1 на основі припущення

$$rc_1(\kappa \downarrow_{sc_1}^{sc}) \leq_{OS} rc_2(\kappa \downarrow_{sc_2}^{sc}) \quad (2.1)$$

Обмеження c_3 належить до нижчого рівня ієрархії, ніж c_1 . Тому справедливо, що $\forall \kappa : rc_3(\kappa \downarrow_{sc_3}^{sc}) \leq_{OS} rc_1(\kappa \downarrow_{sc_1}^{sc})$. Об'єднаємо обидві сторони нерівності із c_2 , і з монотонності для \times над \leq_{OS} одержимо:

$$rc_3(\kappa \downarrow_{sc_3}^{sc}) \times rc_2(\kappa \downarrow_{sc_2}^{sc}) \leq_{OS} rc_1(\kappa \downarrow_{sc_1}^{sc}) \times rc_2(\kappa \downarrow_{sc_2}^{sc}) \quad (2.2)$$

Обмеження c_1 і c_2 знаходиться на одному рівні ієрархії, а c_2 має більший ваговий коефіцієнт, ніж c_1 , відповідно до виразу (2.1). Оскільки компаратор «кращий у найгіршому випадку», згідно означення, повинен мати найгіршу

поведінку на кожному рівні, то рівність $rc_1(\kappa \downarrow_{sc_1}^{sc}) \times rc_2(\kappa \downarrow_{sc_2}^{sc}) =_S rc_2(\kappa \downarrow_{sc_2}^{sc})$ існуватиме для всіх кортежів κ , означених над змінними в sc відносно рівності (2.1)

Тому, в кінцевому підсумку на основі (2.2) отримаємо, що:

$$rc_3(\kappa \downarrow_{sc_3}^{sc}) \times rc_2(\kappa \downarrow_{sc_2}^{sc}) \leq_S rc_2(\kappa \downarrow_{sc_2}^{sc})$$

Компаратор “лексикографічно кращий” розглядається як покращена версія компаратора “кращий у найгіршому випадку”. З точки зору ієрархії обмежень, відповідне м’яке обмеження відрізняється означенням умов релевантності RC функції при компараторі “кращий за ваговими сумами”:

$$\forall \psi \equiv [v_1 = d_1, \dots, v_k = d_k] : rc(d_1, \dots, d_k) = (\dots, \{cw(c) \times SF(c\psi)\}, \dots),$$

де $\{cw(c) \times SF(c\psi)\}$ – мультимножина.

Випадок “локального” компаратора є складнішим, оскільки він порівнює успішність кожного обмеження окремо і не виконує сумування за рівнями. Це відображається мультимножинами, заданими над непорівнюваними елементами. Кожна така мультимножина відповідає певному обмеженню. Основна ідея включення мультимножин замість звичайних множин у класах технологічних задач із м’якими обмеженнями визначається властивостями введених операцій.

Для випадку компаратора «локально кращий» розглянемо ІО $ConstrHrch = \bigcup_{i=1}^{k_{max}} \{c_1, \dots, c_{n_i}\}$ та її обмеження з мітками $c_i : l_i$ над множиною змінних (v_1, \dots, v_k) . Тоді c_i відповідає м’якому обмеженню (rc_i, sc_i) , для якого $sc_i = (v_1, \dots, v_k)$, а

$$\forall \psi \equiv [v_1 = d_1, \dots, v_k = d_k] : rc_i(d_1, \dots, d_k) = \begin{cases} (\dots, [\langle \rangle c_i^l]_{i=1..k_{max}}^{l=1..n_i}, \dots) & \text{yêù î } \psi \ddot{E} -c_i | \\ \emptyset & \text{yêù î } \psi \ddot{E} c_i | \end{cases}$$

Мультиплікативна операція відповідає об’єднанню мультимножин, розширених до кортежів.

Для компаратора «локально метрично кращий» розглядається значення функції успішності та виконується порівняння успішності в окремих обмеженнях.

Зокрема, у випадку м'якого обмеження отримаємо, що:

$$rc_i(d_1, \dots, d_k) = (\dots, \{SF(c_i, \psi) [\langle \rangle c_i^i]_{i=1..k_{max}}^{l=1..n_i} \}, \dots) \text{ при } \psi \in \mathbb{R}^{-c_i}.$$

Клас технологічних проблем на основі м'яких обмежень для компаратора «краще впорядкований» можна означити шляхом уточнення початкової ієрархії. Зокрема, для кожної ІО $ConstrHrch$ із ваговою функцією cw можна виконати побудову нової ІО шляхом уточнення початкової ієрархії $ConstrHrch / cw$. Клас технологічних проблем на основі м'яких обмежень для уточнення ієрархії $ConstrHrch / cw$ із «локально кращим» компаратором відповідатиме класу технологічних проблем на основі м'яких обмежень початкової ієрархії обмежень $ConstrHrch$ із компаратором «краще впорядкований».

Ми розглядаємо формальну структуру технологічних задач на основі обмежень у контексті пошукових задач на основі обмежень із оцінками та пошукових задач на основі м'яких обмежень. Важлива особливість формальної структури пошукових задач на основі м'яких обмежень [129-131] полягає в можливості представлення відношення часткового порядку, що є доцільним з точки зору мультикритеріальної оптимізації. Пошукові задачі на основі обмежень із оцінками дозволяють впорядковувати множини оціночних значень.

Оскільки домен змінних у пошуковій задачі на основі м'яких обмежень є скінченним, то перетворення обмежень між пошуковими задачами на основі м'яких обмежень і пошуковими задачами на основі обмежень із оцінками є здійсненим. Припущення щодо існування загального порядку надає обидвом формальним структурам однакового рівня експресивності (виразності). В попередньому викладі ми розглядали питання відповідності між кортежем преференції та преференціями обмежень. Структура оцінювання визначається або на основі адитивної операції, або на основі операції пошуку мінімального значення, що застосовується для порівняння оцінювань у різних присвоєннях.

Формальна структура часткового задоволення [132-134] обмежень визначає простір послаблених пошукових проблем разом з їх метриками. Задача часткового задоволення обмежень відповідає випадку пошукової задачі на основі обмежень із

оцінками, в якій визначена структура послаблення разом із введеною функцією вимірювання метричної відстані.

Базові види формальних структур, такі як: пошукові задачі на основі обмежень з ваговими, ймовірнісними, можливісними коефіцієнтами, а також пошукові задачі з нечіткими та лексикографічними обмеженнями – розглядаються як окремі класи пошукових задач на основі обмежень із оцінками і пошукових задач з м'якими обмеженнями.

Введення ієрархій (системи) обмежень вимагає побудови механізму оперування з преференціями. Реалізація такого механізму дозволить оперувати з обмеженнями шляхом визначення його як обов'язкового (жорсткого) та преференційного (м'якого).

При вирішенні технологічної проблеми автоматизована інтелектуальна система виконуватиме оцінювання пропонованих рішень, найперше, з точки зору задоволення або порушення обов'язкових обмежень. Задоволення або порушення преференційних обмежень буде перевірятися після завершення відповідної процедури перевірки для обов'язкових обмежень.

Використання компараторів дозволяє визначити спосіб побудови множини рішень для ієрархії обмежень. Компаратори класифікують як локальні, глобальні та доменні [135-137]. Для локальних компараторів кожне обмеження розглядається індивідуально. Глобальні компаратори виконують сумування помилок всіх обмежень певного рівня. Доменні компаратори розглядають кожне обмеження індивідуально, але на відміну від локальних компараторів, два рішення, не порівнювані на більш високих рівнях, можуть порівнюватися на більш низьких рівнях.

Як зазначалось вище, процес побудови рішення технологічної проблеми в інтелектуальній системі з процедурної точки зору можна, відповідно, розглядати як процес роботи з присвоєннями, одні з яких оцінюються локально, а інші – глобально. Присвоєння ψ^1 будемо вважати *локально кращим* за присвоєння ψ^2 , якщо ступінь задоволення обмежень для присвоєння ψ^1 є вищим за ступінь

задоволення обмежень присвоєння ψ^2 на рівнях $1 \dots k-1$, а на рівні k – ступінь задоволення присвоєння ψ^1 дорівнює ступеню задоволення присвоєння ψ^2 , але існує хоча б одне обмеження, ступінь задоволення якого вищий у присвоєнні ψ^1 :

$$\forall c \in \bigcup_{i=1}^{k-1} \text{ConstrSyst}_i \ddot{\text{E}} \left[SD(c\psi^1) > SD(c\psi^2) \right]$$

$$\forall c' \in \text{ConstrSyst}_k \ddot{\text{E}} \left[SD(c'\psi^1) = SD(c'\psi^2) \right] \wedge$$

$$\left[\exists c'' \in \text{ConstrSyst}_k \vdash \left[SD(c''\psi^1) > SD(c''\psi^2) \right] \right]$$

2.5 Метричні характеристики введених формальних конструкцій

«Локально предикатно кращим» компаратором для множин, систем й обмежень технологічних проблем будемо вважати *локально кращий* компаратор, що використовує функцію успішності.

«Локально метрично кращим» компаратором будемо вважати *локально кращий* компаратор, що використовує доменну метрику для обчислення успішності задоволення обмежень їх множин, систем та ієрархій.

Для отримання можливості порівнювати формальні означення для оцінки рівня задоволення обмежень введемо функцію успішності для обмежень, їх множин, систем та ієрархій. Оскільки згідно початкового припущення, що задоволення та порушення позначається граничними значеннями, то доцільно здійснювати оцінку успішності для рівня множин, систем та ієрархій обмежень.

Для заданої множини обмежень $\text{ConstrSet} = \{c_1, c_2, \dots, c_{n_k}\}$ введемо функцію успішності множини (системи) обмежень для локально кращого компаратора:

$$SF(\text{ConstrSyst}\psi) = SF(\text{ConstrSet}_i\psi) = \sum_{i=1}^{n_k} [1 | \psi \in c_i].$$

Відповідно, успішність ієрархії обмежень визначатиметься як

$$SF(\text{ConstrHrch}\psi) = \sum_{i=1}^{k_{\max}} SF(\text{ConstrSyst}_i\psi)$$

Рішенням ієрархії обмежень технологічної проблеми будемо вважати присвоєння з максимальним значенням функції успішності:

$$Sol(ConstrHrch) = \psi \mid \max_{SF} SF(ConstrHrch \psi).$$

Введення доменно кращого компаратора дозволить, відповідно, розширити концепцію локально кращого компаратора, для введення можливості порівняння присвоєнь, які неможливо порівняти в рамках концепції локально кращих компараторів для технологічних проблем з накладеними обмеженнями.

В процесі побудови рішення технологічної проблеми присвоєння ψ^1 будемо вважати доменно кращим порівняно з іншим присвоєнням ψ^2 , якщо для кожного рівня від 1 до $k-1$, оцінки рівнів є не порівнюваними $SF(ConstrSyst_{i_1} \psi) \Leftrightarrow SF(ConstrSyst_{i_2} \psi)_{i_1 \neq i_2, i_1, i_2 = 1 \dots k-1}$, а на рівні k значення функції успішності є строго більшим принаймні для одного обмеження і більшими або рівними для решти обмежень:

$$\exists c, c' \in ConstrSyst_k \mid [SF(c) > SF(c')] \wedge [SF(c) > SF(c'')], \forall c'' \in ConstrSyst_k, c'' \neq c'$$

«Доменно метрично кращі» і «доменно предикатно кращі» компаратори для технологічних проблем можна отримати в результаті застосування відповідних функцій успішності для «доменно кращих» компараторів. Введемо спосіб побудови «глобально кращого» компаратора для технологічної проблеми на множині «глобальних» компараторів. Зведемо таку послідовність побудови до функції успішності відповідних рівнів.

Для технологічної проблеми присвоєння ψ^1 будемо вважати «глобально кращим», ніж деяке присвоєння ψ^2 , якщо для кожного рівня від 1 до $k-1$ сумаризація успішності обмежень після застосування присвоєння ψ^1 є рівною результату застосування присвоєння ψ^2 $SF\left(\bigcup_{i=1}^{k-1} ConstrSyst_i \psi_1\right) = SF\left(\bigcup_{i=1}^{k-1} ConstrSyst_i \psi_2\right)$, а на рівні k є строго більшою: $SF(ConstrSyst_k \psi_1) > SF(ConstrSyst_k \psi_2)$.

Подібним чином можна означити ступінь задоволення всіх глобальних компараторів. Для технологічної проблеми, функцією успішності ієрархії обмежень для «глобального компаратора» та присвоєння $\psi \in \Psi_V$ будемо вважати:

$$SF(\text{ConstrHrch}\psi) = SF\left(\bigcup_{i=1}^{k_{\max}} \text{ConstrSyst}_i\psi\right) = SF(c_1\psi), \dots, SF(c_{n_k}\psi)$$

Ефективним методом застосування глобальних компараторів є введення вагових коефіцієнти для обмежень, що виражають, яким чином обмеження відповідного рівня можуть порівнюватися.

Ваговий коефіцієнт обмеження ієрархії обмежень технологічної проблеми визначається на основі функції $cw: \text{ConstrHrch} \rightarrow W^{set}$, де W^{set} – впорядкована множина вагових коефіцієнтів. Введемо для множини W^{set} впорядкування $\leq_{W^{set}}$, тоді нерівність $cw(c^1) \leq_{W^{set}} cw(c^2)$ можна інтерпретувати як представлення переважності вагових коефіцієнтів. Для обмежень, що накладаються на технологічні проблеми, вагові коефіцієнти представляються натуральними або дійсними числами.

Після того, як вибрано вид глобального компаратора, можна задати вибір функції успішності для обмежень. Згідно означень класичних компараторів, перспективними для застосування при технологічних проблемах є такі види компараторів: «найкращий для вагових сум», «найкращий метричний», «компаратор обліку незадоволених обмежень» й інші. Проте, хоча використання компараторів є ефективним інструментом, його слід застосовувати з певною обережністю для запобігання необґрунтованому відсіканню обмежень у множинах, системах та ієрархіях, що, очевидно, знижуватиме якість інтелектуальних функцій системи та її адаптивність в цілому. Наприклад, часто виникає проблема, коли в системі встановлюється певний ваговий коефіцієнт обмеження, який визначає границю необхідного порушення чи задоволення. Тоді, відповідно, на даному рівні кожне обмеження, ваговий коефіцієнт якого виходить за інтервал необхідності задоволення чи порушення, буде ігноруватись. Окремо дослідимо перспективи побудови лексикографічного компаратора для технологічних проблем як застосування одного з видів глобального компараторів.

Твердження 2.1 Припустимо, що пошукова задача на основі обмежень з напівкільцем LP є уточненням LP' і розглянемо також пару кортежів $k^1 = (k_1^1, \dots, k_{|con|}^1)$

та $k^2 = (k_1^2, \dots, k_{|com|}^2)$ (де $k_i^1, k_i^2 \in D$) таких, що, якщо $def(k^1) \Leftrightarrow_s def(k^2)$, тоді матиме місце, $def'(k^1) \Leftrightarrow_s def'(k^2)$

Припустимо, що нерівність $def'(k^1) \Leftrightarrow_s def'(k^2)$ не має місця. Це, відповідно, означатиме, що $def'(k^1)$ і $def'(k^2)$ є впорядкованими. Наявність будь – якого впорядкування між цими напівкільцевими значеннями визначатиме відповідне впорядкування між $def(k^1)$ і $def(k^2)$, згідно означення 2.4. Але, оскільки ці елементи є не порівнюваними, то ми прийшли до суперечності, що доводить твердження.

Твердження 2.2 Кожне «краще впорядковане» рішення ψ для ієрархії C є локально кращим.

Припустимо, що «краще впорядковане» присвоєння ψ не є локально кращим. Тоді існує присвоєння ψ^1 , що є локально кращим, ніж ψ . Нехай C_k - перший рівень, де присвоєння ψ^1 і ψ мають різні значення функції оцінки помилок на деяких обмеженнях. Оскільки присвоєння ψ^1 є локально кращим, ніж присвоєння ψ , тоді:

$$\forall c^1 \in C_k : EF(c^1\psi^1) \leq EF(c^1\psi) \quad (2.3)$$

Рівень C_k буде першим рівнем, на якому будуть відрізнятися всі функції помилок. Тому наступне твердження слідує з того, що (ψ, ψ^1, C) є «краще впорядкованим»:

$$\exists c^1 \in C_k : EF(c^1\psi) < EF(c^1\psi^1) \quad (2.4)$$

Це суперечить формулі 3.4. Як результат ми отримаємо, що для жодного рішення ψ^1 не існує жодного рішення ψ , яке є «локально кращим», ніж ψ . Тому, відповідно, присвоєння ψ повинно бути «локально кращим» рішенням.

Проте існують «локально кращі» рішення, які не є «краще впорядкованими». Наприклад, розглянемо ієрархію $C = C_1 = \{c, c^1\}$, де $sw(c^1) <_w sw(c)$. Припустимо, що існують рішення ψ^1 і ψ такі, що $EF(c\psi^1) > EF(c\psi)$, $EF(c^1\psi^1) < EF(c\psi)$. Обидва рішення могли бути локально кращими, але тільки ψ може бути «краще впорядкованим», оскільки він є «краще впорядкованим» ніж ψ^1 .

Пізніше ми наведемо точні специфікації відношення між компаратора «локально кращим» і «краще впорядкований».

Твердження 2.3 Для заданої ієрархії C , вагової функції sw та присвоєння ψ^1 і ψ^2 наступне твердження матиме місце: «краще впорядкований» $(\psi^1, \psi^2, C) \Leftrightarrow$ «локально кращий» $\psi^1, \psi^2, C / sw$.

Нехай ψ^1 і ψ^2 - присвоєння з ієрархії C і нехай ψ^1 є краще впорядкованим присвоєнням ніж ψ^2 . Нехай k – перший за порядком рівень, де функції помилок на присвоєннях ψ^1 і ψ^2 - відрізняються, і нехай $c^1 \in C_k$ - є обмеженням з максимальним ваговим коефіцієнтом $sw(c^1)$ таким, що $EF(c^2\psi^1) \neq EF(c^1\psi^2)$. Припустимо, що $c \in C_{kl}$ має місце для деяких $l \in 1..n_k$ з ієрархії C / sw . Необхідно показати, що ψ^1 є локально кращим, ніж ψ^2 в ієрархії C / sw .

1. $EF(c^2\psi^1) = EF(c^2\psi^2)$ існує для кожного $c^2 \in C_{ij}$, $i \in 1..(k-1)$, $j \in 1..n_i$ оскільки, аналогічно існує для кожного $c^2 \in C_i$, $i \in 1..(k-1)$ (функції помилок для ψ^1 і ψ^2 уперше починають відрізнятись на рівні k).

2. $EF(c^2\psi^1) = EF(c^2\psi^2)$ має місце для кожного $c^2 \in C_{kj}$, $j \in 1..(l-1)$. Спершу зазначимо, що $c^2 \in C_k$ і що $sw(c^1) <_w sw(c^2)$ (відповідно до означення 3.14 і нерівності ($j < l$)). Обмеження c^1 має максимальний ваговий коефіцієнт в C_k такий, що функція помилок для ψ^1 і ψ^2 є різною. Як наслідок, $EF(c^2\psi^1) = EF(c^2\psi^2)$ для $c^2 \in C_{kl}$.

3. Існує також нерівність $EF(c^1\psi^1) < EF(c^1\psi^2)$, оскільки c є першим за рівнем і вагою обмеженнями з чіткими значеннями функцій помилок у ψ^1 і ψ^2 , а ψ^1 краще впорядковане, ніж ψ^2 .

4. $EF(c^2\psi^1) \leq EF(c^2\psi^2)$ існує для кожного $c^2 \in C_{kl}$ тому, що c^1 краще впорядковане, ніж c^2 ($c^2 \in C_k$, а $sw(c^1) =_w sw(c^2)$).

Продемонстровано, що вираз $EF(c^2\psi^1) = EF(c^2\psi^2)$ існує для $\forall c^2 \in C_{ij}, (i < k) \vee ((i = k) \wedge (j < l))$, а також, що існує $EF(c^1\psi^1) < EF(c^1\psi^2)$ та $\forall c^2 \in C_{kl} : EF(c_2\psi^1) \leq EF(c^2\psi^2)$. Це означає, що ψ^1 локально краще, ніж ψ^2 у C/sw .

Припустимо тепер, що ψ^1 локально краще, ніж ψ^2 у C/sw . Нехай функція помилки вперше відрізняється для $c^1 \in C_{kl}$. Отже, справедлива нерівність $EF(c^1\psi^1) < EF(c^1\psi^2)$. Твердження $\forall_i < k \forall c^2 \in C_i : EF(c^2\psi^1) = EF(c^2\psi^2)$ є наслідком означення компаратора «локально кращий» ($\forall_i \forall_j$ таких, що $(i < k) \vee ((i = k) \vee (j < l)) \forall c^2 \in C_{ij} : EF(c^2\psi) = EF(c^2\psi^2)$). Аналогічно наявний вираз $EF(c^2\psi^1) = EF(c^2\psi^2)$ для $\forall c^2 \in C_k$ таких, що $sw(c^1) <_w sw(c^2)$.

Твердження $EF(c^2\psi^1) \leq EF(c^2\psi^2)$ справедливе для $\forall c^2 \in C_k : sw(c^2) =_w sw(c^1)$ тому, що $c^2 \in C_{kl}$, а функції помилок на ψ^1 і ψ^2 вперше відрізняються на рівні C_{kl} і ψ локально краще, ніж ψ у C/sw . Отже, всі необхідні умови задоволені, і ψ є краще впорядкованим, ніж ψ у C .

Твердження 2.4 Присвоєння ψ є краще впорядкованим рішенням для ієрархії C з ваговою функцією sw тоді і тільки тоді, коли ψ є локально кращим рішенням для уточнення початкової ієрархії C/sw .

Це твердження дозволяє визначити помилку ієрархії $IO(V, D, C)$ для компаратора «краще впорядкованим» (і ступеня задоволення) як помилку $IO C/sw$ для компаратора «локально краще».

Твердження 2.5 Нехай SQ - ієрархічне впорядкування для C , а множина OS - множина впорядкування розв'язків для SQ . Тоді OS є множиною «краще впорядкованих» рішень.

Побудуємо доведення на основі індукції за кількістю обмежень m . Для $m = 1$ матимемо ієрархію $C = \{c_1\}$, і буде існувати тільки одна $SQ = \langle c_1 \rangle$. Отримаємо $OS = OS_1 = \{\psi^1 \mid \forall os : EF(c_1\psi^1) \leq EF(c_1os)\}$ і, відповідно, кожне присвоєння $\psi^1 \in OS$ є «краще впорядкованим» рішенням.

Припустимо тепер, що твердження має місце для ієрархії з m обмеженнями і, відповідно, розглянемо випадок $m+1$ обмеження. Припустимо, що $\psi^1 \in OS_{m+1}$. Покажемо, що для кожного присвоєння ψ^2 – або ψ^1 є краще впорядкованим, ніж ψ^2 для SQ_{m+1} , або ψ^2 не є краще впорядкованим для SQ_{m+1} (ψ^1 і ψ^2 є не порівнюваними для SQ_{m+1}).

1. $\psi^2 \notin OS_{m+1} \wedge \psi^2 \in OS_m$: Функція помилки для кожного $c_i (i \in 1..m)$ визначається унікально, що слідує з припущення $\psi^2 \notin OS_m$ та означення множини впорядкування рішень. Із припущення $\psi^2 \notin OS_{m+1}$ і мінімальності значення функції помилки для обмеження c_{m+1} випливає нерівність $EF(c_{m+1}\psi^1) < EF(c_{m+1}\psi^2)$. З двох даних властивостей слідує, що ψ^1 є “краще впорядковане”, ніж ψ^2 .

2. $\psi^2 \in OS_{m+1}$: Функція помилки для кожного обмеження є однаковою, отже не існує такого обмеження $c_i (i \in 1..m+1)$ такого, що $EF(c_i\psi^1) > EF(c_i\psi^2)$ (або $<$) і ні ψ^1 , ні ψ^2 не є краще впорядкованими, ніж інші присвоєння для SQ_{m+1} .

3. $\psi^2 \notin OS_m : \psi^1 \in OS_m$, отже ψ^1 не може бути краще впорядкованим, ніж ψ^1 для SQ_m , виходячи з індукційного припущення. Покажемо, що додавання обмежень c_{m+1} не змінює нічого для SQ_{m+1} . Значення функції помилки для деяких $i \in ..m$ відрізняється для ψ^1 і ψ^2 (з того, що $\psi^2 \notin OS_m$). Нехай i - перше таке число. Тоді справедливо, що $EF(c_i\psi^1) > EF(c_i\psi^2)$. Оскільки обмеження c_i міститься на найбільш важливому рівні і приймає найбільші значення вагового коефіцієнта, присвоєння ψ^2 не може бути краще впорядковане, ніж ψ^1 .

Таким чином, кожне присвоєння $\psi^1 \in OS$ є краще впорядкованим рішенням.

Проте існують краще впорядковані рішення, які не можуть бути отримані через жодне впорядкування ієрархії як її множина впорядкування рішень.

Твердження 2.6 Для заданої ациклічної множини обмежень алгоритм індиго виконує обчислення впорядкованого “метрично кращого” рішення.

Вхідні обмеження для алгоритму індиго визначають ієрархічне впорядкування SQ на основі OC_0, OC_1, \dots, OC_n . Алгоритм індиго мінімізує функцію помилки в

порядку, заданому ієрархічним впорядкуванням SQ . Це є також вимогами раніше введених тверджень, отже, отримуємо впорядковано «метрично краще» рішення, як результат застосування алгоритму.

ВИСНОВКИ ДО РОЗДІЛУ 2

1. Представлено розширення базового формально-логічного апарату інформаційно-пошукових задач на основі обмежень при побудові інтелектуальної інформаційної системи на основі обмежень шляхом розділення та контролю множин задоволених і порушених обмежень для підтримки прийняття рішень в умовах невизначеності при контролі технологічних процесів.

2. Представлено підхід до опису технологічних проблем на основі множини змінних (технологічних параметрів) із введеними доменами (довірчими інтервалами) і введеними множинами, системами й ієрархіями релевантних обмежень.

3. Отримане формальне розширення зберігає класичні означення для міток, розмірності, графової інтерпретації, простору розв'язків, присвоєнь.

4. Запропонована нова інтерпретація процесів задоволення та порушення обмежень, часткового задоволення обмежень у термінах контролю процесу рішення пошукових задач шляхом виділення множин помилок (незадоволених обмежень), що, відповідно, дозволяє виділяти хибні знання та дії оператора технологічного процесу. Відповідно, процес пошуку рішення розглядається у вигляді присвоєнь, а самі інформаційно-пошукові задачі (з точки зору кількості рішень) поділяються на недообмежені або надобмежені.

5. Представлене формулювання пошукової задачі як стратегії автоматизованої інтелектуальної системи щодо рівня знань оператора, що визначається кількістю технологічних помилок, може, відповідно, бути або посиленим, або послабленим.

6. Запропоновано інтерпретацію технологічних проблем на основі методу контролю множини порушених і задоволених обмежень, що дозволяє кожному

обмеженню присвоїти оціночне значення, яке задає загальну перевагу обмежень.

7. Введено впорядкування для множини оцінювань, що дозволяє описувати рівні порушення та задоволення обмежень, причому спосіб побудови структури оцінювання визначає залежність оціночних значень від множин порушених і задоволених обмежень на рівні множин, систем та ієрархій.

8. Визначено та формально описано очікувані класи технологічних проблем зі структурою оцінювання в формі $(0,1)$; у формі оцінювань із ваговими коефіцієнтами (шляхом обчислення сум вагових коефіцієнтів для множин, систем та ієрархій та, відповідно, визначення максимальних і мінімальних сум); у формі оцінювань із ймовірнісними коефіцієнтами, які, відповідно, визначають ймовірність релевантності обмеження фактичній технологічній проблемі, оскільки, згідно початкового припущення, система фіксує порушення та задоволення тільки релевантних обмежень.

9. Визначено спосіб оперування з ймовірнісними оцінками для рівня множини, системи й ієрархії обмежень шляхом обчислення сумарного значення ймовірнісних оцінок як рішення технологічної проблеми, так і вибору рішення – присвоєння з максимальним оціночним значенням ймовірнісної релевантності, оскільки саме таке рішення максимально відповідає очікуванням автоматизованої інтелектуальної системи.

10. Представлено формальну структуру оцінювання з можливісними коефіцієнтами, що дозволяє виконувати оцінювання можливої необхідності задоволення чи порушення множини (системи, ієрархії) накладених обмежень при побудові присвоєнь у процесі пошуку рішення технологічної проблеми. Пошук оптимального присвоєння з введеною функцією мінімізації по сумарним значенням переваг множини порушених і введеною функцією максимізації за сумарним значенням переваг множини задоволених обмежень розглядається як основна складова процесу пошуку рішень. Введення переваг у цьому випадку виконує ранжування множини релевантних обмежень за ступенем важливості.

11. Введено оцінювання для змінних, що дозволяє представляти множини преференцій і виражати їх вплив на процес задоволення та порушення обмежень під час вирішення технологічних проблем. Окрім того, введення вагових значень змінних дозволяє будувати часткові присвоєння з введеними мінімаксними функціями для сумарних вагових значень задоволених і порушених обмежень, що, відповідно, дозволяє формувати формальну структуру рішення з можливими помилками, які оцінюються системою.

12. Представлено інтерпретацію обмеження в формі правила: умова релевантності – умова задоволення.

13. Виконано оцінку можливих доменних компараторів для випадку технологічних проблем, що дозволяє сформувати формальну структуру побудови рішення для введеної ієрархії обмежень.

РОЗДІЛ 3

ПОБУДОВА КОНЦЕПЦІЇ АБДУКТИВНОГО ВИВЕДЕННЯ ЗНАНЬ НА ОСНОВІ МУЛЬТИМЕДІЙНИХ ВХОДЖЕНЬ

3.1. Сутність і види абдуктивних міркувань у відповідних фреймворках

У [138-147] і загальних концепціях філософії було вперше введено ідею абдукції, а саме виділяється три форми міркувань:

1) **дедукція** – вид аналітичного процесу, який базується на застосуванні загальних правил до деяких часткових випадків, що має своїм результатом процес інференції (логічного висновку);

2) **індукція** – синтетичні міркування, що дозволяють виводити певне правило з заданого випадку та результату;

3) **абдукція** – вид синтетичної (від синтез) інференції (логічного висновку) для випадку наявного правила та результату.

Абдукція є частковою формою міркувань як форми синтетичних (синтезуючих) міркувань, що виводять випадок із правила і результату. В найпростішій формі абдукція може бути описана правилом:

$$\frac{sc, rc \Rightarrow sc}{rc}$$

Тобто з факту, що rc імплікує sc , і спостереження sc , ми можемо вивести rc . Абдукція може розглядатися як апробаційне прийняття гіпотези як пояснення спостережуваних фактів відповідно до відомих законів. Очевидно, що це є слабким видом інференції, оскільки ми не можемо сказати, що віримо в істинність пояснення, а віримо тільки в те, що воно може бути істинним.

Таким чином, абдукція полягає в обчисленні певних пояснень для спостережень і є прикладом немонотонних міркувань, оскільки пояснення, що відповідає одному стану бази знань може ставати суперечливим при поступленні деякої нової інформації. Абдуктивні міркування характеризуються існуванням

множинних пояснень і вибір преференційного пояснення є важливою складовою даної проблеми.

Дослідження дедукції виконується в контексті логічного програмування [148-163]. Фреймворк абдуктивного логічного програмування означається у вигляді триплета $\langle CLP, Hyp^{Set}, C^{LF} \rangle$, що складається з логічної програми CLP , множини базових виводимих атомів Hyp^{Set} і множини логічних формул C^{LF} , які називаються обмеженнями цілісності. Атоми в Hyp^{Set} є можливими абдуктивними гіпотезами, які можуть розглядатися як припущення для пояснення заданих спостережень у контексті CLP за умови, що ці припущення є консистентними (несуперечливими) з обмеженням цілісності в C^{LF} . У багатьох випадках зручно визначати Hyp^{Set} як множину предикатних символів, що виводяться, з деяким планованим значенням таким, що кожен базовий атом, чий предикатний символ є в Hyp^{Set} можливою гіпотезою.

Для початково заданого абдуктивного фреймворку загальне визначення абдуктивних пояснень можна ввести таким чином.

Означення 3.1. Нехай $\langle CLP, Hyp^{Set}, C^{LF} \rangle$ є абдуктивним фреймворком і нехай $Case$ є ціллю $Goal$. Тоді під абдуктивним поясненням для $Case$ будемо розуміти множину $\Delta \subseteq Hyp^{Set}$ базових виводимих атомів, таких що:

- 1) $[CLP \cup \Delta \models Case]$
- 2) $CLP \cup \Delta \cup C^{LF}$ є консистентним (несуперечливим).

Означення 3.2. Нехай $\langle CLP, Hyp^{Set}, C^{LF} \rangle$ є абдуктивним фреймворком, Δ_{init} є множиною виводимих входжень і $Case$ є ціллю. Вважатимемо, що Δ є абдуктивним поясненням для $Case$ при заданому Δ_{init} , якщо $\Delta' \subset \Delta$ є абдуктивним поясненням для $Case$.

При цьому слід відмітити, що таке формулювання імплікує, що задана множина виводимих входжень Δ_0 повинна бути консистентною (несуперечливою) із заданими обмеженнями цілісності C^{LF} . У деяких випадках інтерес для дослідження представляє обчислення мінімальності абдуктивних пояснень.

Означення 3.3. Нехай $\langle CLP, Hyp^{Set}, C^{LF} \rangle$ є абдуктивним фреймворком, Δ_{mit} є множиною входжень, що виводяться, і $Case$ є ціллю. Вважатимемо, що $\Delta \in \Delta_{mit}$ – мінімальним поясненням для $Case$, якщо Δ є поясненням для $Case$ при заданому Δ_{mit} і для жодної власної підмножини Δ' для Δ ($\Delta' \subset \Delta$), Δ' є поясненням для $Case$ при заданому Δ_{mit} .

Введена ідея абдукції, згідно описаного представлення, має природнім чином побудований обчислювальний відповідник у межах власного розширення стандартної SLD-базованого обчислювального механізму логічного програмування. Зокрема, коли обмеження цілісності представляється у формі *заперечень*, тобто у формі $\neg(p_1, \dots, p_n)$, то процедура доведення в даному випадку може бути успішно адаптована для обчислення мінімальних абдуктивних пояснень. Як основний, слід виділити, що такі обчислювальні процедури утворені з двох етапів: 1) абдуктивна фаза; 2) фаза консистентності (несуперечливості). В абдуктивній фазі обчислювана процедура базово виконує кроки SLD-резолюції і виконує збір абдуктивних гіпотез, необхідних для пояснення заданих початкових цілей. У кожному випадку, коли виконується вибір нової гіпотези для виконання обчислень, відповідна процедура здійснює перевірку її консистентності (несуперечливості) відносно до релевантних обмежень цілісності (в даному випадку кожна така процедура припускає принаймні одне входження, що виводиться, в кожному обмеженні цілісності), що відкриває, відповідно, нову фазу консистентності (несуперечливості). У загальному випадку, легко відмітити, що фаза консистентності (несуперечливості) може вимагати, своєю чергою, побудови нової абдуктивної фази і т. д.

Основною ціллю пропонованого підходу є обґрунтування того факту, що технологія абдукції може бути ефективно використана для задач видобування даних. В існуючих дослідженнях цієї проблематики, як правило, дані напрями досліджуються незалежно (видобування даних і абдуктивні міркування як концепції штучного інтелекту).

Основна ідея полягає в побудові зв'язків між методами індукції як видом машинного навчання та методами абдукції [164-173]. Для побудови початкового

рішення необхідно розглянути можливості абдуктивного фреймворку загального призначення з точки зору побудови рішень для абдуктивних запитів і дослідження відомих систем, що використовують в своїх обчисленнях методи абдукції й індукції.

ACLP є системою, що поєднує абдуктивні міркування та методи задоволення обмежень шляхом інтеграції фреймворку абдуктивного логічного програмування (ALP) із фреймворком логічного програмування в обмеженнях (CLP). Відповідно, фреймворк ACLP, побудований у ході спроб адресації проблеми забезпечення рішень декларативного програмування високого рівня або середовища моделювання для проблем штучного інтелекту, які водночас мають прийнятний рівень обчислювальної ефективності.

В ACLP роль абдуктивних міркувань полягає в забезпеченні автоматичної редукції (звуження) представлень проблем високого рівня і цілей верхнього рівня відповідно до нижчого рівня обчислювальних задач для незалежних форм представлення загальної проблеми. Використання техніки логічного програмування в обмеженнях є фундаментальним рішенням для підсилення ефективності обчислювального процесу абдуктивного висновку так, як це застосовується на високому рівні представлення проблеми, що розглядається. Таким чином, цей підхід забезпечує зв'язок між властивостями домену проблем високого рівня і доменно-незалежними методами рішення.

Загальна обчислювальна модель для ACLP складається із взаємодії між гіпотезами та генерацією обмежень через абдуктивний висновок з перевіркою консистентності (несуперечливості) для припущень, що виводяться, і задоволення обмежень для відповідно згенерованих їх рівнів. Інтеграція абдуктивних міркувань з пошуком рішення для накладених обмежень у рамках ACLP є видом об'єднання: що вирішувач обмежень не тільки дозволяє вирішити кінцевий набір обмежень, згенерований при процедурі абдуктивної редукції, але також динамічно впливає на процес абдуктивного пошуку рішення. Це дозволяє виконати раннє підрізання абдуктивних редукцій шляхом установки нових відповідних CLP – обмежень на абдуктивні рішення, що проектуються.

На даному рівні система імплементується як відповідний мета-інтерпретатор,

що є надбудовою для відповідної CLP-мови. Наприклад, *ECLPS*, що використовує вирішувач обмежень CLP для управління обмеженнями над скінченними доменами. Результатуюча структура системи буде загального виду, і її реалізація може бути схожою до інших існуючих вирішувачів обмежень.

Фреймворк ACLP застосовується до різних типів проблем. Зокрема, початкові застосування були сконцентровані на проблемах планування, складання розкладів і загальних задач планування. Таким чином, фреймворк ACLP використовується також для імплементації систем, що можуть реалізовувати задачі розпізнавання за умов неповної інформації, вирішувати суперечності в задачах з накладеними вимогами, ідентифікувати причини суперечностей і проектувати оновлення, здатні відновлювати консистентність специфікацій та інтелектуалу інтеграцію інформації в цілому. Тут ACLP використовується як базовий фреймворк для розробки інформаційних зв'язків для семантичної інтеграції інформації над джерелами, утвореними з веб-сторінок. Незважаючи на те, що більшість з цих додатків не має широкого застосування (за винятком додатків, що здійснюють планування та побудову розкладів) цей клас рішень є корисним щодо індикації загальних методологічних направляючих розробки абдуктивних додатків. Цінність рішень даного класу (на прикладі додатків, що здійснюють планування), полягає в продукуванні експертних рішень високого рівня якості, які можна порівняти до рішень, «згенерованих вручну», експертами зі значним досвідом роботи з проблемами даного класу. Разом з тим, це забезпечує гнучку платформу, на якій можна експериментувати зі змінами і преференціями.

Незважаючи на велику кількість додатків в області абдукції і їх потенційних застосувань, загальний абдуктивний логічний фреймворк, як правило фокусується, на питаннях важливості представлення, тоді як в загальному випадку глибоке дослідження задач обчислювальної ефективності для проблем практичного розміру та комплексності не виконується (для теоретичного дослідження комплексності абдуктивних фреймворків).

Під А-системами розуміють Пролог-базований абдуктивний фреймворк, орієнтований на дослідження питань масштабованості. Він отримується, в

основному, злиттям ідей і концепцій, що слідують з попередніх відомих абдуктивних процедур.

На найбільш загальному рівні це дозволяє сформулювати правила висновку в формі переписуваних правил для станів. Відповідно, вихідні правила розширюють правилами, що можуть обробляти скінченні домени виразів для обмежень. На внутрішньому рівні для рівня імплементації **A**-системами використовують схожі схеми з вирішувачами обмежень для скінченних доменів. Тобто вони здійснюють оцінку всієї детермінованої інформації перед виконанням не детермінованих виборів. Відповідно, коли всі недетерміновані вибори є успішно оціненими і всі набори обмежень є консистентними, тоді рішення вважається знайденим.

Набори обмежень, що будуються в режимі виконання приймають активну участь в процесі пошуку. Це означає, що поточний стан набору обмежень може викликати нові виведення або запускати процедуру бектрекінгу. Такий рівень рішення досягається імплементацією для кожного обмеження домену, що є набором міркувань над множиною термів Herbrand(a) і скінченного домену виразів для обмежень, які дозволяють керувати моніторингом стану виразу відносно до поточного набору обмежень. Така програмна поведінка в **A**-системах приймає форму реіфікації (конкретизації), оскільки версія «Sicstus Prolog» (на відміну від інших «Prolog»-систем) використовує наперед визначений модуль для виводу на множині термів Herbrand(a). Відповідно, **A**-система використовує теж окремий вирішувач для термів Herbrand(a), що також підтримує процес реіфікації (конкретизації). Таким чином, **A**-системи дозволяють включати окремий шар на верхньому рівні вирішувача обмежень для скінченних доменів, що буде забезпечувати схожу функціональність із загальними виразами для скінченних доменів.

3.2 Інтерпретація ймовірнісної складової

Ймовірнісна абдукція Хорна є комбінацією формальної логіки і ймовірнісних представлень [174-175]. Загальна ідея полягає в побудові множини незалежних

виборів (choices) із логічною програмою, що визначає послідовність виборів (choices). Формальна логіка, що використовується в даному випадку базується на так званих «слабких» представленнях (тобто не містить диз'юнкцій), а всі значення, пов'язані з невизначеністю, вбудовуються в процедуру підтримки виборів користувача. При цьому також не накладається жодних імовірнісних залежностей на вибори виконані користувачем.

Такий підхід утворює простий фреймворк для абдукції на основі тверджень Хорна з імовірностями асоційованими з гіпотезами. Такий фреймворк включає припущення щодо бази правил і припущень щодо незалежності множини гіпотез. Тому можна показати, що кожен вид імовірнісних знань, що представляється в мережі дискретних Байєсових суджень можуть бути також представлені в даному фреймворку. Основний внесок, таким чином, полягатиме в пошуку залежностей між логічними й імовірнісними концепціями очевидних міркувань. Все це забезпечуватиме ефективну мову представлення, що забезпечуватиме компроміс між евристичною й епістемологічною адекватністю (тобто між інтуїтивним і теоретично обґрунтованим розумінням). Це також показує як Байєсові мережі можуть бути розширені за рівень пропозиційної мови і як саме мова, що використовує тільки безумовні незалежні гіпотези, може представляти довільні ймовірнісні знання. Це аргументує також загальний стан, що пояснює доцільність побудови нових гіпотез для пояснення залежностей замість того, щоб розглядати залежності на рівні мови.

У рамках такого фреймворку ми, відповідно, можемо визначити абдуктивну шему в вигляді $Shema^{ab} = \langle F_{Set}, Hyp^{Set} \rangle$, де F_{Set} – множина тверджень Хорна. Змінні в F_{Set} є неявно універсально квантифікованими. Нехай F_{Set} є множиною базових інстанцій елементів з F_{Set} , а Hyp^{Set} є множиною атомів, що називаються очікуваними або можливими гіпотезами: $Hyp^{Set} = Hyp^{Exp} \cup Hyp^{Psb}$.

Система будує такі припущення відносно базової теорії:

1. Не існує правил у F_{Set} заголовки яких уніфікується з деяким членом у Hyp^{Set} .
2. Якщо F_{Set} є множиною базових інстанцій елементів F , то є можливим

присвоєння числових натуральних числових значень до кожного базового атома такого, що для кожного правила в F_{Set} атоми в тілі правила є строго меншими, ніж атоми в заголовках.

3. Для кожного значення AF^{CF} , що не передбачалося спочатку: якщо AF^{CF} є істинним, тоді принаймні одне з тіл rc_1 (таке, що $AF^{CF} \leftarrow rc_1$ повинно бути істинним). Окрім того, можна стверджувати, що правила в F_{Set} мають місце для кожного атома, який представляється як базовий (тобто такий, що не є гіпотезою).

4. Тіла правил в F_{Set} для кожного атома є взаємовиключними.

На інтуїтивному рівні логічна програма представлятиме наслідки виборів користувача. Даний фреймворк є абдуктивним в сенсі того, що пояснення для спостереження g забезпечуватимуть стислий опис формальних структур в яких g є істинним. Відповідно, мережі суджень можуть означатися як ті, що мають незалежну ймовірність над множиною альтернатив.

Системи, побудовані на імовірнісній абдукції Хорна, не виконують явне оперування з обмеженнями цілісності. Відповідно, кожен вид їх імплементації базується на засобах «Prolog». Сутність реалізації полягає в організації пошуку методом границь і віток, де кожне часткове пояснення з найменшим коефіцієнтом вартості (тобто, найвищою ймовірністю) розглядається в кожен момент часу.

Крім того, такий спосіб імплементації дозволяє підтримувати пріоритетну чергу для множин гіпотез, що може бути розширена до рівня множини пояснень, включаючи також підрівень часткових пояснень. У кожен момент часу множина всіх пояснень розглядається як множина пояснень, що вже згенеровані з додаванням множини пояснень, які можуть бути згенеровані з наявних часткових пояснень у черзі пріоритетів.

Абдукція з вартісними коефіцієнтами (СВА) [176-178] є однією з важливих проблем у побудові міркувань в умовах невизначеності. Сутність цього підходу полягає в пошуку кращого пояснення для множини фактів шляхом пошуку доведення з мінімальною вартістю (вагою) для таких фактів. Тому вживається також термін *абдукція з ваговими коефіцієнтами*, яка розглядається як особливо корисне

застосування, зокрема в контексті інтерпретації природної мови.

Як і для інших застосувань та інтерпретацій теорії абдукції, основне завдання для абдукції з ваговими коефіцієнтах полягає в пошуку на основі даних, що описують спостереження або події відповідної множини гіпотез, які найкращим чином пояснюють наявну множину даних на вході.

На формальному рівні фреймворк абдукції з ваговими коефіцієнтами – це спосіб представлення мультимедійних знань із домену предметної області, що моделюються кортежем: $AF^{CF} = (Hyp^{Set}, Rules, CF, Case^{Set})$, де: Hyp^{Set} – множина гіпотез або пропозиційних тверджень; $Rules$ – множина правил виду: $\{rc_1 \wedge rc_2 \wedge \dots \wedge rc_n^1\} \rightarrow rc_{n+1}$ (де $rc_1, \dots, rc_{n+1} \in Hyp^{Set}$); c – відповідна функція, $(c : Hyp^{Set} \rightarrow Rules^+)$, де $c(rc)$ розглядається як вартість (вагове значення) очікуваності гіпотези $rc \in Hyp^{Set}$; $Case \in Hyp^{Set}$ – множина цілей або підтверджень.

Таким чином, кінцевим завданням є пошук доведення з найменшою вартістю (ваговим коефіцієнтом) для заданого підтвердження, де вартість доведення обчислюється сумою вартостей всіх гіпотез, необхідних для завершення доведення. Відповідно, гіпотеза може стати істинною двома способами:

- 1) як істинна шляхом «оплати» вартості її припущення;
- 2) шляхом доведення.

При цьому варто відзначити, що гіпотеза, як наслідок правила $Rule$, може бути доведена без визначення вартості (ваги) як такої, що є істинною через перетворення всіх антецедентів в $Rule$ в істинні шляхом припущення або доведення. Якщо певна гіпотеза не є консеквентом для жодного правила, тоді вона не може бути доведеною і, відповідно, може бути встановлена як істинна тільки шляхом припущення. Якщо вартість гіпотези є достатньо великою, тоді вона не може бути встановлена шляхом припущення і може бути тільки доведена. Для цього множина гіпотез $HypSet$ розділяється на дві множини – Hyp_1^{Set} і Hyp_2^{Set} . Перша множина містить гіпотези, що не можуть бути доведені (тобто такі, що ніколи не проявляються в правих частинах правила). Друга множина містить всі інші гіпотези.

В загальному випадку, в абдуктивних міркуваннях множина AF^{CF}

розглядається як множина здійснених рішень для проблеми доведення з найменшою вартістю, якщо, відповідно, вона є підмножиною для Hyp_1^{Set} , що є достатнім для доведення множини цілей $Goal$. Відповідно, оптимальним рішенням для проблеми доведення з найменшою вартістю є здійснене рішення, що мінімізує задану загальну вартість, тобто суму вартостей всіх гіпотез AF^{CF} .

3.3 Оцінка ефективності абдуктивного підходу

Оцінка комплексності. Для загальної проблеми абдукції з вартісними коефіцієнтами в роботах продемонстровано зростання складності її рішення (в градієнті найгіршого випадку), але в існуючих дослідженнях даної тематики було розроблено різні підходи. В оглянутих роботах представлено спосіб пошуку абдуктивних доведень мінімальної вартості використанням методів лінійного програмування, базуючись на перетворенні початкової проблеми в лінійну проблему задоволення обмежень.

Водночас за певних умов абдукція з вартісними коефіцієнтами може розглядатися як така, що має поліноміальне рішення. Для цього випадку використовується множина достатніх умов існування поліноміального рішення для абдукції з вартісними коефіцієнтами. Існують інші техніки лінійного програмування, здатні виконувати пошук відповідних субоптимальних рішень.

Розв'язність процедури інференції (логічного висновку). Ланцюг зворотного висновку в стилі «Prolog» з доданою операцією факторизації і без обмежень щодо упорядкування (що означає, що кожен, а не тільки найбільш лівий літерал твердження може бути вирішений) здатним генерувати всі можливі пояснення, консистентні (несуперечливі) з базою знань. Це означає, що кожне очікуване можливе пояснення, консистентне (несуперечливе) з базою знань, категоризується поясненням, яке може бути згенероване в процедурах ланцюга зворотного висновку та факторизації.

Задовільним є випадок, коли процедура гарантовано не генерує жодних пояснень, які не консистентні (суперечливі) з базою знань. Це неможливо в

загальному випадку, оскільки консистентність пояснення з базою знань має перевірятися поза абдуктивною системою інференції міркувань. Слід також відмітити, що не всі неконсистентні пояснення генеруються системою. Навпаки: система може генерувати тільки ті пояснення, що припускають існування літералів, які можна досягнути, відповідно, з початкових формул застосуванням ланцюга зворотного висновку. Таким чином, задача визначеності консистентності (несуперечливості) є нерозв'язною в загальному випадку хоча існують вирішувані під випадки, і ряд пояснень може бути швидко відкинуто як такі, що неконсестентні з початковою базою знань. Зокрема, в більшості випадків припущення може бути відкинуто якщо воно порушує обмеження щодо виду або впорядкування. Тому обмеження виду є особливо ефективними для усунення не консистентних пояснень, особливо в інтерпретаціях на основі природної мови.

Було побудовано ряд фреймворків [179-181] у загальному контексті логічного програмування для інтеграції індуктивних і дедуктивних представлень. Деякі з них були реалізовані, відповідно, певні залишились не реалізованими. Специфікації існуючих середовищ можуть містити також певні види синтаксичної упередженості, а також наявні обмеження щодо цілісності. В загальному випадку, така система являє собою абстрактне середовище для експериментування з різними методами висновку та способами побудови міркувань. Крім того, відмінною рисою таких систем є простота поєднання різнопланових методів, що дозволяє інтегрувати абдукцію й індукцію з загальними правилами.

В концепції абдуктивного навчання (ACL) [182-184] на вході система отримує деякі абдуктивні теорії $T = \langle CLP, Hyp^{Set}, C^{LF} \rangle$ як спосіб представлення знань і дві множини базових атомів, як приклади. Відповідно, засоби ACL дозволяють знаходити абдуктивну теорію з $CLP' \supset CLP$ та $C^{LF} \supset C^{LF}$ як такі, що $T' = Case_{sample} CASE_{Set}^+$ та $\forall Case_{sample}^- \in CASE_{Set}^-, T' \neq_{rc} Case_{sample}^-$. Тут $CASE_{Set}^+$ відповідає за кон'юнкцію всіх позитивних прикладів (позначаються знаком +); відповідно, $CASE_{Set}^-$ відповідає за диз'юнкцію всіх негативних прикладів (позначаються знаком -).

Таким чином, вивчення абдуктивних теорій дозволяє виконувати процес

навчання, базуючись на неповних знаннях, оскільки виведені предикати можуть бути використані для представлення неповних знань.

Таким чином, існуючі алгоритми дозволяють вивчати, перш за все, всі правила, після чого всі обмеження. Вивчення правил виконується ІЛР алгоритмом, що функціонує зверху вниз, а тестування покриття виконується з використанням абдуктивної процедури доведення. В цей алгоритм закладено пучкову пошукову стратегію – спеціальну евристичну функцію, що дає відмінні вагові значення для прикладів, які покриваються засобами абдукції або без її використання. На виході фази вивчення правил отримується відповідна множина правил і множина пояснень за принципом: одне пояснення (можливо, порожнє) для кожного позитивного прикладу і всі можливі пояснення (серед яких не має порожніх) – для кожного негативного прикладу. Вимога відсутності жодного з пояснень для негативних прикладів не є порожнім необхідна для виключення їх всіх у наступній фазі вивчення обмежень.

Обмеження вивчається на основі ІСЛ, причому вхідний потік ІСЛ складається з абдуктивних пояснень для позитивних прикладів (позитивні інтерпретації). Абдуктивні пояснення для негативних прикладів розглядаються, відповідно, як негативні інтерпретації. В підсумку, вивчені правила і правила з множини початкових знань утворюють основу системи знань в ІСЛ.

В загальному, як показують дослідження в даній області, застосування абдуктивних технік покращує ефективність процедур та алгоритмів видобування даних в цілому. Такі реалізації класифікуються на верхньому рівні абдуктивного фреймворку з вартісними значеннями, що дозволяє відповідно видобувати й інтегрувати змістовні знання отримані з необроблених даних. Абдуктивний фреймворк АСЛ здійснює конструювання нових абдуктивних теорій або обмежень цілісності як виду індукції засобами індуктивного логічного програмування при заданій початковій базі знань. В підході, що проектується доцільнішим є використання алгоритмів видобування даних на основі частих входжень наборів елементів для виведення нових фактів зокрема, у формі раніше невідомих розширюваних знань (наприклад, за відсутності значень початкових даних). Такі

виведені факти пізніше використовуються для покращення результатів попереднього видобування даних, наприклад, у задачі класифікації інстанцій.

3.4 Введення задачі класифікації на множині контенту

Ідея використання кластеризації полягає у пошуку схожостей між правилами і демонстрації відмінностей між правилами в тому ж самому кластері за допомогою абдукції. Основна складність даної проблеми полягає у визначенні метрики, яка здатна порівнювати різні правила.

Після введення оптимальної метрики, ми можемо отримати засобами кластеризації кілька множин правил. Фіксуючи множини та порівнюючи схожі правила (тобто такі, що мають схожі висновки з різними заголовками) ми можемо абдуктивно вивести причину існування різних заголовків для правил.

Таким чином, проблема полягає в пошуку розв'язків для таких підпроблем: для заданого набору прикладів з коректною класифікацією кожного прикладу потрібно вибрати значення атрибутів, необхідних для внесення змін з метою отримання відмінних класифікацій.

Таким чином, при коректному виборі метрики і відповідного асоційованого з нею алгоритму кластеризації разом з правильним абдуктивним формулюванням проблеми стає суттєво більш можливим отримання рішення.

Зокрема, одним з підходів до даної проблеми (після того як правило було кластеризовано) повинно бути автоматичне додавання до теорії виводимих значень для кожної зміни, що може бути внесена.

Класифікаційна проблема є одним найбільш важливих питань в області машинного навчання, що полягає відповідно, як ми бачили раніше в прогнозуванні класу об'єктів при заданих характеристиках таких об'єктів і відповідної множини прикладів коректних і хибних класифікацій для інших відомих об'єктів. Тобто в такому підході ми отримуємо суміш дескриптивних і прогнозуючих підходів вирішення проблеми.

Роботи по машинному навчанні представляють цілий ряд алгоритмів здатних класифікувати нові об'єкти з високою якістю відповідно до заданих прикладів. Проте опис екземплярів і нових об'єктів може бути виконаний тільки засобами слабких мов, що як правило базуються на ідеях атрибутів в яких кожен об'єкт описується як такий, що складається з атрибутів та їх значень які відмінні від значень класу.

Для того, щоб розширити можливості мови до рівня представлення гіпотез і прикладів першого порядку є важливим визначити як процес навчання в логіці першого порядку узагальнює навчання виду атрибут-значення. На сьогоднішній день більшість систем даного класу використовує логіку першого порядку для задання структури об'єкта. В пропозиційному випадку об'єкти можуть розглядатися як кортежі значень атрибутів. Відповідно у мовах першого порядку є також можливо описувати об'єкти комплексними типами на верхньому рівні (множини, списки).

Більшість досліджень в даній області належать до проблеми індуктивного логічного програмування, тому більшість таких результатів можуть бути адаптовані до проблеми вбудовування їх в абдуктивний фреймворк.

На практиці є можливим розбиття імовірнісного значення об'єкта на відповідні ймовірності його пропозиційних атрибутів, що дозволяє оцінити входження об'єкта в тренувальні набори замість входжень його атрибутів.

Використання технік логічного програмування для індуктивних задач, дозволило досягти верхньої границі у підсиленні дескриптивної складової класифікаційної проблеми, шляхом використання підмножини логіки першого порядку як нового опису відповідних гіпотез. Проте такий підхід був результатом значного збільшення простору пошуку. В літературних джерелах обґрунтовується, що навчання на основі атрибутів у більшості випадків дозволяє досягти кращих результатів по відношенню до реляційних відповідників, навіть якщо останні базувалися на більш інформативних початкових знаннях. Основна причина полягає в тому, що атрибут не навчання функціонує в значно меншому просторі гіпотез, що дозволяє дослідити даний простір більш ґрунтовно. Другою важливою причиною є

те, що при пропозиційному навчанні приймається до уваги відповідна додаткова інформація, шляхом імовірнісного аналізу гіпотез над тренувальним набором.

Основне поле досліджень полягає у використанні засобів абдуктивних міркувань для задач видобування даних. Слід дослідити, яким чином класифікація може бути інтерпретована як одна з задач абдуктивного логічного програмування, що дозволяє використовувати доменні обмеження задані користувачем. Моделі класифікацій на основі дерев рішень інтерпретуються абдуктивним способом, використовуючи обмеження і отримуючи покращення ефективності у випадках відсутності певних даних. Також слід виконати розширення загального фреймворку з метою розгляду імовірнісної інформації на основі базових та виводимих формальних теорій. Абдуктивний фреймворк, що базується на вартісних коефіцієнтах є придатним для додатків видобування даних, що дозволятиме також покращити загальну якість результатів.

Дане дослідження базується на результатах отриманих з використанням абдуктивних міркувань в області видобування даних. Існуючі класифікатори, навіть якщо вони є базованими на деревах рішень та індуктивних засобах на основі алгоритмів C 4.5 дозволяють отримувати як вихідний результат множину правил для класифікації нових заданих прикладів. Таким чином, більшість існуючих класифікаторів на основі правил функціонують на основі припущення про те, що в момент класифікації ми маємо всю інформацію щодо нових заданих прикладів. Імовірнісні підходи дозволяють класифікаторам на основі правил отримувати найбільш імовірний клас на частоті відсутності атрибута в тренувальному наборі. Водночас, такий вид припущень в більшості випадків інколи призводить до хибної класифікації. Тому використання абдуктивного підходу є особливо корисним щодо вибору: яке саме класифікаційне правило слід застосувати у випадку, якщо новий приклад з відсутньою інформацією потребує класифікації, яку можна виконати, використовуючи знання про домен предметної області.

Слід показати, як будувати пояснення до заданих класифікацій шляхом обчислення відсутніх значень атрибутів у процесі застосування абдукції. Крім того, слід довести, що вибір способу представлення результатів роботи алгоритму

видобування даних у формі дерева рішень або дерева класифікацій є обґрунтованим. Також має значення спосіб представлення додаткових доменних знань у формі обмежень і спосіб вибору техніки міркувань (абдукції), що може суттєво покращити поведінку видобутої моделі (класифікації). Крім того, проєктований метод дозволяє обробляти додаткові доменні знання в формі обмежень цілісності. Доцільно також використовувати механізм виявлення асоціативних правил за повної відсутності зовнішніх знань. Можемо виконати розширення абдуктивного фреймворку для представлення ймовірнісних знань як вартісної оцінки гіпотез. У базі знань класифікаційні правила отримуються на основі шляхів у дереві рішень їх злиттям із асоціативними правилами, отриманими з наборів даних; класифікаційні результати в експериментах об'єднуються з експресивними поясненнями виконаних виборів користувача (користувачів) на момент класифікації.

В літературних джерелах ці правила позначаються як кореляційні. Важливою є адаптація існуючих процедур, здатних знаходити правила кореляцій, до великих баз даних.

Основне завдання полягає в тому, щоб продемонструвати як класифікаційна задача може бути перетворена безпосередньо в абдукційну проблему шляхом застосування дерев рішень. Для вирішення цього завдання необхідно ввести формальні означення, які описують процес класифікації на основі дерев рішень і дозволяють виконувати трансформацію дерев рішень у відповідну абдуктивну логічну програму.

Для заданої мережі рішень з обмеженнями CN і шляху $route$ у CN позначимо через $Case^{Example}(route)$ деякий приклад: $sc_1 = v_1, \dots, sc_k = v_k$, де кожне sc_i не є листовим вузлом у CN і кожне v_i є значенням, що позначає гілку, яка виходить з sc_i . Означимо через $Case^{CLASS}(route)$ - клас, що позначає лист вузла для $route$.

Визначимо концепції класифікації прикладів засобами дерев рішень за відсутньої інформації. Для цього слід переписати звичайне означення класифікації у вигляді прикладу окремого випадку, коли немає відсутньої інформації.

Означення 3.4. Нехай CN^{Sovle} є мережею рішень із введеними обмеженнями:

$CN^{Solve} : CH^{CNSolve} = \bigcup_i \left[CS_i = \bigcup_j CSet_j \right]$, $Solve^{sample}$ - екземпляр (приклад) рішення та $Class$ –

кейс клас рішення. Будемо вважати, що екземпляр $Solve^{sample}$ може бути класифікований як $Class$ засобами CN через Λ (позначається через $CN \rightarrow Class, \Delta$), якщо існує шлях $route$ у CN із вузлом $Case^{CLASS}(route) = Class$, причому $Solve^{sample} \cup Case^{Example}(route)$ є теж відповідним екземпляром (прикладом) рішення і $\Lambda = Case^{Example}(route) / Solve^{sample}$.

Якщо $CN \xrightarrow{CN, \{ \}} Case_{sample}$, то стверджується, що екземпляр (приклад) $Case_{sample}$ класифікується як CN засобами CN , що може бути представлено як $CN \xrightarrow{CN} Case_{sample}$.

Слід також відзначити, що в представленні $CN \xrightarrow{CN, \Lambda} Case_{sample}$ умова за якої $Case_{sample} \cup Hyp_{Names}^{Set}(route)$ повинен бути екземпляром, гарантує, відповідно, що значення атрибуту для $Case_{sample}$ є сумісними з множиною тестів, представлених через $(route)$. Іншими словами, якщо для деякого атрибуту rc , $rc = v \in Case_{sample}$ і $rc = v' \in Hyp_{Names}^{Set}(route)$, тоді $v = v'$. Крім того, Λ , представлятиме розширення до рівня прикладу $Case_{sample}$, необхідне для виконання його класифікації як CN з вибраного шляху.

Розглянемо спосіб перетворення мережі рішень CN в абдуктивну логічну програму з обмеженнями. Нехай CN – мережа рішень і $route$ – шлях у CN . Тоді правило $Rule_{route} = [rc_{route} \rightarrow sc_{route}]$ асоційоване з $route$ буде твердженням виду $Solve^{sample} \leftarrow \{sc_1(v_1), \dots, sc_n(v_n)\}$, при якому:

- 1) $Case^{Example}(route) = Class$;
- 2) $Solve^{sample}(route) = \{sc_1 = v_1, \dots, sc_n = v_n\}$.

Слід зазначити, що імена атрибутів розглядаються як унарні предикатні символи і пара атрибут/значення $a = v$ відображається в атом $rc(route)$. Більше того, при заданому (v) в **Означенні 3.4** при заданому шляху $route$, $Hyp_{Names}^{Set}(route)$ слід розглядати як екземпляр (приклад). Програма P_T на основі тверджень Хорна, асоційована з деревом рішень CN , є множиною правил виду:

$$P_T = \{r_{route} \mid route\},$$

де $route$ є шляхом в CN .

В підсумку, ми виконуємо зв'язування з деревом рішень CN множини CLF канонічних обмежень цілісності, що містять заперечення виду:

$$\leftarrow rc(x), rc(y), x \neq y$$

для кожного атрибута $rc \in RC^{Set}$.

Імена атрибутів розглядаються як виведені предикати, і множина всіх виведених предикатів представлятиме можливі класифікації (детальніше розглянемо нижче). Обмеження цілісності, означені в такий спосіб, забезпечують ситуацію за якої в кожній консистентній множині виведених предикатів кожен атрибут зустрічатиметься не більше одного разу.

Нехай $\Delta_{Case_{sample}}$ є множиною атомів для заданого екземпляру (прикладу) $Case_{sample} = \{rc_1 = v_1, \dots, rc_k = v_k\}$:

$$\Delta_{Case_{sample}} = \{rc_1(v_1), \dots, rc_k(v_k)\}.$$

Таким чином, наведені формальні представлення дають всі необхідні складові для прив'язки абдуктивного фреймворку до дерева рішень CN .

Означення 3.5. Для заданої мережі рішень CN абдуктивний фреймворк AF_{CN}^{ab} , асоційований з CN , представлятиметься кортежем: $AF_{CN}^{ab} = \langle CLP_{CN}, Hyp_{Names}^{Set}, CH_{CN} \rangle$, де CLP_{CN} – логічні програми з обмеженнями, асоційовані з CN , CH_{CN} – множина обмежень накладених на CN ; Hyp_{Names}^{Set} – множина імен атрибутів гіпотез.

Наступне твердження дозволяє формалізувати відповідність між класифікаціями в CN й абдуктивними поясненнями в AF_{CN}^{ab} .

Твердження 3.1. Нехай CN є деревом рішень і AF_{CN}^{ab} є відповідним абдуктивним фреймворком, а $Case_{sample}$ – екземпляром (прикладом). Тоді Δ є $\Delta_{Case_{sample}}$ буде мінімальним поясненням для $Class$ відносно AF_{CN}^{ab} , якщо і тільки якщо для деякого шляху $route$ у CN $CN \xrightarrow{Class, \Delta} Case_{sample}$ і $\Delta = \Delta_{\Delta}$.

(\Leftarrow) Припустимо, що $CN \xrightarrow{Class, \Delta} Case_{sample}$. Тоді, згідно введених означень для деякого шляху $route$ в CN , $Case_{sample} \cup Solve^{sample}(route)$ є екземпляром (прикладом)

і $\Lambda = \text{Solve}^{\text{Sample}}(\text{route}) / \text{Case}_{\text{Sample}}$. Розглянемо правило rc_{route} :

$$\text{Class} \leftarrow rc_1(v_1), \dots, rc_n(v_n).$$

Нехай $SC(r_{\text{route}}) = \{rc_1(v_1), \dots, rc_n(v_n)\}$. Тоді $SC(rc) = \Delta_{\text{Solve}^{\text{sample}}(\text{route})}$ і

$[route_{CN} \cup SC(rc_{\text{route}})] \models \text{Class}$. Нехай $\Delta = SC(rc_{\text{route}}) / \Delta_{\text{Case}_{\text{sample}}}$. Звідси випливає, що

$[route_{CN} \cup \Delta_{\text{Case}_{\text{sample}}} \cup \Delta] \models \text{Class}$ та $\Delta = \Delta_{\Lambda}$. Більше того, оскільки $\text{Case}_{\text{sample}} \cup \text{Solve}^{\text{Sample}}(\text{route})$

є екземпляром (прикладом рішення), то $(\Delta_{\text{Case}_{\text{sample}}} \cup \Delta) \subseteq \Delta_{[\text{Case}_{\text{sample}}] \cup \text{Solve}^{\text{Sample}}(\text{route})}$, і

$SC_{CN} \cup \Delta_{\text{Case}_{\text{sample}}} \cup \Delta \cup CH_{CN}$ є консистентними. Таким чином, Δ виконує роль пояснення

для Class при заданому $\Delta_{\text{Case}_{\text{sample}}}$. Згідно способу побудови Δ є також $\Delta_{\text{Case}_{\text{sample}}}$ –

мінімальним поясненням.

Припустимо, що $\Delta \in \Delta_{\text{Case}_{\text{sample}}}$ – мінімальним поясненням для Class . Очевидно, що

$\Delta \cap \Delta_{\text{Case}_{\text{sample}}} = \{\}$ (на основі мінімальності Δ). Оскільки, згідно способу побудови

CH_{CN} , $[CH_{CN} \cup \Delta_{\text{Case}_{\text{sample}}} \cup \Delta] \models \text{Class}$, то існує шлях $route$ в CN , за якого rc_{route} є правилом

виду:

$$\text{Class} \leftarrow [rc_1(v_1), \dots, rc_n(v_n)] \text{ і } \{rc_1(v_1), \dots, rc_n(v_n)\} \subseteq [\Delta_{\text{Case}_{\text{sample}}} \cup \Delta].$$

Нехай $\Lambda = \text{Solve}^{\text{Sample}}(\text{route}) / \text{Case}_{\text{sample}}$. За способом побудови і мінімальності Δ

маємо, що $\Delta = \Delta_{\Lambda} = \{rc_1(v_1), \dots, rc_n(v_n)\} / \Delta_{\text{Case}_{\text{sample}}}$. В підсумку, буде очевидно, що

$\text{Case}_{\text{sample}} \cup \text{Solve}^{\text{sample}}(\text{route})$ є екземпляром (прикладом) на основі консистентності

$SC \cup \Delta_{\text{Case}_{\text{sample}}} \cup \Delta \cup CH_{CN}$ і загального спостереження $\Delta_{\text{Case}_{\text{sample}} \cup \text{Solve}^{\text{sample}}(\text{route})} = \Delta_{\text{Case}_{\text{sample}}} \cup \Delta$.

3.5 Імплементация релевантных доменных знаний

Задачі класифікації з точки зору абдуктивних технік можна розглядати як альтернативний процес, і водночас, як еквівалентний спосіб виконання класифікації на основі дерева рішень. При цьому слід відзначити, що абдуктивні міркування, які вимагаються в такому альтернативному представленні, є досить обмеженими і не використовують повною мірою потенціальну потужність абдуктивних міркувань у логічному програмуванні. Зокрема, виконання відповідних перетворень вимагає

використання наперед визначених канонічних множин обмежень цілісності, в яких при побудові пояснень уникається включення відмінних значень для одного і того ж атрибуту. Вважається, що ефективним способом використання абдуктивних міркувань є додавання доменно-специфікованих знань для покращення результативності задач класифікації. Таким чином, *доменно-специфіковані* знання можна брати до уваги при виконанні класифікації багатьма способами, зокрема шляхом виділення класифікації або надання преференції певному виду класифікації над іншими видами. Таким чином, приєднання доменно-специфікованих знань до стандартних алгоритмів класифікації на основі дерев рішень не завжди даватиме пряму та просту імплементацію і, таким чином, може вимагати значних модифікацій базових алгоритмів. Разом з тим, абдуктивні фреймворки і їх існуючі імплементації, як правило, вже містять механізми, які можуть бути безпосередньо використані для представлення і управління доменно-специфікованими знаннями (тобто знання, представлені в певному домені).

В загальному випадку ми використовуємо обмеження цілісності (за винятком канонічних) в абдуктивному фреймворку як спосіб вираження доменно-специфікованих знань.

Як показують існуючі застосування, обмеження цілісності можуть бути використані для додавання знань, що стосуються значень різних атрибутів, включаючи тв., які не входять до початкового дерева рішень. Таким чином, ця особливість може допомогти покращити ефективність задач класифікації, оскільки обмеження цілісності можуть додавати знання, релевантні до атрибутів, що вже входять до дерева рішень, але не представлені в ньому явно.

В багатьох випадках дерева рішень можуть використовувати ймовірнісні вимірювання на рівні результату тестування пар атрибут/значення. Іншими словами, кожна вітка дерева має мітку у вигляді значення, що відповідає атрибуту, який позначає батьківський вузол, а також мітку у вигляді ймовірнісного значення, що відображає імовірність того, що результат спостереження приписуватиме саме дане значення для заданого атрибуту. Такий вид імовірнісної інформації є корисним при класифікації нових екземплярів з відсутньою інформацією про атрибути. Навіть в

цьому випадку в задачах класифікації додаткові доменні знання можна взяти до уваги для динамічного одержання кращих значень імовірності на можливих класифікаціях нових екземплярів (прикладів) із недостатньою специфікацією (відсутністю частини релевантної інформації).

Представлена вище методика потребує специфікованих зовнішніх знань про домен, який ми використовуємо як обмеження цілісності в абдуктивному фреймвоку. В даному дослідженні ми розглядаємо доменні знання як правила, що є похідними від людини-експерта. Як було показано вище, відмінним способом отримання правил в домені є їх виведення з вхідних даних.

Використання асоціативних правил (так званих активних правил) є можливим і простим в абдуктивних фреймворках загального призначення і дозволяє побудувати повністю автоматизований абдуктивний фреймворк поєднанням двох найбільш важливих парадигм видобування даних: 1) парадигма класифікації на основі дерев рішень; 2) парадигма асоціативних правил.

Тому в якості окремої задачі можна виокремити побудову абдуктивних технік і методик видобування даних на основі асоціативних правил як обмежень цілісності шляхом використання наборів фактичних даних (в нашому випадку, даних технологічного процесу буріння).

Проблема пошуку обмеження цілісності може бути вирішена шляхом інтерактивного аналізу певних асоціативних правил отриманих на основі апріорних алгоритмів. Оскільки існує деяка збалансованість між негативними і позитивними значеннями певних атрибутів, отримані правила мають високі значення коефіцієнта довіри і коефіцієнта підтримки. Але для підвищення їх практичного застосування необхідно використовувати окремі метрики (наприклад, метрики «піднімання» та «впливу»). Отримані в результаті асоціативні правила дозволяють представляти фактичні залежності між атрибутами.

Тестування результатів задач класифікації з використанням абдуктивного фреймворку, що починає свою функціональність з дерева рішень як теорії й асоціативних правил щодо обмежень цілісності, доводить здатність системи виводити значення коректного класу навіть, якщо не має відсутніх значень.

Абдуктивний висновок з обмеженнями може розглядатися як спосіб інтерпретації для виконання відтинань у дереві рішень, хоча в ряді випадків абдуктивний висновок може бути більш потужним ніж відтинання.

Основне завдання полягає в тому щоб продемонструвати шлях поєднання результаті алгоритмів класифікації у видобуванні даних. Він базується на дереві рішень з додатковими доменними знаннями, які походять з асоціативної моделі видобування даних і, відповідно, базуються на асоціативних правилах. Ці правила є цікавими з точки зору отримання вхідного потоку для абдуктивного середовища, що здатне класифікувати, а також специфікувати відповідні міркування, використані для побудови класифікацій (у тому числі при прогнозуванні відсутніх значень). В рамках такого підходу отримуються нові особливості, що представляються на рівні нової абдуктивної шемі, яка використовується в даному випадку, а саме:

- 1) цілі є прийнятними;
- 2) припущення виконується на різних рівнях специфікованості;
- 3) надлишковість припущень повинна враховуватись (зокрема, при побудові доведень).

Виконаний аналіз показує, що абдукція з вартісними коефіцієнтами є одним із кращих фреймворків, що задовольняє переліченим вимогам. Також щодо абдуктивного фреймворку, що базується на предикатній логіці, використаний в аналізі, вартісні значення (вагові значення), що асоціюються з кожним виводимим твердженням і правилами у початкових знаннях, що представляють асоційовані з ними міри невизначеності.

На рисунку 3.1 представлено кінцеву діаграму класифікаційної системи. Показано, що вхідний потік даних складають інстанціації для класифікації і відповідний набір даних. Дерево рішень індукується з набору даних за допомогою алгоритму класу C 4.5, а інстанціації алгоритмів класу Argotі видобувають часті входження і відповідні асоціативні правила. Абсолютне значення підтримки частих входжень є параметром, що використовується для обчислення вартості їх абдукції (видобування).

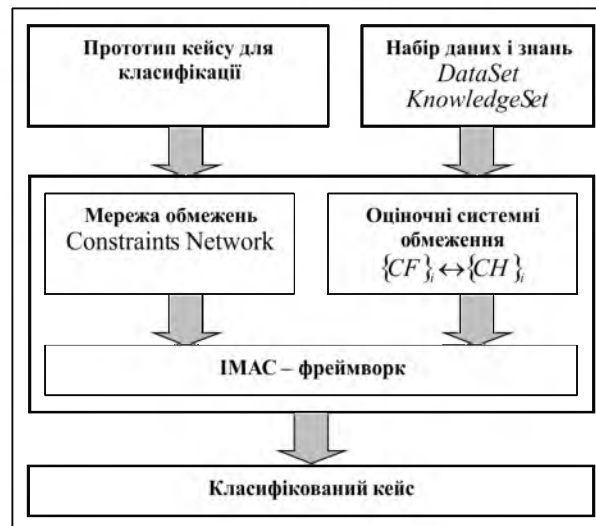


Рисунок 3.1 – Абдукційно-класифікаційна схема виділення кейсів

Таким чином, проєктований фреймворк базується на абдукції з вартісними (ваговими) коефіцієнтами, де кожен виведений елемент має асоційований з ним вартісний (ваговий) коефіцієнт, який необхідно активувати при побудові припущення. В запропонованому підході правила також мають асоційоване значення вартості (ваговий коефіцієнт), яке необхідно бути активований, якщо правило використовується в процесі класифікації. Відповідно, кінцевим класом буде вважатися клас, що імплікується мінімальною вартістю (мінімальним сумарним ваговим коефіцієнтом) відповідного абдуктивного доведення.

Необхідність розширення базового фреймворку шляхом включення вартісних (вагових) коефіцієнтів приймається на основі таких міркувань:

1) потрібно введення кількісної міри для вибору серед відмінних можливих рішень (у тому числі абдуктивних доведень). Слід відмітити, що класифікація, побудована за даним підходом, може мати більше одного пояснення;

2) в базі знань використовуються імовірнісна інформація для відкидання частини малоімовірних доведень.

Процедури, що використовуються для аналізу пошукового простору, є процедурами повного перебору в глибину з можливими відтинаннями, що виконуються шляхом використання базової структури обмеження цілісності. Іншим видом обмежень, що мають міститися в кожному здійсненому рішенні, є

верифіковане виконання кроків обробки даних.

3.6 Оцінка застосування коефіцієнтів впевненості у абдуктивному фреймворку

Наведемо основні характеристики абдуктивного фреймворку, що використовується:

1) кожен атом, що входить принаймні в одне правило є асоційованим із цілочисельним значенням, яке представляє його вартість (вагу). Згідно розширеної інтерпретації тверджень Хорна матимемо:

$$\left[\text{Head} : -\text{Body} \right] \rightarrow \left[\text{SC} : -\text{RC} \right],$$

$$\text{SC}^{CF_w(\text{SC})} \leftarrow \text{RC}_1^{CF_w(\text{RC}_1)} \wedge \text{RC}_2^{CF_w(\text{RC}_2)},$$

де SC, RC_1, RC_2 представляють відповідні входження в правилах з їх вартісними ваговими значеннями, наведені в правій частині. 2) припустимо, що заголовок правила має вартісну вагу меншу або рівну тій, що припускається в тілі:

$$CF_w(\text{SC}) \leq CF_w(\text{RC}_1) + CF_w(\text{RC}_2).$$

3) надлишковості використовуються для пошуку доведення з найменшою вартістю (ваговим значенням).

Вартості (вагові значення), що, відповідно, асоціюються кожному атому, правилу або рішенню співвідносяться з їх імовірністю.

Вагове значення атома визначається кількістю кортежів, в яких цей атом є хибним:

$$\text{glb}(|\text{DataSet}|) - \text{glb}(\text{Rule}^{\text{Set}}) = \text{glb}(\overline{\text{Rule}^{\text{Set}}}),$$

де $|\text{DataSet}|$ – кількість кортежів у наборі даних $|\text{DataSet}|$. Відповідно, в термінах імовірності вартість (вага) атома R з $(\overline{\text{Rule}^{\text{Set}}})$ становитиме:

$$CF_w(\text{RulesSet}.R) = |\text{DataSet}| \cdot RC(R).$$

Подібним чином кожне правило має вартість (вагу), рівну кількості кортежів,

в яких правило є хибним. Наприклад, для використання правила $SC \leftarrow RC$ обчислення вартості (ваги) складатиме:

$$CF_w(SC \leftarrow RC) = \text{glb}(RC \wedge \overline{SC})$$

У термінах коефіцієнта впевненості для правила матимемо:

$$CF_w(SC \leftarrow RC) = \text{glb}(RC \wedge \neg SC) = |DataSet| CLP.RC(R \wedge \neg SC) =$$

$$\begin{aligned} & |DataSet| CLP.RC(\neg SC | RC) CLP.RC(RC) = \\ & |DataSet| (1 - CLP.RC(SC | RC)) CLP.RC(RC) = \end{aligned}$$

$$(1 - \text{verify}(SC \leftarrow RC)) \text{glb}(RC)$$

Абдуктивним рішенням Sol^{AB} для цілі $Goal.Case_i$ вважатимемо множину атомів $rc_1 \wedge \dots \wedge rc_n$, тобто $Sol^{AB} \rightarrow Goal.Case_i$, де $Sol^{AB} = \{rc_1 \wedge \dots \wedge rc_n\}$.

В цій ситуації оптимальною є вартість $CF_w(Sol^{AB})$, інверсно пов'язана з частотою рішення Sol^{AB} .

Твердження 3.2. Із припущення щодо взаємного виключення серед інвертованих атомів матимемо, що:

$$|DataSet| CLP.RC(\overline{Sol^{AB}}) = CF_w(Sol^{AB})$$

Доведення. $|DataSet| CLP.RC(\overline{Sol^{AB}}) = |DataSet| CLP.RC(\overline{rc_1} \vee \dots \vee \overline{rc_n})$. Шляхом

використання гіпотези щодо взаємного виключення отримаємо, що

$$\begin{aligned} & |DataSet| CLP.RC(\overline{rc_1} \vee \dots \vee \overline{rc_n}) = |DataSet| (CLP.RC(\overline{rc_1}) + \dots + CLP.RC(\overline{rc_n})) = \\ & CF_w(rc_1) + \dots + CF_w(rc_n) = CF_w(Sol^{AB}) \end{aligned}$$

Такий підхід є нещільною апроксимацією ймовірності $CLP.RC(\overline{Sol^{AB}})$, оскільки не має прив'язки до проміжку $[0, 1]$. Проте, в даному випадку, це не проблематично, оскільки відсутня реальна потреба в фактичних значеннях інвертованої частоти, а існує лише потреба в рішеннях з низькою вартістю, найбільш імовірних у певних відношеннях. Таким чином, останнє твердження демонструє нам зв'язок між вартістю та імовірністю абдуктивного рішення. В підсумку, фреймворк узагальнюється настільки, наскільки це можливо, в такий спосіб:

1) усі предикати є виведеними, навіть ті, що знаходяться в заголовках правил;

2) база знань може містити петлі.

Процедура доведення, імплементована в цьому підході, може обробляти обидві дані вимоги. Ця процедура була вперше розроблена як надбудова до SICStus Prolog v3, що дозволяє обробляти петлі на етапах пошуку і, відповідно, відтинати певні можливі доведення відповідно до вагових значень атомів, що доводяться. Залишкові процедури, такі як: сортування здійснених рішень і видалення деяких інших обмежень цілісності на атоми – виконуються в наступних кроках обробки.



Рисунок 3.2 – Видобування знань в процесі кейс-солвінгу

На рисунку 3.2 представлено загальну структуру системи, а на рисунку 3.3 деталізовано особливості офлайнових обчислень при побудові бази знань.

Додатково спроектований фреймворк допомагає користувачу зрозуміти причини та міркування, згідно яких ті чи інші інстаціації були асоційовані з виділеним класом.

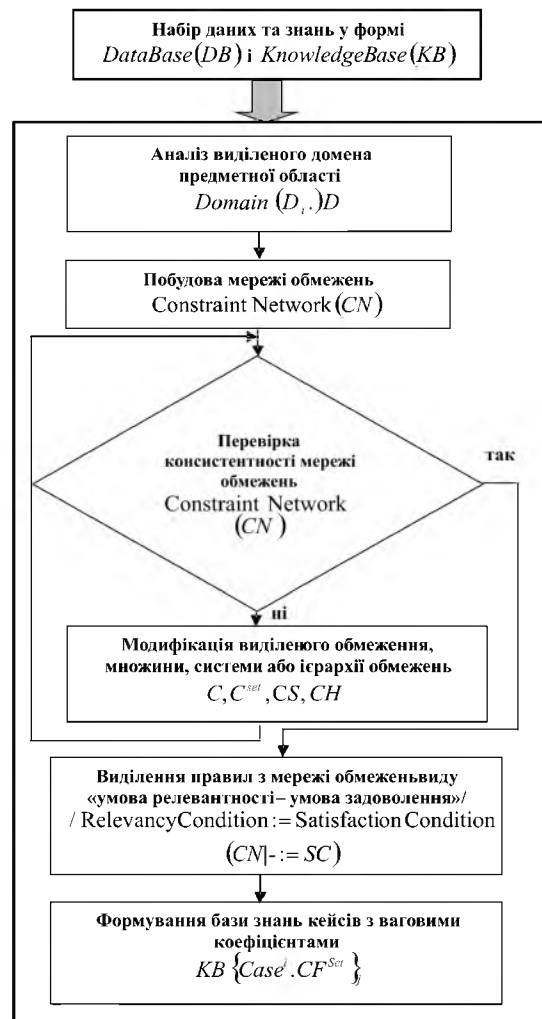


Рисунок 3.3 – Локалізація процедури побудови БЗ кейсів

Слід відмітити, що всі правила видобуваються автоматично з набору даних шляхом використання апріорного алгоритму. Єдине, що ми отримуємо на вході, – це набір даних і приклад для класифікації. Інша особливість полягає в тому, що інформація, яка індукується та використовується абдуктивною системою в загальному випадку, повністю ігнорується класифікаторами на основі дерев рішень.

Приклади реалізацій показують, що абдуктивні системи здатні виводити певну інформацію щодо екземпляра, які необхідно класифікувати, використанням домену знань, виведеного з набору даних. Додаткова похідна інформація (тобто виведена наявними засобами) дозволяє будувати певні покращення функціонування класифікатора, тобто отримувати класифікації вищої якості.

Сутність тестування системи полягає в запуску задач класифікації на різних екземпляра для демонстрування того, що у всіх випадках отримується однакова

класифікація (як з дерева рішень так і з абдуктивного фреймворку відповідно), але з деякою відчутною різницею щодо пояснення таких результатів. Різниця полягає тільки в способах пояснення отриманих результатів, оскільки не всі моделі представлення знань будуть такими, що можуть інтерпретуватися людиною відповідним чином.

Крім того, існують випадки, коли абдуктивна система не використовує жодного класифікаційного правила. Можлива ситуація коли вартість (вагове значення) абдукції в деяких випадках є далекою від мінімуму. В таких випадках значенням класифікаційного пояснення є таким: інформація щодо об'єкту, який необхідно класифікувати, не є достатньою для прив'язки його до певного специфікованого цільового класу. Відповідно, обирається найбільш імовірний клас.

При проведенні тестування множини записів розбиваються на два відмінні набори даних – тренувальний набір і тестувальний набір. Різні набори створюються задля уникнення неоднозначності внаслідок окремого часткового розбиття. Відповідні експериментальні результати показують, що класифікація на основі дерева рішень і абдуктивна класифікація є еквівалентними відносно точності класифікації. За умови, що немає відсутніх даних у тестовому наборі.

Абдуктивний фреймворк демонструє суттєві покращення функціональності при додаванні нової інформації (наприклад, при видобуванні асоціативного правила в базі знань). Можна бачити, що додавання асоціативних правил до бази знань абдуктивної системи дозволяє такій системі підвищувати точність класифікації шляхом отримання результатів, досить близьких до класифікатора на основі дерева рішень.

Враховуючи наявність великих масивів даних, які легко збираються та зберігаються засобами відповідних інформаційних технологій, особливої актуальності набуває задача видобування даних. Тому саме видобування даних розглядають як один з найбільш важливих кроків у процесі відкриття та видобування знань. На сьогодні отримано ряд алгоритмів і технік в ряді предметних областей, а також спроектовано відповідні середовища, що реалізують технології видобування даних. Такі середовища, відповідно, підтримують ряд етапів, які

безпосередньо передують процесу видобування даних (наприклад, очистка даних), а також процеси, які слідує після застосування алгоритмів і технік видобування даних (наприклад, засоби візуалізації результатів). Також важливим є забезпечення концептуального контексту, в якому можна обробляти результати кроків процедури видобування даних для отримання відповіді на задачі більш високого рівня складності, ніж ті, що безпосередньо забезпечуються обчисленими моделями видобування даних. Наприклад, видобування асоціативних правил з деякої множини транзакцій дозволяє отримати відповідь на ряд базових запитань, а також вирішувати більш цікаву задачу за допомогою відповіді на питання, які безпосередньо не представлені в даних. Це вимагає використання, наприклад, асоціативних правил, обчислених із урахуванням наявних даних і структури проєктованих очікуваних рішень.

Таким чином, ряд дослідників виділяє результати застосування алгоритмів видобування даних як вхідні дані відповідних середовищ, що можуть покращувати якість таких даних і ефективно їх використовувати (наприклад, для задач логічного висновку та побудови експертних суджень на основі результатів видобування даних) для задоволення вихідних потреб користувача в засобах автоматизації обробки даних.

3.7 Обґрунтування процедури видобування мультимедійних даних і знань

Як відомо, індукція – це один із методів узагальнення. Більшість алгоритмів видобування даних базується, відповідно, на індуктивних техніках, що полягають в певному способі виведення інформації, яка узагальнюється з вхідних даних. Проте дуже часто застосування індуктивних технік на масивах сирих даних не достатньо для генерації відповідей на запити користувача. Це, природно, зумовлює потребу додаткової обробки даних із використанням дедуктивних технік. З метою отримання відповідей на запити високого рівня складності необхідне середовище, здатне підтримувати процес машинних міркувань, оскільки у більшості випадків пошук рішення для таких запитів можна розглядати як аналіз множини гіпотез, що

задовольняє певним вимогам. Методу абдукції є особливо ефективним в задачах класифікації і пояснення результатів застосування технології видобування даних. Зокрема, можна вводити класифікаційні правила, виведені з даних, і інформацію, отриману від експертів предметної області. Також можна використовувати асоціативні правила з введеними коефіцієнтами впевненості, що дозволяє підвищити загальну якість результатів і рівень автоматизації процесу без безпосередньої потреби в наявності попередніх знань і експертного досвіду щодо предметної області.

Видобування даних може бути означено як процес пошуку кореляцій або шаблонів серед різних полів у великих базах даних (перш за все, реляційних). Відповідно, видобування даних є важливим кроком процесу видобування знань, що полягає в застосуванні інтелектуальних методів з метою видобування шаблонів даних з великих баз даних. Зокрема, найбільш важливими моделями для опису даних інформативним способом, розробленими в даній області, є: 1) *класифікаційні задачі* – прогнозують категорійні мітки для заданих прикладів; 2) *асоціативні правила* – виконують пошук значимих залежностей серед множин атрибутів; 3) *метод кластеризації* – полягає в пошуку оптимальних розбиттів множини елементів (наприклад, розбиття наборів даних).

Класифікація даних вимагає виконання двох послідовних кроків. Перший полягає у побудові моделі, що описує заданий набір прикладів шляхом прив'язки мітки класу до кожного елементу. Другий розглядає використання певної моделі для класифікації нових прикладів (наприклад, прогнозування категорійної мітки). У літературних джерелах використовуються різні моделі обчислення класифікаційних задач, але найбільший інтерес представляє саме *дерево рішень*.

Дерево рішень являє собою деревовидну структуру, в якій кожен внутрішній вузол позначає певний вид тестування атрибуту, кожна вітка, відповідно, представляє результат тесту, а листові вузли – класи. Індукція дерева рішень полягає у побудові такого дерева з тренувальних наборів прикладів, щоб його можна було використати для класифікації нових прикладів для заданих значень їх атрибутів шляхом слідування від кореня до листків. Завдяки цій структурі цілком можливе

перетворення дерев рішень у відповідні класифікаційні правила, які можна легко вставити в середовище побудови суджень (логічного висновку). Слід відмітити, що ряд інструментів машинного навчання (такі, наприклад, як алгоритм С 4.5) містять генератор для побудови класових наборів правил.

Особливістю такого підходу є можливість вбудовувати класові набори правил в середовище побудови суджень (логічного висновку). Це дозволяє в деяких випадках краще класифікувати нові приклади за наявності зовнішньої інформації (як доменні знання, релевантні до певної предметної області).

Нехай $AnswerSet$ – множина імен атрибутів і $\{Case_i, Class\}_i$ – множина класів (можливих класифікацій). Припустимо, що кожній множині рішень можна співставити присвоєння скінченної послідовності значень над скінченою множиною значень V . Відповідно, *приклад* рішення $Solve^{sample}$ слід розглядати як множину пар атрибут/значення виду:

$$Solve^{Solution} = \{rc_1.value = v_1, \dots, rc_n.value = v_n\},$$

де rc_i є іменем i -того атрибуту.

Означення 3.6. Мережею обмежень CN над $AnswerSet$ і $\{Case_i, Class\}_i$ є множина обмежень, для якої:

- 1) кожен нелистовий вузол, позначений атрибутом, $rc \in AnswerSet$;
- 2) кожен листовий вузол, позначений класом, $CF_w \in Case, Class$;
- 3) кожна вітка, позначена значенням, $v \in V$;
- 4) всі значення, що помічають усі вітки, які виходять із заданого вузла, є відмінними;
- 5) мітки шляху теж є відмінними;
- 6) суміжні системи обмежень є несуперечливими.

Відповідно, остання вимога формалізує той факт, що на кожному шляху для кожного атрибута може бути виконаний тільки один тест.

Асоціативні правила корисні з точки зору визначення кореляції між атрибутами великих множин входжень даних. Відповідно, асоціативні правила демонструють умови щодо значення атрибуту, що зустрічається з певною частотою

в заданому наборі даних.

Своєю чергою, асоціативні правила забезпечують інформацію даного типу в формі тверджень «якщо-тоді». Ці правила обчислюються на основі даних і, на відміну від правил «якщо-тоді» в математичній логіці, асоціативні правила є ймовірнісними за своєю суттю. А саме: фактично, на додаток до частини антецеденту (частини «якщо») і консеквенту (частини «тоді») асоціативне правило має також два коефіцієнта, які виражають ступінь невизначеності, що стосується даного правила. Таким чином, в асоціативному аналізі антецеденти та консеквенти є множинами входжень, що не перетинаються (тобто, не мають спільних елементів). Відповідно, перший коефіцієнт називається коефіцієнтом *підтримки* правила. Він є певною кількістю транзакцій, що містять всі елементи в частинах антецедентів і консеквентів для правила. В більшості випадків це виражається як процентне відношення загальної кількості записів у базі даних. Другий коефіцієнт розглядається як коефіцієнт *впевненості (довіри)* для правила. Коефіцієнт *впевненості (довіри)* є співвідношенням кількості транзакцій, що містять усі елементи як консеквентні, так і в антецеденті (виражає коефіцієнт підтримки) до кількості транзакцій (містить усі елементи в антецеденті).

Коефіцієнт підтримки представляється як імовірність того, що випадково вибрана транзакція з бази даних буде містити всі входження у антецеденті і консеквенті, а коефіцієнт *впевненості*, відповідно, імовірність того, що випадково вибрані транзакції будуть містити всі пункти в консеквенті за умови, що відповідні транзакції міститимуть всі входження в антецеденті.

Коефіцієнт *підняття* є одним із параметрів асоціативного аналізу. Він є співвідношенням *впевненості (довіри)* до очікуваної *впевненості (довіри)*; очікувана *впевненість (довіра)* є *впевненістю (довірою)*, коли антецедент не впливає на консеквент. Таким чином, очікувана *впевненість (довіра)* може бути визначена як кількість транзакцій, що містять консеквент, поділена на загальну кількість транзакцій.

Таким чином, коефіцієнт підняття є значенням, що надає інформацію щодо зростання ймовірності для консеквентна (частина правила після «then») при

заданому антецеденті (частина правила після «if»).

Задача видобування асоціативних правил за заданих мінімальних порогових значеннях підтримки і впевненості (довіри) може бути розбита на дві підпроблеми:

1) пошук всіх входжень (множин входжень), що мають підтримку транзакцій, вищу за значення мінімальної підтримки. Відповідно, підтримка для множини входжень є кількістю транзакцій, що містять задану множину входжень. Множина входжень із значенням підтримки, більшим або рівним пороговому значенню, розглядається як множина частих входжень (або *обширна множина входжень*). Ця підзадача переадресується на виконання алгоритму APRIORI.

2) генерація всіх асоціативних правил із мінімальною підтримкою та впевненістю (довірою) з множини всіх обширних множин входжень. Ця задача переадресується на виконання прямого алгоритму, що містить такі кроки: 1) для кожної обширної множини входжень LE знайти всі непорожні підмножини; 2) для кожної такої підмножини $LE^{sub} \subset LE$ з LE вивести правило $LE \rightarrow LE^{sub}$, якщо і тільки якщо:

$$CF = \frac{SF(LE)}{SF(LE^{sub})} \geq CF^{min} = CF^{thres},$$

де: CF – коефіцієнт впевненості (довіри); SF – коефіцієнт підтримки; CF^{min} – мінімальне значення коефіцієнта впевненості; CF^{thres} – порогове значення коефіцієнта впевненості.

Видобування асоціативних правил використовується для рішення задач класифікації [185-187].

Технологія кластеризації даних полягає в розбитті даних на групи схожих об'єктів. Зрозуміло, що представлення даних невеликою кількістю кластерів призводить до втрат в точності, але дозволяє досягти певного рівня спрощення загальної задачі. Таким чином, задача кластеризації полягає в структуризації заданої множини некласифікованих інстанціцій множини екземплярів створенням концепцій, що базуються на схожостях, подібностях, які можуть бути знайдені в наборі тренувальних даних.

На відміну від асоціативних правил і класифікаторів, в даному випадку типово

немає специфікованого результату або атрибуту, що прогнозується. Це означає, що процес пошуку кластерів слід розглядати як процес неконтрольованого навчання. В цьому випадку основна відмінність щодо контрольованого навчання полягає у відсутності цільового предикату, що розділяє інстанції тренувального набору на категорії. Відповідно, категорії формуються самим об'єктом навчання.

Задача кластеризації може бути формалізована таким чином.

Задано: множина некласифікованих інстанцій мови екземплярів.

Знайти: знайти множину концепцій, що покриває всі задані приклади, таким чином, що:

- 1) схожості між прикладами одних і тих же концепцій максимізуються;
- 2) схожості між прикладами відмінних концепцій мінімізуються.

Технології кластеризації є об'єктом активних досліджень в ряді галузей, таких як: статистика, розпізнавання шаблонів і машинне навчання [188-191]. Технологія видобування даних, в загальному, додає до засобів кластеризації ускладнення, а саме: використання великих наборів даних із багатьма атрибутами різних типів. Все це накладає особливі обчислювальні вимоги щодо використання релевантних алгоритмів кластеризації.

В ряді застосувань деякі форми кластеризації є визначальними. Основна перевага полягає в тому, що кластеризація об'єктів за подібностями відкриває нові можливості щодо їх застосування.

Дамо формальне визначення проблеми *видобування наборів частих входжень*.

Базові означення:

Означення 3.7. Множина входжень. Множиною входжень вважатимемо скінчену множину, що позначається через ES . Елементи ES позначатимемо через es, es', es_1, \dots і будемо посилатися на них, як на *елементи входжень*.

Означення 3.8. Множина елементів. Множина елементів розглядається як множина ES . Множини елементів будемо позначати через ES, ES', \dots .

Означення 3.9. n -множина елементів. Множину елементів ES будемо називати n -множиною елементів, якщо потужність $|ES| = n$, (де $|ES|$ позначає потужність ES).

Означення 3.10. Транзакція. Транзакцією CN_T будемо вважати деяку множину елементів і позначатимемо її через CN_T, CN'_T .

Означення 3.11. Набір даних. Набором даних $DataSet$ вважатимемо мультимножину транзакцій. Для заданого набору даних $DataSet$ максимальною довжиною транзакцій $DataSet$ будемо вважати:

$$l_{DataSet} = \max \{ |CN_T| \mid CN_T \in DataSet \}$$

Визначимо формально проблему видобування частих наборів даних.

Означення 3.12. Нехай $DataSet$ є набором даних і ES є набором входжень (елементів). Тоді ES будемо вважати частим набором входжень (множиною елементів) відносно $DataSet$ і з коефіцієнтом підтримки SF , де $0 < SF \leq 1$, якщо:

$$|\{CN_T \in DataSet \mid ES \subseteq CN_T\}| \geq SF |DataSet|$$

Нехай $nSet_n(SF, DataSet)$ є множиною всіх n -наборів (множин) елементів, що є частими входженнями відносно SF та $DataSet$. Тоді розв'язувана проблема означається як задача визначення $nSet_n(SF, DataSet)$ для кожного n , для якого $0 < n \leq l_{DataSet}$.

Ряд алгоритмів для пошуку частих входжень наборів елементів базується на порівневій генерації можливих кандидатів для алгоритму APRIORI. Порівневий підхід виконується з метою підтримки простору пошуку достатньо малим щоб помістити його в основну пам'ять. Така стратегія зумовлює необхідність кількох проходжень через кожен набір даних.

Літературні джерела представляють відмінні підходи для отримання сталої кількості проходжень через набір даних. Техніка розділення потребує двох проходжень через базу даних. На першому етапі набір даних розділяється на кілька частин, достатньо малих, щоб поміститися у виділений об'єм пам'яті. На наступному кроці кожне розділення уточнюється використанням порівневого алгоритму і тоді результати кожного розбиття зливаються. Така процедура дозволяє будувати надмножину рішень. На наступному кроці видаляються хибні додатні елементи з даної надмножини. Проте існує проблема що: якщо набір даних є достатньо великим, то тоді результуюче розбиття може бути надто малим відносно

нього. Це, своєю чергою, призведе до утворення надвеликої супермножини, що суттєво знижуватиме загальну ефективність алгоритму.

Іншим важливим підходом для зменшення кількості проходжень набором даних є підхід, що базується на оцінюванні малих випадкових зразках (вибірках) наборів уданих. Відповідно, генерується множина шаблонів, що є ймовірно частими в цілому наборі даних, і на наступному кроці верифікуються точне значення частоти їх входжень в залишку набору даних. Якщо трапляється збій в генерації кандидатних входжень, то забезпечується механізм, який у другому проходженні обчислює решту частих шаблонів. Відповідно, при зниженні порогу підтримки ймовірність збою (невдачі) може бути зменшена. Але для низьких значень ймовірностей це суттєво збільшуватиме кількість кандидатних входжень. Більше того, якщо ми маємо справу з великими наборами даних, то ймовірно, що мала вибірка не буде достатньо репрезентативною для всього набору даних, і це означатиме існування високої ймовірності невдачі. В даному випадку кандидатних входжень для верифікації в другому проходженні буде надто багато, щоб їх помістити в основну пам'ять. Це зумовить необхідність використання більше, ніж двох кроків. Виконано тестування відповідних алгоритмів, що показало високе споживання ресурсів пам'яті навіть для наборів даних середнього розміру. Для запропонованого покращення базових алгоритмів емпірично обґрунтовано зменшення кількості пам'яті, необхідної в очікувано кращому випадку. Відповідно, вимоги щодо пам'яті залежать від розміру набору даних. Цей факт розглядає дані алгоритми як практично не застосовні, якщо набори даних є достатньо великими (обширними). Додаткові підтвердження обмеження масштабованості для існуючих алгоритмів видобування частих наборів входжень було реалізовано кілька відомих алгоритмів і виконано їх незалежне тестування. Результати показали, що жоден із існуючих алгоритмів не є придатним до повноцінного масштабування за надвеликих наборів даних із надвеликою кількістю транзакцій.

Розробка алгоритму для видобування частих наборів даних при розріджених наборах даних вимагає невелику кількість пам'яті і зберігає невелику сталу кількість проходжень над масивами вхідних даних. Він дозволяє подолати проблему обробки

великої кількості транзакцій шляхом зменшення пошукового простору зі збереженням обґрунтованості.

Основною проблемою для дослідження є пошук наборів даних заданого розміру безпосередньо без генерації менших наборів даних, як це виконується в порівневих підходах. Таким чином, видобування всіх частих наборів даних дозволить зменшити використований ресурс пам'яті і підвищити часову ефективність. З іншого боку, виконання видобування максимального набору даних може не бути достатнім для побудови відповіді на запит, оскільки, фактично, максимальний набір елементів не дозволяє обчислити точне значення коефіцієнта підтримки вужчого (меншого) набору даних. Разом із тим, для надвеликого набору даних із великою кількістю малих транзакцій можливо обчислювати часті n_0 – набори даних, хоча більшість наявних алгоритмів не дають позитивного результату через надмірні вимоги щодо ресурсів пам'яті.

Перш за все слід показати, що проблема знаходження частих n -наборів даних можна перетворити в проблему пошуку частих символів над великим потоком символів над заданим алфавітом, відому як проблема агрегованих маркерів або проблема аналізу «гарячих» списків. Використання наявних на сьогодні алгоритмів для обробки проблеми агрегованих дозволить вирішувати початкову проблему частих наборів даних шляхом тільки двох послідовних проходжень над набором даних із додатковим підготовчим кроком, метою якого є обчислення певних статистичних значень над набором даних. Для вирішення поставленої задачі потрібно показати, що для розріджених наборів даних (тобто наборів даних з невеликою кількістю елементів на транзакцію відносно загальної кількості можливих елементів) необхідна кількість основної пам'яті, що вимагається в проектованому алгоритмі, є достатньо низькою і не залежить ні від кількості елементів, ні від розміру набору даних. Таким чином, при пошуку асоціативних правил при $n > n_0$ нам потрібно визначити часті набори елементів потужності n і n_0 . При використанні стандартних порівневих алгоритмів, таких як APRIORI, необхідно виконати до n_0 проходжень через набір даних. При використанні запропонованого підходу можна виконувати дві інстанціації алгоритму паралельно, що вимагає до

трех проходжень набором даних загалом. Таким чином, основне завдання полягатиме в розробці алгоритму, який для розріджених наборів даних вимагатиме обмеженої кількості пам'яті при збереженні невеликої сталої кількості проходжень над вхідним набором даних. Більше того, можна побудувати розширення даного алгоритму, що працюватиме над заданими потоками даних. В даному випадку основним результатом буде розробка алгоритму, який при малому споживанні пам'яті буде здатен видобувати часті n_0 - набори елементів над підмножинами потоку з розміром, пропорційним довжині зчитаного потоку.

Проектований підхід дозволяє обчислювати часті набори даних розміру n_0 безпосередньо, тобто без обчислення менших наборів даних, шляхом двох проходжень через набір даних. Необхідна кількість пам'яті є відомою наперед і є прийнятною за умов початкової гіпотези, що заданий набір даних є розріджений.

Запропоноване рішення базується на підході щодо рішення проблеми агрегованих маркерів, представленою в [192-193]. В цій роботі запропоновано простий алгоритм пошуку всіх запитів з частотою більшою чи рівною заданому порогового значенню із заданого потоку запитів (агреговані маркери), використанням пам'яті з заданою оцінкою та виконанням двох проходжень через потік. Очікувані результати базуються на методі звуження проблеми обчислення частих наборів даних до проблеми пошуку агрегованих-класу маркерів. Означимо формально проблему агрегованих маркерів (проблему аналізу «гарячих» списків).

Означення 3.13. Алфавіт. Позначатимемо деяку скінчену множину графічних маркерів через L^{Set} . Відповідно, L^{Set} буде позначатись через l, l', l_1 і посилатися на них, як на виділені графічні маркери.

Означення 3.14. Потік. *Потоком маркерів* будемо вважати послідовність $seq^1 = l_1, \dots, l_n$, для якої $l_i \in L^{Set}$ для кожного $1 \leq i \leq n$. Довжину потоку n будемо позначати як модуль seq^1 .

Означення 3.15 Частота. Для заданого потоку seq^1 і маркеру l через $freq(l^1)$ позначатимемо кількість входжень l в seq^1 .

Означення 3.16 (Проблема агрегованих маркерів). Нехай L^{Set} є множиною

маркерів, а seq^1 – потоком маркерів і num є дійсним числом, для якого $0 < num \leq 1$. Тоді проблему агрегації маркерів означимо як задачу визначення підмножини $L^{Set}(num, seq^1)$ як:

$$L^{Set}(num, seq^1) = \{l \in L^{Set} \mid |freq(l)| > num |seq^1|\}.$$

Якщо l належить до $L^{Set}(num, seq^1)$, то можна стверджувати, що l є видом агрегованого маркера відносно L^{Set} , seq^1 і num .

В даному контексті важливо відзначити, що в конкретному застосуванні проблем видобування частих наборів даних і агрегованих маркерів, вхідні множини $DataSet$ і seq^1 є, як правило, великими і можуть бути зчитані тільки послідовно, тобто відповідно до ідентифікаторів транзакції в першому випадку і до порядку послідовності – в другому.

Функціональність алгоритму наступна: для заданого потоку seq^1 і дійсного числа num , що називається пороговим значенням, алгоритм вимагає одного проходження через вхідний потік для пошуку надмножини шуканих $L^{Set}(num, seq^1)$. Може бути також виконане друге проходження для знаходження другого значення $L^{Set}(num, seq^1)$ зі збереженням тих же характеристик ефективності.

Основна ідея такого рішення полягає у прив'язці до кожного n_0 -набору елементів певного запиту і подальшої побудови відповідного потоку запитів $seq^1_{DataSet}$, починаючи з заданого набору даних $DataSet$, у такий спосіб, що проблема визначення $nSet_n(SF, DataSet)$ перетворюється в проблему визначення $L^{Set}(num, seq^1_{DataSet})$, де num є функцією від num , $l_{DataSet}$ і n_0 . Після означення такого перетворення можна використати довільний існуючий алгоритм для проблеми агрегованих маркерів для рішення вихідної проблеми видобування частих входжень наборів даних. Зокрема, можна використати алгоритми, які утримують кількість проходжень через набір даних настільки малою, наскільки це можливо.

Для заданої скінченної множини SEQ і натурального числа n_0 позначимо через SEQ^{n_0} множину всіх підмножин $L^{Set} \subseteq SEQ$ таких, що $|L^{Set}| = n_0$. Для двох заданих послідовностей $seq^1 = \langle x_1, \dots, x_n \rangle$ і $seq' = \langle x'_1, \dots, x'_m \rangle$ позначимо через $seq :: seq'$

послідовність $\langle x_1, \dots, x_n, x'_1, \dots, x'_m \rangle$. Для цього визначимо перетворення, яке для заданої проблеми видобування частих входжень наборів даних буде відповідну проблему агрегованих маркерів.

Означення 3.17. Нехай $iSet$ є набором елементів, а $DataSet$ – набором даних і n_0 – натуральним числом, для якого $n_0 \leq I_{DataSet}$. Тоді:

1) алфавіт L_{iSet}^{Set} визначається як множина $iSet^{n_0}$, причому кожна множина в $iSet^{n_0}$ є символом алфавіту;

2) для кожного $CN_T \in DataSet$ потік асоційований з CN_T , є послідовністю $seq^1_{CN_T} = \langle iSet_1, \dots, iSet_{n_{CN_T}} \rangle$, для якої $\{iSet_1, \dots, iSet_{n_{CN_T}}\} = CN_T^{n_0}$, і кожне $iSet_j \in CN_T^{n_0}$ трапляється в $seq^1_{CN_T}$ точно один раз;

3) якщо $DataSet = \{CN_{T_1}, \dots, CN_{T_n}\}$, тоді потік $seq^1_{DataSet}$, асоційований з $DataSet$, є послідовністю:

$$seq^1 = seq^1_{CN_{T_1}} :: seq^1_{CN_{T_2}} :: \dots :: seq^1_{CN_{T_n}}$$

Слід відзначити, що в наведеному означенні не визначається потік, асоційований із транзакціями. Це ж стосується потоку, асоційованого з набором даних $DataSet$. Дійсно, для заданої транзакції CN_{T_i} існує багато способів побудови потоку $seq^1_{CN_{T_i}}$, що відповідає йому, і може існувати багато способів побудови $seq^1_{DataSet}$. Тому вибір $seq^1_{DataSet}$ є не релевантним з точки зору коректності такого перетворення.

Покажемо, що проблема видобування частих наборів елементів $nSet_n(SF, DataSet)$ може бути відображена в проблему агрегованих маркерів $L^{Set}(num, seq^1_{DataSet})$, де $seq^1_{DataSet}$ є довільним потоком, побудованим згідно попереднього означення, і num є відповідною функцією для SF , n_0 та $I_{DataSet}$.

Твердження 3.3. Нехай $iSet$ – набір елементів, $DataSet$ – набір даних, n_0 – натуральне число, для якого $n_0 \leq I_{DataSet}$, SF – дійсне число, для якого $0 < SF \leq 1$. Нехай $L^{Set^{iSet}}$ та $seq^1_{DataSet}$ є, відповідно, алфавітом і послідовністю маркерів згідно попереднього означення. Нехай також $\overline{Solve}^{Solution} = (I_{DataSet}, n_0)$.

Якщо набір елементів $iSet$ є частим n_0 -набором елементів відносно SF і $DataSet$, тоді $iSet$ є агрегованим маркером відносно $L^{Set^{iSet}}$, $seq^1_{DataSet}$ і $num = SF / \overline{Solve^{Solution}}$. Тоді:

$$iSet \in nSet_n(SF, DataSet) \Rightarrow iSet \in L^{Set}(num, seq^1_{DataSet})$$

Нехай $|DataSet| = k$ та $DataSet = \{CN_{T_1}, \dots, CN_{T_n}\}$ матимемо:

$$1) |seq^1_{DataSet}| = \sum_{i=1}^k \binom{|CN_{T_i}|}{n_0} \leq \sum_{i=1}^k \binom{i_{DataSet}}{n_0} = \overline{Solve^{Solution}}, k;$$

2) згідно способу побудови

$$seq^1_{DataSet} \text{ матимемо } |\{CN_T \in DataSet \mid iSet \subseteq CN_T\}| = freq_{DataSet}(iSet).$$

Згідно означення $nSet_n(SF, DataSet)$ матимемо:

$$iSet \in nSet_n(SF, DataSet).$$

Згідно наведених означень:

$$|\{CN_T \in DataSet \mid iSet \subseteq CN_T\}| = (CF, k).$$

Згідно алгебраїчного способу побудови матимемо, що:

$$freq_{seq_{DataSet}}(iSet) \geq (CF, k).$$

Згідно означення для num матимемо:

$$freq_{DataSet}(iSet) \geq CF / \overline{Solve^{Solution}} \geq \left\lfloor k, \overline{Solve^{Solution}} \right\rfloor$$

Згідно означення для $L^{Set}(num, seq^1_{DataSet})$ матимемо:

$$freq_{DataSet}(iSet) \geq num |seq^1_{DataSet}|,$$

$$iSet \in L^{Set}(num, seq^1_{DataSet}).$$

Це твердження дозволяє отримати нову техніку знаходження частих n_0 -наборів елементів, що може бути описана таким чином:

1) побудувати потік $seq^1_{DataSet}$, що відповідає $DataSet$;

2) знайти множину агрегованих маркерів $L^{Set}(num, seq^1_{DataSet})$.

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Показано, яким чином абдуктивні міркування можна використовувати в контексті задач класифікації і для підвищення загальної ефективності, якщо система працює з неповними даними та зовнішніми доменними знаннями.

2. Запропонований підхід можна досконалити поєднанням відмінних парадигм видобування даних, таких як: класифікація, асоціативні правила і кластеризація, – шляхом використання абдуктивного фреймворку з обмеженнями.

3. Отримані результати показують, що абдуктивний фреймворк може розглядатися як деякий шар, здатний приймати знання від користувача і відповідні запити мови програмування високого рівня, після чого обчислювати рішення. Вихідною точкою дослідження є те, що поєднання видобування даних з фреймворком логічного програмування забезпечуватиме нас можливістю використання доменних знань у процесі видобування знань і їх використання, що разом дозволить отримати результати для покращення якості дерева класифікації, якщо робота з ним виконується на більш складному рівні, ніж рівень простого обходу, а загальне фреймворк-застосування відповідно базується на абдуктивному підході.

4. Базова гіпотеза полягає в твердженні, що застосування класифікаційного дерева до нових спостережень може розглядатися як спосіб прямого абдукційного обчислення, як тільки дерево представляється колекцією правил очевидним чином. Все це відкриває нові можливості, оскільки абдуктивні обчислення можуть приймати досить складну форму, зокрема вони можуть приймати до розгляду кілька типів знань одночасно.

5. В даному контексті видобування реляційних даних, що є видобуванням даних в описовій мові першого порядку, може розглядатися як ефективне поєднання логічного програмування і відносно невеликого пошукового простору завдяки використанню ймовірнісних значень. Таким чином, використання індуктивного ймовірнісного фреймворку може бути більш ефективним для доменних знань, описаних засобами, відмінними від логіки першого порядку.

6. Побудовані представлення містять асоціативні правила, як обмеження цілісності всередині фреймвоку і як правила бази знань для абдуктивних доведень. Звичайний фреймворк в аналізі на основі асоціативних правил для визначення коефіцієнта підтримки або коефіцієнта довіри дає хороші результати не в усіх випадках, якщо не виконано точну установку порогових значень.

7. В запропонованому підході очікуваний результат полягає у використанні відмінних метрик, що дозволятимуть встановлювати точну кореляцію серед атрибутів. Такий вид кореляції є важливим в області абдуктивних міркувань, оскільки вони можуть краще виразити залежності між каузальними подіями й ефектами.

РОЗДІЛ 4

РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛІМЕДІЙНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ У ПРОЦЕСІ БУРІННЯ

4.1 Оцінка ефективності імплементації мультимедійного контенту на рівні оболонки експертної системи

Базовим елементом при формуванні експертної навчальної системи слід вважати специфіку представлення експертних даних у контексті аналізованого мультимедійного представлення на базі спеціалізованого програмного й апаратного рішення. Саме проєктоване мультимедійне представлення потребує детального структурного аналізу. Це досягається застосуванням інтелімедійних включень на основі метатегів у мультимедійний інтерфейс програмованого рішення. Також передбачено обов'язковість інтеграції з іншими програмними реалізаціями (наприклад, експертною системою), не обмежуючи особливості поєднання з іншими експертними системами, які не впливають на складність інтерфейсу [194-199].

Особливістю якісного мультимедійного представлення вважаємо, по-перше, мультимедійні дані, що ґрунтуються на експертних знаннях, та необхідність спроектованого технічного забезпечення для ефективної інтеграції інтелімедійних даних у навчальну експертну систему. Також особливістю мультимедійного представлення визначено важливість інтеграції анімаційних елементів і графічних представлень у конкретний навчальний процес.

Представлене програмне рішення порівнює ІМАС – підхід із традиційним програмуванням, у якому один скрипт контролює всі властивості базової програми. Автор стверджує, що підхід «скрипт-майстер» добре працює, коли програма є спрощеною за своєю природою, але дуже швидко ускладнюється, при спробі виконання інтеграції методів гіпермедіа. «Скрипт-майстер» підходить для програми, виконання якої здійснюється наперед визначеним лінійним шляхом. Програма, прототип інтелімедійної інформаційної системи підтримки прийняття

рішень у процесі буріння, є легшою у використанні при застосуванні методів гіпермедіа, тобто вона виконується нелінійним шляхом, а залишається за користувачем.

«ToolBook» і «Microsoft Visual Basic» – інструменти в середовищі Microsoft Windows. Ці пакети використовують більш інтуїтивний підхід до програмування мультимедійної аплікації, такої як прототип системи підтримки прийняття рішень у процесі буріння. Обидва ці пакети використовують підхід об'єктно орієнтованого типу для розробки системи. Цей підхід дозволяє використання гіпермедіа, дозволяючи об'єктам відправляти й отримувати повідомлення.

Таким чином, жоден «скрипт-майстер» не повинен знати шлях, який кожен окремий користувач приймає в системі. Це може бути покладено на користувача, хоча програміст може ввести обмеження, якщо необхідно. Гнучкість, що забезпечується подієво-керованою програмою, спрощує завдання програмування для нелінійних програм.

Для того, щоб це оцінити, в дослідженні розроблено таку ж програму з використанням усіх вищезазначених мультимедійних авторських пакетів, крім Visual Basic. Метою дослідження було порівняти ці пакети за різними спільними рисами. Оцінка кожного середовища полягала у вивченні їх можливостей авторизації, а також можливостей, які стосуються якості кінцевого продукту, що отримує користувач. Автор виявив, що середовище «ToolBook» є гнучким тому, що воно забезпечує спрощення ярликів для новачків, щоб розробити прості системи. Варто зазначити, це досить складно і важче для вивчення при проектуванні складної мультимедійної системи. Після регульованого порівняння пакетів дослідник дійшов до висновку, що «Toolbook» є, безсумнівно, найпотужнішим і найгнучкішим із оцінюваних пакетів.

При виборі мультимедійного програмного забезпечення відіграють роль ряд факторів. Доречним є вибір відомого середовища, щоб зосередити навчальний процес на компонентах системи, а саме: на експертній системі та мультимедійному інтерфейсі. Вдається дослідити більш детально ті питання, які стосуються комбінації двох технологій, не витрачаючи значні зусилля для вивчення нового

середовища розробки. Варіанти, доступні для програмного забезпечення експертних систем, можуть слугувати тільки для перевірки такого рішення.

Visual Basic, у порівнянні з ToolBook, не має формальної оцінки його можливостей (особливо, мультимедійних). Тому використання ToolBook є преференційним.

Як зазначено в дослідженні процес здійснення розробки експертної системи вважають непростим завданням. Дослідники порівнюють зазначений вище процес із розробкою пакета обробки тексту або електронної таблиці. Мета дослідження полягає у конкретизації проблеми поєднання робочих середовищ – експертну систему та мультимедіа. Сам процес дослідження є більш ефективним при створенні інтелектуальної системи, що використовує представлення знань у формі мультимедійного контенту для отримання можливості звернення суб'єкта навчання до поточних актуальних питань самого навчального процесу.

Найпростішим варіантом розв'язку задачі розробки інтелектуальної системи, що використовує представлення знань у формі мультимедійного контенту, є використання існуючої класичної оболонки експертної системи. Визначено питання для відбору оболонки класичної експертної системи. Дослідження авторів було використано як основа для подальшої розробки методу оцінки наявних оболонок експертних систем. Припустимо, що вибір оболонки експертної системи базується на чотирьох етапах:

1. Визначення необхідних функцій і можливостей.
2. Виявлення потенційних оболонок.
3. Оцінка оболонок на основі необхідних функцій і можливостей.
4. Вибір відповідної оболонки.

У процесі дослідження визначено декілька потенційних можливостей і функцій аналізованої системи-прототипу. Ці характеристики представлено у вигляді ключових визначень, поданих на рисунку 4.1.

Характеристики оболонки експертної системи: робота з «Microsoft Windows», з'єднання з «Esta/Stress», ланцюг прямого логічного висновку, мова програмування, засоби обґрунтування, технічна підтримка, документація та підручники, збереження

виконаних операцій, підтримка істинності, вирішення суперечностей, управління невизначеністю, швидкість розробки прототипу.

Необхідно зазначити, що всі оболонки експертних систем працюють на персональних комп'ютерах. Таким чином, однією з базових вимог дорозглянутих оболонок була доступність для використання на обраному персональному комп'ютері. Список потенційних характеристик оболонок і їх відображення в реальних системах представлено на рисунку 4.1:

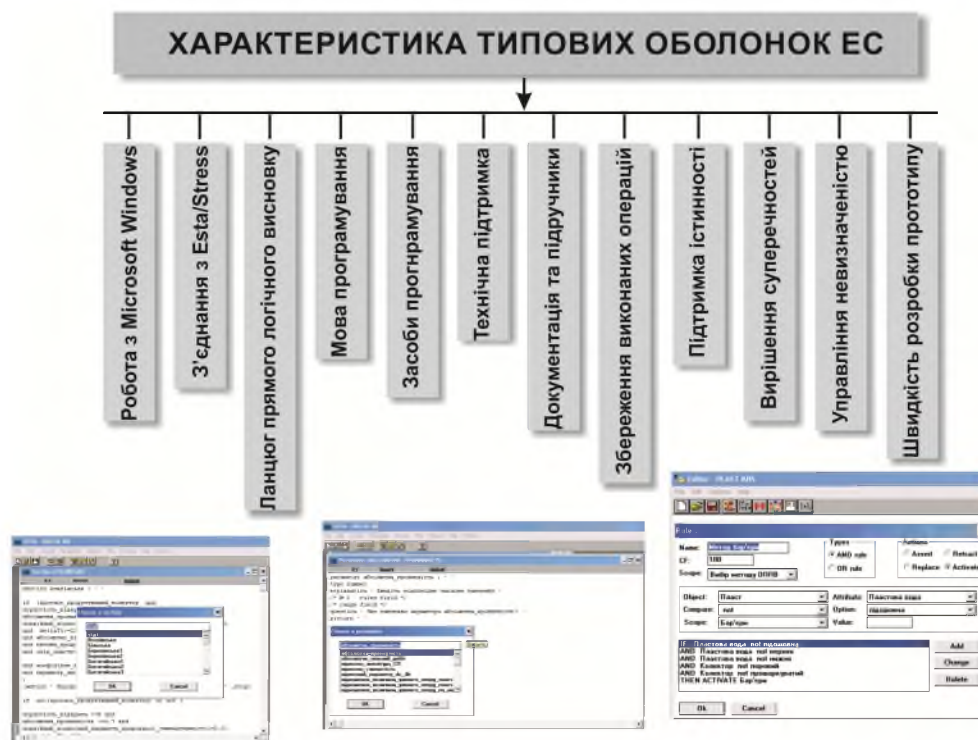


Рисунок 4.1 – Характеристика типових оболонок ЕС

Оцінка оболонок складається з двох етапів. Перші три характеристики, представлені на рисунку 4.1, передбачають мінімальні вимоги, які слід вважати необхідними. Априорі, експертна система повинна відповідати зазначеним вимогам, тому будь-яка оболонка експертної системи, яка не відповідає цим критеріям, не розглядалась.

Найбільш важливою з передбачених вимог є можливість з'єднати вибрану оболонку експертної системи з «Esta/Stress» у середовищі «Microsoft Windows», оскільки вона була обрана як програмне забезпечення для розробки інтелектуальної системи-прототипу з використанням мультимедійного інтерфейсу.

Потреба в методі прямого ланцюга логічного висновку (одна з типових характеристик ЕС) пов'язана з характером інтелектуальної системи-прототипу. Цей тип інтелектуальної системи полягає у формуванні висновків на основі обмеженого числа визначених даних. Припускалося, що в інтелектуальній системі-прототипі може бути деяке число «рішень» на основі конкретних характеристик окремих параметрів. Тому використання прямого процесу ланцюга міркувань – основний критерій формування проектованої інтелектуальної системи в процесі зворотного логічного висновку.

Вибір аналізованих оболонок для створення експертних систем базувався на аналізі перших трьох зазначених критеріїв. Для подальшого аналізу та порівняння вибрано оболонки «Esta/Stress». Другий етап оцінки оболонок експертних систем базувався на порівнянні другорядних характеристик цих оболонок для подальшого визначення сумісності з проблемною галуззю.

Оцінка мови щодо програмування кожної оболонки здійснювалась на основі характеристик «легкості розуміння» та «гнучкості». Функція гнучкості визначалася здатністю додавати нові можливості до оболонки експертної системи або змінювати поточні для реалізації в конкретному рішенні. З наявних оболонок експертних систем використано програмне рішення «Esta/Stress».

Засоби обґрунтування щодо використання оболонки експертної системи дозволяли реалізувати інтелектуальну систему в конкретному інформаційному продукті. Ця характеристика розглядалася з точки зору гнучкості. Інші характеристики, містять засоби обґрунтування, тобто вони є менш гнучкими, тому що не піддаються програмуванню. Програмований засіб обґрунтування, властивий «Esta/Stress», забезпечує розробника інтелектуальної системи гнучкістю представлення знань.

Важливим є вибір технічної підтримки для використовуваного програмного забезпечення. Технічна підтримка реалізується у вигляді експерта, ознайомленого з продуктом, при виникненні проблеми з його використанням. У нашому випадку оболонка «Esta/Stress» забезпечує повну програмну та технічну підтримку реалізації

інформаційного продукту. Інші види оболонок експертних систем є менш придатними в досліджуваній галузі.

Важливим елементом технічної підтримки для використовуваного програмного забезпечення в контексті проекрованої інтелектуальної системи є повний доступ до відповідного документаційного супроводу та посібників із аналізованої проблеми.

Зручність використання інтелектуальної системи полягає в забезпеченні простоти використання та застосування елементів зміни таким чином, щоб суб'єкт діалогу зміг зупинитись на середині процесу і мати можливість відновити сеанс, зберігаючи всі поточні зміни. Це стосується об'ємних складних систем, які передбачають введення експертом великих обсягів даних.

Ще одним важливим аспектом експертної системи є її адаптивність – здатність до змін у конкретних даних. Інша характеристика однієї з типових оболонок ЕС – підтримка істинності. У цьому випадку експертна система здатна розпізнавати недостовірність окремих даних у контексті їх зміни. Характеристика здатності підтримки істинності пов'язана з можливістю виконувати аналіз параметрів, змінюючи деякі дані для визначення їх впливу на зроблені висновки. Оболонка «Esta/Stress» є оптимальною в конкретному випадку, оскільки здатна забезпечувати підтримку істинності.

Інша характеристика оболонки ЕС – вирішення суперечностей – є також важливою для експертної системи. Процес вирішення побудований на принципі використання конкретного пріоритету відповідно до правил, реалізованих у базі знань. Якщо задоволено умови двох або більше правил, тоді експертна система визначає послідовність застосування першочерговості. Також, «Esta/Stress» є оптимальною оболонкою, яка забезпечує фіксоване вирішення суперечностей. Користувач за необхідності може визначити конкретні пріоритети для кожного правила. Також одночасно передбачене виконання двох і більше правил. Система володіє внутрішньою методикою вибору одного правила й можливістю постійного продовження роботи.

Однією з базових концепцій і можливостей експертних систем (як «Esta/Stress») є їх здатність оперувати з невизначеністю. Експертна система може керувати невизначеністю, дозволяючи користувачеві формувати запитання для можливості надати оцінку та переконатися, що вони дійсно належать цій системі. Висновки, сформовані експертною системою, також можуть бути невизначеними; належна оболонка експертної системи буде забезпечувати конкретні засоби, що включатимуть заходи невизначеності в цих висновках. Ряд систем (наприклад, система «Esta/Stress») здійснюють керування фактами та правилами невизначеності.

Ще одна характеристика оболонки ЕС – простота й легкість розробки системи-прототипу. Це важливо, адже період розвитку для системи-прототипу обмежений. Було виявлено, що прототип однієї системи зробити швидше і легше, ніж прототипи інших систем.

Було встановлено, що програмні продукти «Esta/Stress» є більш оптимальними та доступними оболонками експертних систем. Хоча слід зазначити, що ці оболонки, на відміну від інших аналізованих, не є оптимальними для швидкого прототипування, хоча система є більш гнучкою і більш потужною в порівнянні з іншими доступними оболонками. «Esta/Stress» пропонують потужну макромову програмування, що доповнюється можливістю вставляти код з інших мов програмування високого рівня та представленням гнучкого механізму обґрунтування.

Програмний продукт «Esta/Stress» забезпечує повну підтримку процесу прямого ланцюга логічного висновку. Інші оболонки надають слабку підтримку прямого логічного висновку, що може лише ускладнити інтелектуальну систему в контексті регулювання процесу логічного висновку для задоволення потреби конкретного проекту.

Вибір програмного та апаратного забезпечення було здійснено на основі підходу, описаного вище. Було визначено, що «Esta/Stress» буде найкращою для програмування інтелектуальної системи з мультимедійним контентом. Ці пакети програмного забезпечення доступні в середовищі «Microsoft», обраного для апаратного забезпечення. Це програмне забезпечення та апаратні засоби, що

використовуватимуться для розвитку прототипу інтелектуальної системи представлення знань в формі мультимедійного контенту, є базованими на знаннях, тобто системами класу *Knowledge Based Intelligent System with Multimedia Content* [200-205].

4.2 Розробка структури та реалізація прототипу інтелектуальної системи як версії інтелектуальної системи з мультимедійним інтерфейсом

Прототип системи був розроблений і реалізований для більш детального вивчення проблеми, пов'язаної зі створенням інтелектуальної системи з мультимедійним контентом. Визначено проблемну галузь – розробка інтелектуальної системи представлення знань у формі мультимедійного контенту. Заснована на знаннях інтелектуальна система представлення знань у формі мультимедійного контенту відображає основні складові вибраного технологічного процесу таким чином, що суб'єкт навчання, який реалізує свої знання на основі запропонованого рішення, може отримати пояснення та додатковий ТЕХНОЛОГІЧНИЙ контент, який його цікавить, у контексті логічного висновку відповідно поставленої задачі. Відповівши на кілька запитань та здійснивши певну кількість хибних відповідей, суб'єкт тренінгу отримує можливість повторно закріпити матеріал суто з визначеної проблеми, а також забезпечення текстового та мультимедійного супроводу. Розробка цього прототипу пропонує засоби для визначення питань, пов'язаних з інтеграцією мультимедійного контенту та можливостей експертних систем.

Вважається, що проблемна галузь повинна бути вузькою і чітко визначеною. База знань з відповідної системи має бути обмежена до менш, ніж 100 правил, щоб проблеми залишалися керованими. Крім того, проблемна галузь повинна бути стабільною протягом тривалого часу. І, нарешті, необхідні процеси, доступні для тестування корисності системи. Технологічний процес буріння нафтогазових свердловин є відносно стійкою галуззю знань, яку важко описати, використовуючи тільки текстові представлення. Ця галузь дуже конкретна і чітко визначена за своїм

призначенням. Хоча в розвитку ефективної програми підтримки прийняття рішень у процесі буріння свердловин відіграють роль багато змінних, предметна галузь може бути представлена у вигляді експертної системи з необхідною кількістю правил.

Базу знань для прототипу було складено з кількох джерел для забезпечення правильності системи. Вони містили інтерв'ю з експертами процесу буріння (операторами технологічного процесу), а також огляд фахової літератури з цієї тематики. Було побудовано ряд емпіричних правил та їх шаблонів, які допомагають у розробці системи підтримки прийняття рішень в процесі буріння свердловин класу ІМАС. Було запропоновано ряд ідей, як краще відтворити дані для того, щоб переконатися, що все зрозуміло й застосовується належним чином.

Під час обговорень з експертами було з'ясовано, що види фактичної інформації будуть мати важливе значення для розробки ефективної бази знань системи. До них належать дані про бурове обладнання, види технологічних операцій, а також основні принципи вибору режимів буріння. Експертна система також містить практичні правила, які ілюструють процес прийняття рішень.

Найбільш важливі правила цієї системи містять дані з фахових підручників, що описують технологічний процес буріння свердловин на нафту і газ. Знання, виділені з цих джерел, дозволяють імітувати власний процес міркувань при прийнятті рішень. Вибрані засоби розробки оболонки експертних систем містять унікальні методи зберігання мультимедійних даних з можливістю їх зіставлення. Технологічно система зберігає дані про об'єкти в структурах, що описують означуваними схемами (*Definitions Schemas*). У даному випадку це забезпечує зручний спосіб зберігання та вилучення даних. Типи об'єктів цієї системи мають справу з буровим обладнанням, технологічними операціями та режимами буріння. Ці схеми забезпечують способи визначення характеристик об'єктів, в тому числі і відносно інших об'єктів за допомогою технології комірок (семантичний аналог *фреймових слотів*). Кожна комірка складається з однієї або більше ознак чи відношень, які описують поточний нафтогазовий об'єкт. Система зберігає також інформацію у вигляді фактів. Відповідно до класичного розуміння концепцій інтелектуальних систем у кожен момент часу відомі факти відіграють відповідно

основну роль у побудові процесу міркувань (логічного висновку), вказуючи тим самим, які правила застосовуються в кожній контекстній ситуації. Система реагує на зміну в фактах, що містяться в робочій пам'яті для визначення, чи застосовуються певні правила. Після того, як системі стало відомо про факти, вона знає, що потрібно виконати певні дії. Якщо умови даного правила задоволені відомими фактами, правила включають до списку основних операцій, які повинні бути завершені. Якщо у правилах цього не вказано, то порядок виконання завдань не є суттєвим.

Вважається, що кожне правило має *еквівалентну доменну черговість*, і в рамках домену вони виконуються в довільному порядку. Коли правило спрацьовує (тобто, *антецедент* істинний), то це означає виконання дій у правій частині правила (*консеквент*). Коли певний фрагмент програми виконується, то він виходить із розряду фактів і потім здійснюється виклад функцій початкової програми, що насправді виконує всі дії, необхідні для розгортання імплементації. В той час, як база фактів змінюється, система реагує на виконання необхідних дій для зберігання контролю над ситуацією, визначеною правилами.

Після створення функціональної бази знань необхідно забезпечити відображення інформації у відповідній формі з використанням засобів мультимедійного інтерфейсу. Для відображення графічної інформації такий інтерфейс забезпечує зручні для користувача методи створення екранних форм з усіма стандартними протоколами «Windows». Відповідно, кінцевий інтерфейс користувача виглядає і працює так, як і інші програми операційної системи «Windows», що сприяє його легкому опануванню.

Для опису функціональності графічного інтерфейсу його розділено на три основних частини, що вирішують такі задачі: опис бурового обладнання, опис технологічних операцій у рамках загального технологічного процесу буріння, пояснення особливостей утримання вибраного режиму буріння. Кожна з цих функцій буде детально описана нижче.

Основою для всіх автоматизованих та інтелектуальних систем, що застосовуються у процесі буріння, є детальний опис роботи конкретного бурового

обладнання, експлуатація якого забезпечує досягнення певної мети (наприклад, заданої швидкості проходження, заданого рівня зношення долота).

Таким чином, основне питання полягає в тому, яким чином представляти дані про бурове обладнання так, щоб це було корисно з практичної точки зору. Для цього розглянуто кілька основних аспектів. Очевидно, що такі дані найбільш ефективно зображаються у вигляді растрових картинок в такий спосіб, щоб на цих картинках були позначені всі елементи бурового обладнання, відповідно до переліку. Отримання даних цим шляхом може містити певні спотворення кінцевого цифрового формату (це стосується більшості цифрових форматів, а саме: «.bitmap», «.tiff», «.psx»). Крім того, літературні джерела завжди характеризуються надмірною деталізацією описів що ускладнюють процес параметризації представлення на основі системи ключових маркерів з одного боку, і наявними надміру абстрактними узагальненнями, що не дозволяють параметризацію взагалі, – з іншого.

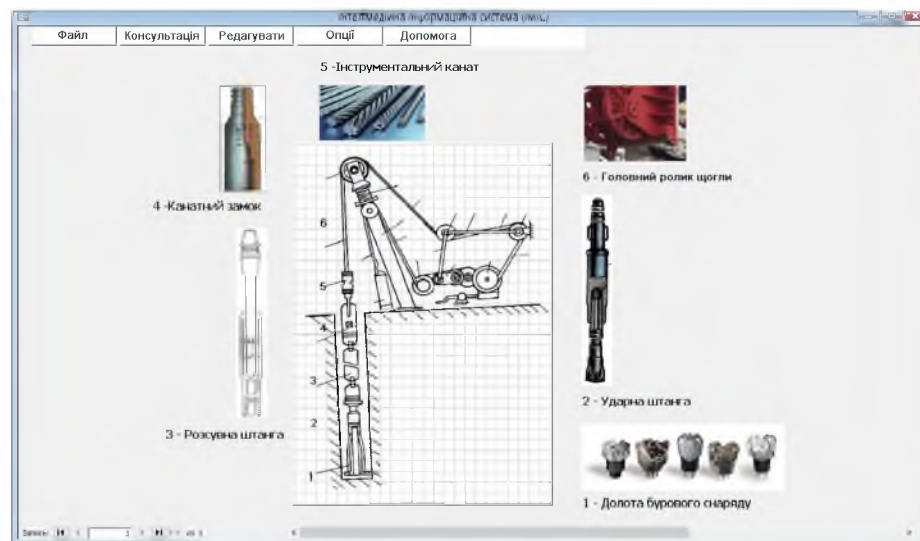


Рисунок 4.2 – Схематичне зображення базового бурового обладнання процесу буріння з введеними ключовими маркерами першого рівня (мінімальний рівень деталізації)

Для пояснення кожного елемента обладнання введено відповідні підписи. Для більш чіткого пояснення розташування відповідних елементів обладнання передбачене їх введення відповідними геометричними формами. Це дає можливість користувачу отримувати більш детальну інформацію саме про цей елемент

обладнання, натиснувши на його назві чи самому елементі обладнання. Після того, як елемент бурового обладнання обрано, він виділяється червоним кольором і з'являється вікно інформації, яке пояснює точне місце розташування та функцію вибраного елемента обладнання. Це вікно також відображає список технологічних операцій, у яких задіяний даний елемент обладнання. Якщо користувач хоче побачити опис певної технологічної операції, її можна обрати зі спадаючого списку. Після вибору технологічної операції з'являється повне (нове) вікно з інформацією про неї. Вікно технологічної операції передбачене для пояснення операції (отримане від фахівців-експертів), щоб користувач (студент, діючий оператор технологічного процесу) зміг виконати її безпечно, коректно та правильно. Інформація на цій екранній формі розбита на відповідні розміри з письмовим описом усіх необхідних факторів для належного виконання технологічної операції. Крім того, якщо в технологічній операції задіяно ряд елементів, то система відобразатиме їх зображення та список. Користувач може переходити між технологічними операціями натисканням відповідних кнопок на екранній формі. Є також кнопки, при натисканні яких буде відображатися відповідний відео - або аудіозапис.

Media Control Interface (MCI) є стандартним компонентом «Microsoft Windows». Він надає набір команд, які дозволяють управляти мультимедійними пристроями. Мультимедійні пристрої можуть бути двох типів: прості та складні. Прості пристрої – це зовнішні апаратні пристрої, що працюють без використання файлів, наприклад, аудіо та відеопрогравачі дисків. Складні пристрої відтворюють файли і вимагають, щоби було вказано ім'я файла (йдеться про аудіопристрої та пристрої накладання зображень, що дозволяють відображати растрові зображення, анімації та цифрове відео). Система надає кілька «mci»-команд для доступу до «MCI»-сумісних мультимедійних пристроїв.

Іншою функцією, що забезпечує прототип системи, є здатність налаштовувати систему під конкретного користувача або групи користувачів. Після натиснення відповідної кнопки користувач може зніціювати процес проектування. Система представляє список цілей користувача і просить обрати одну з них. В наступному кроці система використовує обрану мету разом із основною інформацією, яку

користувач зазначив раніше (наприклад, збурюючі фактори процесу буріння в конкретному нафтогазоносному регіоні).

Відповідна інформація містить особливості роботи бурового обладнання, специфіку виконання технологічних операцій і особливості утримання заданих режимів буріння. Якщо користувач не розуміє введених умов маркування, він може просто натиснути на них для отримання визначення, оскільки ці поля мають гіперпосилання (аналогічні гіперпосилання існують для опису технологічних операцій). Крім того, в будь-який час користувач може вибрати довідки меню «Довідка», для перегляду опису та тлумачення різних термінів, які йому ще незнайомі чи в яких він не впевнений тощо. Меню «Файл» забезпечує друк програми, хоча це є неважливо для поточної інформації, що відображається у вікні. Тобто роздрукувати можна тільки растровий рисунок екранної форми. Ця екранна форма також містить посилання на діалогове вікно технологічної операції таке ж, як і екранна форма бурового обладнання. На екрані програми користувач також має можливість натиснути кнопку типу «More info...», для отримання списку необхідних тем, що детально пояснюють як реалізацію відповідної технологічної операції (можна виводити тьюторіали, конспекти лекцій тощо). Ці теми містять інформацію про основні параметри буріння (контрольовані, неконтрольовані, збурюючі, вихідні).

Експертна система і мультимедійний інтерфейс прототипу пов'язуються через «Dynamic Link Library» (DLL). Створений модуль «DLL» містить всі функціональні можливості, пов'язуючи існуючі бібліотеки, які підтримуються за допомогою об'єктно-орієнтованих функцій, спеціально розроблених для інтелімедійної інформаційної системи, що базується на знаннях. Вибраний інструмент розробки надає спосіб для зв'язку з будь-яким «Windows DLL». Тому ми гарантовано отримуємо типи даних, сумісні з типами даних «Windows», а також відповідні зв'язки між ними. Крім того, при такому підході ми отримуємо доступ до глобальної пам'яті Windows, що забезпечує використання гарантованого обміну інформації [206-210].

Таким чином, існуюча створена експертна система може бути викликана з будь-якого інтерфейсу, що має можливість зв'язуватись з будь-яким «DLL». Це робить створений додаток портативним на рівні «Windows»-згенерованих інтерфейсів.

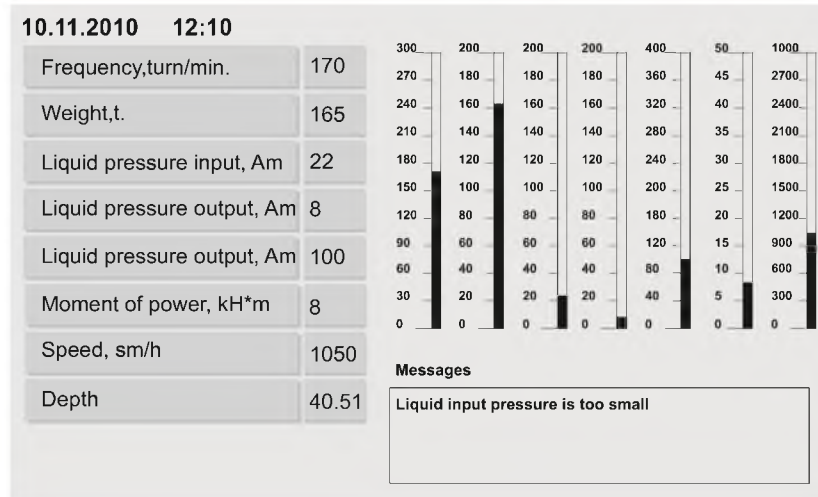


Рисунок 4.3 – Системна емуляція пульта бурильника

Вибраний інструмент розробки дозволяє зв'язувати з «DLL» і наступний виклик функцій для завантажування, а також доступ до бази знань «ІМАС». Хоча цей процес є простим, виникає кілька проблем через складність роботи в багатозначному середовищі. Після здійснення виклику всіх потрібних функцій система почала видавати фатальні помилки через загальні недоліки захисту на регулярній основі. Це пов'язане з управлінням пам'ятю. В такому випадку треба використовувати утиліти типу «MemMaker» для оптимізації ефективного використання пам'яті та максимізації розміру файлу, що запускається.

4.3 Оцінка ефективності отриманого прототипу при підтримці прийняття рішень у процесі буріння

Після того, як прототип було розроблено, розглянуто ряд питань, пов'язаних з ефективністю підходу мультимедійних експертних систем у цілому. Пропонований прототип було розроблено відносно легким для того, щоб приховати його мультимедійні аспекти (такі як растрові зображення, відео- та аудіофайли). Це

дозволяє створити дві версії системи підтримки прийняття рішень у процесі буріння, що базується на знаннях, одна з яких працює з мультимедійним інтерфейсом, а інша – без. Ці дві версії можна випробувати й порівняти, щоб чітко визначити переваги, які дає включення мультимедійних можливостей в експертну систему.

Для оцінки двох версій системи підтримки прийняття рішень у процесі буріння, що базується на знаннях, певна кількість суб'єктів тренінгу мала випробувати системи. А саме: всім суб'єктам запропонували знайти відповідну інформацію про бурове обладнання для того, щоби отримати загальне уявлення про місце і функції окремих елементів обладнання й особливості виконання відповідних технологічних операцій. Всі учасники тесту вважаються знайомими з середовищем Microsoft Windows, технологічною основою його інтерфейсу й основними прикладними програмами. Проте у предметній галузі буріння учасники тесту мають невеликий досвід прийняття рішень, тобто, в цілому, вважаються новачками.

Кожного суб'єкта попросили виконати запит на інформацію про два різних елементи бурового обладнання і дві технологічні операції, з якими ці елементи пов'язані. Крім того, суб'єкти отримали завдання розробити послідовність технологічних операцій для заданого режиму буріння і вибрати окремо дві технологічні операції для додаткового вивчення.

Початкова складність тесту полягає в тому, що дві версії розроблених прототипів були дуже схожими, що унеможливило зіткнення початкових дій між їх використанням. Для уникнення цієї ситуації половина суб'єктів тестувала спершу версію без мультимедійного інтерфейсу, а інша половина почала саме з версії з мультимедійним інтерфейсом. Тестування проводилось, щоб визначити: чи можуть ці навчальні дії спотворити результат. Після завершення консультацій за однією з версій суб'єктам поставили ряд запитань про продуктивність саме цієї версії. Схожі запитання поставили після апробації іншої. Набір поставлених запитань подано в **таблиці 1**. Відповіді на ці запитання використали як основу для побудови рейтингу кожної з систем та їх порівняння в цілому. Потім результати проаналізували для отримання більш змістовних даних, які, в кінцевому підсумку, використали для побудови висновків.

Таблиця 4.1 – Запитання для визначення ефективності ІМАС

Огляд ефективності мультимедіа в інтелімедійній системі, побудованій на знаннях	
Версія без мультимедійного інтерфейсу	Версія з мультимедійним інтерфейсом
1. Чи легка система навігації?	1. Чи легка система навігації?
2. Чи є зрозумілим те, як здійснювати запит на конкретний елемент бурового обладнання?	2. Чи є зрозумілим те, як здійснювати запит по конкретному елементу бурового обладнання?
3. Чи зрозуміло, як розробити відповідний режим буріння?	3. Чи зрозуміло, як розробити програму (режими буріння свердловин на нафту і газ)?
4. Чи функції елементів бурового обладнання та їх розташування є чітко зрозумілими відповідно до письмового опису та графічних представлень?	4. Чи функції окремих елементів бурового обладнання і їх розташування є чітко зрозумілими відповідно до опису технічного регламенту, який доповнюється виділенням зображеннями?
5. Чи це суттєвий спосіб подання даних про предметну галузь буріння свердловин на нафту і газ?	5. Чи це суттєвий спосіб подання даних про предметну галузь буріння свердловин?
6. Могли б Ви виконати програму (режими буріння свердловин на нафту і газ) без подальшого навчання?	6. Чи могли б Ви виконати програму (режими буріння свердловин на нафту і газ) без подальшого навчання?
7. Чи з опису технологічних операцій зрозуміло, як їх виконати?	7. Чи з опису технологічних операцій, включаючи відео, було зрозуміло, як їх виконати?
8. Чи зрозуміло, які елементи бурового обладнання із списку їх назв працюють при виконанні певних технологічних операцій?	8. Чи зрозуміло, які елементи бурового обладнання із списку їх назв і картинок працюють при виконанні конкретних технологічних операцій?
9. Чи можете Ви приблизно згадати, як виконувати технологічні операції, які обирали?	9. Чи можете Ви приблизно згадати, як виконувати конкретні технологічні операції, що були обрані?
10. Чи Ви б рекомендували цю систему для використання іншим фахівцям (студентам)?	10. Чи Ви б рекомендували запропоновану систему для використання іншим фахівцям (студентам, операторам технологічного процесу)?
11. Чи задоволені Ви інформацією, яку отримали від системи?	11. Чи задоволені Ви інформацією, яку отримали від системи?
12. Чи потребує ця система відео і зображень, щоб бути ефективною при виконанні введеного переліку технологічних операцій?	

Питання про ефективність реалізації мультимедіа були пов'язані з двома аспектами прототипу системи. Початкові три питання, які поставили суб'єктам процесу тестування, оцінюють легкість використання системи. Решта питань пов'язані з тим, чи користувач задоволений інформацією, яку отримав із системи, і чи корисна сама система.

Загалом суб'єкти, що апробували систему, погодилися, що як у системі з мультимедійними можливостями, так і без них легко орієнтуватися. Мультимедійні можливості не надто допомагають користувачеві в навігації системою. Стало також зрозуміло, як почати розробку програми режимів буріння свердловин на нафту і газ в обох системах. Ці аспекти системи не відрізнялися, бо мультимедійні можливості були використані для цього пояснення загалом.

Дані про елементи бурового обладнання було важче передати без використання графіки. Блок-схеми елементів бурового обладнання в немультимедійній версії прототипу системи представлені в текстовому форматі, а елементи бурового обладнання згруповані за категоріями залежно від їх розташування.

Щоб дізнатися про певний елемент бурового обладнання, користувач повинен клацнути на його назві. Оскільки немультимедійна версія не має жодних графічних можливостей, то зображення елементів бурового обладнання не показано. Якщо необхідна інформація про місце розташування певного елемента бурового обладнання, то користувач повинен приблизно визначити назву, на основі відповідної категорії. Хоча було зрозуміло, як натиснути на назву виділено елемента, щоб отримати інформацію про нього, це не було змістовним способом представлення цієї інформації.

У мультимедійній версії користувач просто натискає на площину бурової установки, яка його цікавить, і назві елемента бурового обладнання, тоді його функція й місце розташування відображаються в текстовому форматі на додаток до виділених картинок. Якщо елемент виділяється, набагато легше зрозуміти, де саме він знаходиться і яку функцію виконує. Суб'єкти апробації вважали, що

мультимедійні пояснення були зрозумілішими, ніж не мультимедійні, і тому легшими для сприйняття.

Ще одна спектр – це пояснення конкретних технологічних операцій. Немультимедійні описи є строго текстовими, хоча вони організовані в кілька категорій, які визначають злагоджену роботу всіх елементів обладнання. Мультимедійна версія містить ці ж письмові описи, а також дозволяє користувачеві переглянути відео виконання певної технологічної операції. Текстова інформація надає всі необхідні відомості для оператора ТП, щоб мати можливість виконувати технологічну операцію, однак цієї інформації не завжди достатньо для належного її виконання. Багато деталей опису технологічних регламентів нелегко засвоїти. Варто відзначити, відеоопис технологічної операції подає цю інформацію таким чином, що її простіше зрозуміти. Після відеоперегляду технологічної операції всі суб'єкти вважали, що зможуть відтворити її без особливих зусиль. Без відео вони не були впевнені чи зможуть відтворити письмовий опис в усіх операційних діях. Тому немультимедійна версія була оцінена нижче, ніж мультимедійна.

Інший аспект технологічної операції визначає, які саме елементи бурового обладнання залучаються до її виконання. У прототипі немультимедійної версії список назв елементів бурового обладнання був доступний, тому оператор міг це визначити. На додаток до цього списку мультимедіаверсія забезпечує зображення з виділенням (контуризацією) відповідних елементів обладнання. Оператор може перемикатися між цим зображенням і списком за необхідності. У той час, коли зображення чітко описує, які елементи бурового обладнання беруть участь у кожній конкретній операції, список назв не надає такої можливості. Для визначення місця розташування елементів бурового обладнання необхідно здійснювати запит у списку на письмовий опис кожного елемента окремо.

Отже, тестова група вказала, що в мультимедіа-версії більш зрозуміло, які елементи бурового обладнання до якої операції залучаються.

Іншим важливим компонентом *інтелімедійної системи підтримки прийняття рішень у процесі буріння свердловин*, що базується на знаннях, є її здатність проектувати програму режимів буріння з поставленими цілями для конкретного

оператора. Хоча проектувати програму режимів буріння однаково просто в обох версіях системи, її важко виконувати без додаткової інформації. Додаткова інформація, яку оператор потребує в першу чергу, стосується виконання кожної конкретної операції в обраному режимі. На основі роз'яснення для операцій, закладених у систему. Суб'єктів тестування запитували, чи вони зможуть реалізувати програму режимів буріння без подальшого навчання (крім частини цієї системи). Вони були переконані, що зможуть реалізувати програму режимів буріння, маючи доступ до мультимедійної версії системи, але не були впевнені, що змогли б це зробити без мультимедійної версії системи.

Крім того, суб'єктам тестування обох версій поставили запитання, чи потрібні мультимедійні можливості цій системі для ефективнішої роботи. Відповіді на це питання показали, що мультимедійна версія вважається користувачами більш ефективною, ніж немультимедійна. Деякі суб'єкти апробації зазначили, що вагалися у власній інтерпретації письмової інформації про режими буріння, проте, як тільки система видала відеоописи відповідних операцій, впевненість суб'єктів тестування значно підвищилась (у контексті власного розуміння щодо виконання). У процесі виконання операцій програми режимів буріння важливо встановити довіру користувачів при апробації системи і їхнього розуміння отриманих знань у процесі роботи системи. Цей досвід є недоречним, якщо оператор недостатньо впевнений, чи варто вживати заходи, запропоновані системою (це особливо актуально в експертних системах, пов'язаних із процесом буріння, де вартість помилки оператора є надто високою).

Ще одним важливим фактором ефективності системи кількість інформації, яка може бути збережена користувачем (оператором ТП). Це природним чином впливає на те, як буде представлено цю інформацію. Всіх суб'єктів запитували, чи змогли б вони приблизно згадати, як виконувати обрані технологічні операції. Очевидно, що мало хто з суб'єктів був впевнений, що зможе згадати певну технологічну операцію тільки на основі її письмового опису. Після перегляду відео для кожної технологічної операції, суб'єкти були достатньо впевнені, що вони загалом зможуть пригадати, як виконувати ці операції. Використання різних, більш інтуїтивних форм

подання інформації є набагато ефективнішим, ніж просте представлення великої кількості інформації. Більшість суб'єктів описували своє бачення кожної з операцій відповідно до того, що вони бачили, і не обов'язково так, як вони прочитали у відповідних текстових описах. Крім того, відео допомогло прояснити багато незрозумілих ситуацій, які виникли у зв'язку з використанням незнайомої термінології. Замість того, щоб перевантажувати оператора письмовим визначенням відповідних термінів, слід зосередитись на більш важливих питаннях, а тлумачення цих термінів можна описати за допомогою зображень.

Це особливо доречно при описі різних елементів бурового обладнання. Багато елементів обладнання не мають загальних назв або їх загальні назви важко згадати, як і їх власні назви. Оскільки більш важливо зрозуміти розташування кожного елемента бурового обладнання бурової установки, а також його конкретне призначення, тому тільки текстових описів не достатньо. Виявлено, що кращим способом пояснення інформації є графічне зображення. Хоча деякі текстові описи необхідні в багатьох випадках, проте вони використовуються системою для побудови пояснень.

Отже, користувач може відфільтрувати тільки текстову інформацію, яку, на його думку, необхідно уточнити, для кращого розуміння наведених текстових описів. Так моделюється реальний стан процесу буріння свердловин, вказуючи оператору на певний елемент бурового обладнання й пояснюючи, як він працює на прикладі конкретної технологічної операції.

Важливо відзначити, що ті суб'єкти, які консультувалися з немультимедійною версією системи, як правило, оцінюють цю версію вище, ніж ті, хто спершу консультувався з мультимедійною версією. Між іншим, багато з них зазначили, що вони змінили б свої відповіді, якщо б мали змогу переконатися в довершеності мультимедійної версії системи [211-216].

Таким чином, результати опитування показують, що користувачі більше задоволені, коли інформацію їм подать у різних формах. Кожен суб'єкт процесу отримує й розуміє інформацію по-різному. Надаючи інформацію в різних формах, можна задовільнити різні цільові групи (від студентів до реально діючих операторів

ТП), і тоді експертна система може бути кориснішою й економічно ефективнішою. Це є важливим чинником у будь-якому технологічному процесі, який розглядає використання технології експертних систем.

4.4 Тестування отриманого прототипу в режимі інтелектуального тьютора-тренажера

Інтелектуальна тьюторна система (ІТС) – це один із видів автоматизованих систем, що дозволяють реалізовувати спрямований і налаштовуваний тьюторний процес на основі методів зворотного зв'язку з об'єктами тренінгу та з використанням методів експертних систем в інтелектуальному режимі під час виконання навчальних або технологічних задач.

Як показує аналіз існуючих ІТС, системи даного класу [217-219] складаються з 4 основних підсистем або модулів (блоків):

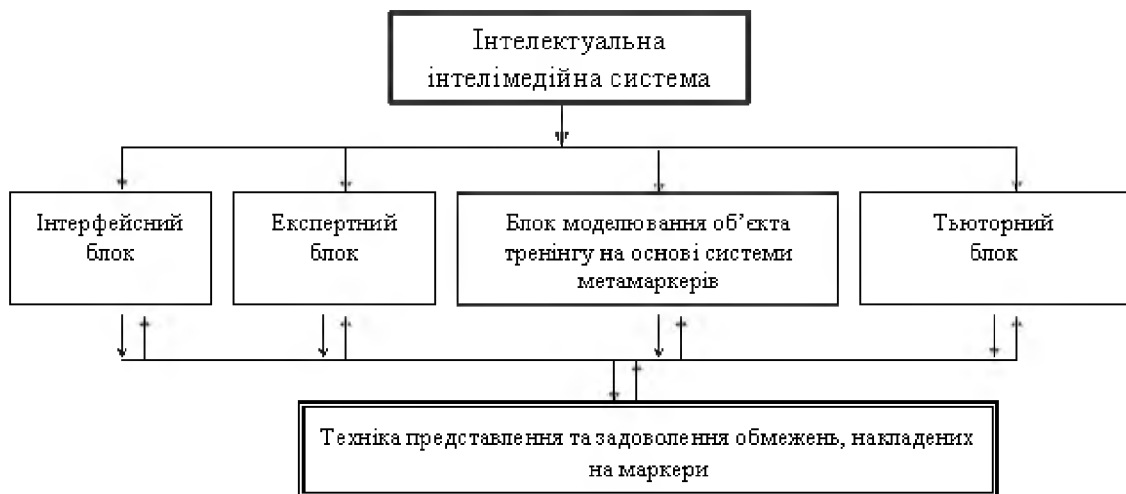


Рисунок 4.4 – Структурна схема інтелектуальної інтелімедійної системи

Інтерфейсний модуль надає об'єкту тренінгу доступ до ресурсів ІТС і підтримує окрему взаємодію користувачів і груп користувачів з ІТС. Розглянемо два основні підходи до реалізації інтерфейсу ІТС:

1. Класичний графічний інтерфейс GUI. Є найбільш поширеною реалізацією класичних ІМАС, інтелектуальних ІМАС та інтелектуальних ІМАС на основі підходу представлення та задоволення обмежень.

2. Побудова віртуальної симуляції об'єктної динаміки предметної галузі *SubjectDomainSimulation*.

Прикладами таких ефективних симуляцій предметних областей є випадки вивчення ходу технологічних процесів (наприклад, у нафтогазовій промисловості).

Експертний модуль є результатом імплементації експертних й інтелектуальних методів підтримки доменної моделі знань *KnowledgeDomainModel* і формалізації експертного досвіду *ExpertActivities*, як правило, в формі системи правил або системи обмежень, що відображає онтологічну доменну модель предметної області з визначеною системою об'єктів, суб'єктів, правил, зв'язків між ними та обмежень, накладених на ці зв'язки.

Таким чином, реалізація ІТС може інтерпретуватись як вид експертної автоматизованої системи або оригінальної реалізації інтелектуальної системи на основі когнітивної моделі *CognitiveModel* предметної області. Прикладом такої реалізації є діагностика та наступні коректуючі дії експерта при нештатному функціонуванні технічної системи.

Модуль об'єкта тренінгу дозволяє вести та підтримувати поточну модель об'єкту тренінгу на основі його системного профілю, що складається з параметрів, які описують поточний і підсумковий рівні засвоєння контенту тренінгового матеріалу; поточний і підсумковий рівень знань, умінь і навичок об'єкта тренінгу; описи і зразки ідеальної поведінки експерта предметної області. На основі моделі система може тримати зворотний зв'язок з об'єктом тренінгу, здійснювати керівні впливи для усунення хибності, неповноти та неточності в знаннях.

Завдання **тьюторного модуля** полягає в генерації коректуючих впливів *CorrectiveFeedbacks* як частини механізму зворотного зв'язку. З точки зору об'єкта тренінгу коректуючі впливи є послідовностями технологічних інструкцій *TechnologicalInstructions*. Тьюторний модуль містить ідеальні моделі *IdealModels*, що описують зразки оперування з об'єктами предметної області, виконані експертом-людиною. Такі представлення також називають тьюторною моделлю *TutorModel*.

Ефективність роботи тьюторного модуля залежить від того, наскільки він точно зможе знаходити невідповідності між шаблонами ідеальної поведінки

експерта, що міститься в його базі знань, та діями, знаннями, уміннями та навичками об'єкта тренінгу, які формуються та проявляються при його роботі з технологічним контентом і при вирішенні технологічних завдань і технологічних проблем.

Таким чином, загальна ефективність і функціональність ІТС безпосередньо залежатиме від ефективності та функціональності імплементації моделей об'єктів тренінгу, суб'єктів тренінгу, ідеальних моделей експертної поведінки, моделей знань, умінь та навичок.

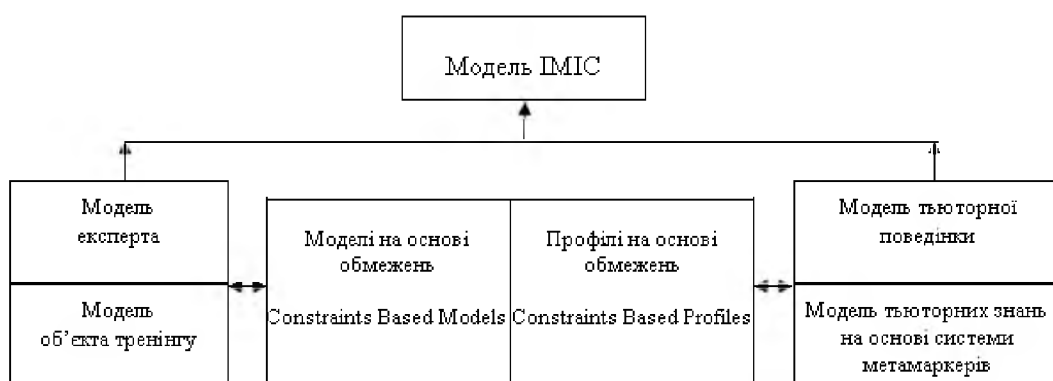


Рисунок 4.5 – Структурна схема моделей модулів ІМАС

Як показав досвід практичних реалізацій, загальна ефективність ІТС визначається рівнем імплементації двигуна висновку (механізму інференції) на основі правил або на основі обмежень. Це є основою інтелектуальності ІМАС на рівнях генерації зворотного зв'язку, генерації коректуючих технологічних інструкцій та на рівні інтеграції елементів баз даних ІТС, елементів баз знань ІТС, елементів сховищ даних ІТС в єдину повнофункціональну навчальну систему.

Основна перевага впровадження ІТС в технологічний процес полягає в можливості введення комп'ютерного аналога експертної поведінки людей-тьюторів та створення інтелектуального, адаптивного, мережевого технологічного середовища. Зокрема, при вивченні групами студентів технічних дисциплін (наприклад, мов програмування), а також проведенні технічних тренінгів для обслуговуючого персоналу автоматизованих технологічних процесів. Користувачів ІТС класифікують згідно з традиційним підходом, що використовується в

автоматизованих технологійчних системах, а саме: категорія викладачів і категорія об'єктів тренінгу (студентів). Їх специфікація така:

1. Автори курсу: *ContentExpert* наповнює ІТС знаннями предметної сфери *SubjectDomain*, формуючи таким чином тренінговий матеріал *TrainingContent*, а також системні адміністратори *SystemAdministrators* та адміністратори контенту *ContentAdministrators*. Завданнями авторів курсу є також визначення тренінгових стратегій *TrainingStrategies*, що забезпечать ефективне управління процесом тренінгу (тьюторингу), визначення тактичних і стратегічних цілей, і підцілей функціонування ІТС у вигляді послідовності системних цілей *SystemGoals*, а також здійснення загальних налаштувань ІТС.

2. Об'єкти тренінгу *TrainingObjects* поділяються на два основні типи за діяльністю: тренінгова діяльність *TrainingActivities* для окремих студентів або груп студентів й управління технологічними процесами *TechnologicalActivities*. Незалежно від видів тренінгових об'єктів робота системи складається з множини сеансів *SystemSessions*. Перебіг кожного сеансу містить такі обов'язкові етапи: ідентифікація поточного стану процесу (тренінгового або технологічного) *CurrentState*; виконання послідовності операцій, релевантних поточному стану *CurrentState Activities*, що для ІТС на основі обмежень полягатиме в генерації множин релевантних, задоволених і порушених обмежень відповідно до стану проблеми. Особливість інтелектуальності системи на основі обмежень у тому, що аналіз результатів дій об'єктів тренінгу, які контролюються на основі відповідних множин обмежень, дозволяє адаптивно вибирати наступні поточні стани тренінгових *Trainingsroblems* чи технологічних проблем *Technologicalroblems*. Завершення кожного сеансу передбачає перевірку досягнення реалізації поточної цілі *CurrentGoal* або підцілі *CurrentSubGoal*.

При задоволенні цілі об'єкт тренінгу отримує системне повідомлення про успішність задоволення поточного стану *CurrentState*, поточної проблеми *CurrentProblem* і в його профіль *CurrentProfile* відповідний запис вноситься. Якщо ціль не задоволена, то об'єкт тренінгу одержує візуалізацію ієрархії обмежень з індикацією порушених обмежень.

Інтелектуальність системи імплементується на багатьох рівнях, а саме:

Рівень 1. Контроль і супровід об'єктів тренінгу при роботі з системою та визначення доступу до ієрархії технологічних проблем.

Рівень 2. Початкова реєстрація та наступна ідентифікація об'єктів тренінгу на основі системи профілів, розпізнавання відповідного профілю системою й інтелектуальне адаптивне поведіння системи з кожним окремим об'єктом, присвоєння об'єктам ієрархії цілей і підцілей.

Рівень 3. Контроль процесу тренінгу об'єкта згідно з активованими стратегіями, підтримка процесу прийняття рішень на основі контролю задоволення множини цілей і підцілей, генерація зворотного зв'язку й ієрархічно впорядкованих рекомендацій системи у вигляді експертних порад.

Завдання ІТС полягають у реалізації теорії тренінгу через виконання, що базується на використанні методів штучного інтелекту, зокрема експертних систем для симуляції діяльності тьютора-людини на основі концепції представлення та задоволення обмежень.

Мета тьюторної системи визначається в контексті задач автоматизованих тьюторних систем і полягає в передачі об'єкту тренінгу певного обсягу знань, закладених у контентний модуль бази знань.

Верифікація степеня задоволення цілі перевіряється на основі контролю множини порушених обмежень та аналізу моделі об'єкта тренінгу *TrainingObjectModel*, представленої у відповідному системному профілі *SystemProfile*. Виконання такої верифікації вимагає наявності деякої ідеальної моделі об'єкта тренінгу *ObjectIdealModel*, яка містить типовий набір обмежень, що описує успішне задоволення поточної цілі *CurrentGoal*.

Інтелектуальні тьюторні системи на основі обмежень є адаптивними, оскільки дозволяють здійснювати контроль технологічного процесу на основі коректуючого зворотного зв'язку *Current Feedback* із об'єктом тренінгу *TrainingObject*. Функція адаптивності дозволяє активувати відповідний рівень в ієрархії технологічних проблем та їх станів відповідно до поточних потреб *CurrentRequirements* і поточного стану *CurrentState*, виділеного (ініціалізованого) об'єкта тренінгу. При ініціалізації

об'єкта тренінгу система визначає першочергові цілі *InitialGoals*, в якості яких виступає проблемна область *SubjectsDomain* і поточна задача *SubjectsTask* з виділеного тренінгового курсу *TrainingCourse*.

Модель тестування *TestingModel* в ІТС на основі обмежень (як і в класичних автоматизованих тренінгових системах) базується на ідеї адаптивного тесту *AdaptiveTest*. Проте природа адаптивності та критерії її оцінки відрізняються, оскільки базуються на концепції представлення та задоволення обмежень.

Як і в класичних моделях, тестування на основі контролю порушених обмежень є ефективним тільки при використанні локального або мереженого програмного забезпечення. Використання друкованих версій такого тесту має додаткову складність, пов'язану з необхідністю обробки не тільки кількісних, але і якісних показників результатів тестування, що визначаються обмеженнями.

В класичному підході до тестування прийнято використовувати метод побудови тесту через поступове підвищення його складності – *IncrementalTestComplexity*. Тобто тестування починається з легких запитань, потім йдуть запитання середньої складності тощо. В результаті, тест вважається завершеним, коли об'єкт тренінгу успішно відповідає на визначену послідовність запитань заданого рівня складності *ComplexityLevel*. Особливість адаптивного тестування на основі обмежень з точки зору контролю та підтримки рівня складності визначається на основі коефіцієнтів важливості обмежень (вагових коефіцієнтів) *WeightedConstraints* і вибраного способу ранжування обмежень *RangedConstraints*.

Основна перевага адаптивного тестування на основі контролю множини порушених обмежень полягає в тому, що оцінка рівня знань носить чітко виражений якісний характер, який не залежить від кількості заданих об'єкту запитань і кількості отриманих відповідей, на відміну від класичного адаптивного тесту, оцінювання знань у якому хоча і співвідноситься з поточним рівнем знань об'єкта тренінгу *TrainingObjects*, але має кількісну природу. Також такий підхід дозволяє виконувати оцінку знань більш точно і з меншою кількістю завдань та, досліджуючи

структуру обмежень, порушених об'єктом тренінгу, визначати технологійні проблеми, рівень знань яких є недостатнім, і генерувати додаткові послідовності тестових запитань, максимально релевантні до виокремлених технологічних проблем.

Проте організація адаптивного тестування на основі контролю порушених обмежень має певну складність: визначення необхідної мінімальної чи максимальної кількості тестових запитань, релевантних до поточного стану поточної технологічної проблеми, визначається мінімальною чи максимальною кількістю релевантних обмежень, закладених на етапі проектування бази знань. Цей фактор визначає також спосіб актуалізації поточного рівня знань об'єкта та вибору відповідної стратегії підтримки області найближчого розвитку. Але за однакової кількості тестових запитань адаптивний тест на основі контролю порушених обмежень є більш ефективним і продуктивним з точки зору якісної складової рівня знань та підтримки актуального рівня знань в області найближчого розвитку.

Надійність результатів тестування в ІТС на основі обмежень визначається областю найближчого розвитку об'єкта тренінгу. Це означає, що запитання тесту підбирається таким чином, щоби відповідати поточному рівню знань об'єкта тренінгу. Тому важливо є постійно контролювати рівень знань об'єкта тренінгу і його співвідношення з ключовими характеристиками області найближчого розвитку.

Класична технологія адаптивної підтримки навігації в автоматизованих системах полягає у визначенні списку гіперпосилань, що відображатимуться, і гіперпосилань, що приховуватимуться, залежно від того, чи вони позначають технологічні проблеми, які знаходяться в межах області найближчого розвитку об'єкта тренінгу чи поза нею. Такий підхід дозволяє реалізувати оптимальну послідовність вивчення технологічного матеріалу на основі адаптивного підходу або на основі емпіричного впорядкування. На відміну від такого підходу, в ІМАС застосовується підхід оптимальності. Він базується на моделях і профілях об'єкта тренінгу, що проводяться на основі інтелектуальних методів сумаризації та контролю множини обмежень *UserConstraintsHistory* .

При роботі з системою груп об'єктів тренінгу кожен із них, зокрема, розглядається як суб'єкт тренінгу *TrainingSubject* і, відповідно, кожен із них відправляє в систему власний варіант вирішення поставленої проблеми *SubmittedSolution*. Оцінка кожного з запропонованих рішень виконується на основі аналізу множини порушених обмежень *ViolatedConstraints* і множини задоволених обмежень *SatisfiedConstraints*, а також порівняння їх вагових коефіцієнтів та коефіцієнтів ранжування. Такий підхід в ІМАС дозволяє класифікувати суб'єктів тренінгу за їх когнітивними здібностями *CognitiveSkills*. Ефективність цього підходу в тому, що студенти можуть одержувати знання на основі аналізу розв'язків проблем, запропонованих іншими об'єктами тренінгу, через порівняння множин релевантних, порушених і задоволених обмежень, а також їх вагових і рангових коефіцієнтів, заданих експертом на етапі формування бази знань.

Структура технології адаптивного представлення в ІТС на основі обмежень визначається завданням співвідношення контенту тренінгового матеріалу *TrainingContent* до поточного рівня знань об'єкта *CurrentKnowledgeLevel* та їх сумарної відповідності параметрам області найближчого розвитку об'єкта тренінгу.

Під адаптивним тестом в ІМАС розуміють класичний адаптивний тест, складність завдань у якому регулюється кількістю порушених і задоволених обмежень під час вирішення об'єктом тренінгу поточного стану поточної проблеми. В класичному адаптивному тесті складність тестових завдань прямопропорційна кількості правильних відповідей на попереднє запитання. Тобто, чим більшу кількість правильних відповідей дає студент, тим більш складні запитання йому пропонує система. І навпаки: чим більше неправильних відповідей дає об'єкт тренінгу, тим швидше знижується складність наступних тестових запитань. Відповідно, ефективність технологічної стратегії визначатиметься кількістю та складністю запитань, релевантно співвіднесених до рівня знань об'єкта тренінгу за відповідними технологічними проблемами. При використанні адаптивного тестування на основі обмежень зв'язок кількості правильних чи неправильних відповідей і рівень складності тестових питань визначаються на основі контролю множини обмежень, їх вагових коефіцієнтів і способу ранжування.

Інтелектуальність в ІМАС досягається шляхом введення елементів технології штучного інтелекту *ArtificialIntelligence*; зокрема, таких як: база знань (*KnowledgeBase*), логічний висновок (*LogicalInference*). Завдання класичної автоматизованої системи *AutomatedSystem* полягають в організації комп'ютер-базованого (веб-орієнтованого) навчання *ComputerBasedLearning* у технологічному середовищі *TechnologicalEnviroment*, що створюється окремим комп'ютером або обчислювальною мережею, а також контролі результатів навчальної діяльності об'єкта тренінгу *TraineeObjectActivities* у формі тестового контролю рівня знань *KnowledgesLevelControl*, рівня вмінь *SkillsLevelControl* і рівня навичок *ExpierencesLevelControl* роботи в предметній області *SubjectDomain* або області знань *KnowledgeDomain*. Інтелектуальна автоматизована система дозволила одержати нові можливості, а саме:

1. Визначати недостатні знання *InsufficientKnowledges*, помилкові знання *ErroneosKnowledges* на основі аналізу технологічної діяльності об'єкта тренінгу *TraineeObjectActivities*.

2. Виконувати переспрямування технологічної діяльності *TraineeActivitiesRedirecting* об'єкта тренінгу на відповідний розділ контенту тренінгового матеріалу *TraineeContentTopic*.

3. Генерувати для об'єкта тренінгу додаткові контекстно залежні пояснювальні блоки *ContextExplanationBlock* відповідно до поточних результатів тренінгової діяльності *TraineeCurrentResults* і системного профілю об'єкта тренінгу *SystemTraineeProfile*.

4. Виконувати адаптацію тренінгових стратегій *TraineeStrategies* відповідно до особливостей профілю об'єкта тренінгу *TraineeProfileFeatures*.

5. Визначати стратегію управління тренінгом відповідно до поточних і кінцевих результатів об'єкта тренінгу в сеансах роботи з системою при вивченні курсів тренінгу, тем, підтем і модулів.

6. Формувати сценарій технологічного процесу адаптивно до контексту поточної технологічної ситуації, знань про предметну область, а також послідовності динамічно обновлюваних профілів об'єкта тренінгу.

1. Визначаються елементи тренінгу *TraineeElements*, пов'язані один з одним у межах тренінгового курсу *TraineeCourse*, розділу тренінгового курсу *TraineeCoursePart*, теми, підтеми, змістовного модуля.

2. Генерується квадратна матриця, розмірності якої відповідають кількості виділених тренінгових елементів *SelectedTraineeElements*.

3. У заголовках рядків та стовпців розміщуються позначення тренінгових елементів з накладеними системами обмежень *ImposedConstraintsSystem* виду $L_i C_i$, де L_i – i -й тренінговий елемент, C_i – відповідна йому система обмежень.

4. Утворені клітинки матриці заповнюються дробовими значеннями, у чисельниках і знаменниках яких вказуються R, \bar{R} . Розміщення символу R в чисельнику означає, що контент тренінгового матеріалу тренінгового елемента, представленого в номері рядка, логічно зв'язаний з контентом тренінгового матеріалу тренінгового елемента, зазначеного в номері стовпця. Розміщення символу R у знаменнику означає релевантність відповідних систем обмежень, накладених на тренінгові елементи. Використання символу \bar{R} означає відсутність логічного зв'язку або відсутність релевантності. Надлишкові зв'язки, які збігаються, та релевантності не вносяться в матрицю. Введене представлення матриці логічних зв'язків та релевантностей є однією з можливих імплементацій графологічних зв'язків *LogicalDependenciesGraph*.

Розглянемо приклад матриці логічних зв'язків та релевантностей для випадку контенту тренінгового матеріалу, що складається з N тренінгових тем, з накладеними на них системами обмежень.

Подана матриця логічних зв'язків та релевантностей контенту тренінгового матеріалу, представленого через виділені тренінгові елементи, не визначає прямолінійну стратегію послідовності вивчення тренінгових елементів, як у класичних ІМАС, що використовують матриці логічних зв'язків без введених релевантностей систем обмежень. У нашому випадку така послідовність визначатиметься на основі існуючої ієрархії систем обмежень *ConstraintsSystemHierarchy* і не буде прямолінійною.

Матриця відношень черговості *RelationPriorityMatrix* визначає послідовність слідування тренінгових елементів у тренінгових стратегіях. Розмір матриці визначається кількістю введених тренінгових елементів, де кількість рядків і стовпців збігається. Як заголовки рядків і стовпців використовуються представлення тренінгових елементів з накладеними системами обмежень. Комірки утвореної матриці заповнюються дробовими значеннями виду $Row^iRR/Column^jRR$. Ця формула побудована на основі способу представлення множин обмежень, накладених на тренінгові елементи, а саме: їх ранжування за ступенем релевантності *RelevancyRanging* до тренінгового елемента.

Таким чином, значення в чисельнику представляє ступінь релевантності системи обмежень тренінгового елемента рядка відносно тренінгового елемента у стовпці. Значення у знаменнику показує ступінь релевантності системи обмежень тренінгового елемента стовпця відносно тренінгового елемента рядка. В результаті, послідовність тренінгових елементів у вибраній тренінговій стратегії визначається на основі порівняння ступенів релевантності їх систем обмежень.

Автоматизована система вважається розподіленою, якщо механізм її функціонування базується на основі мережевих засобів і методів і реалізація технологічного процесу та взаємодія з об'єктом тренінгу здійснюється на основі апаратних та програмних засобів комп'ютерної мережі. Більшість новітніх систем автоматизованого тренінгу, які відносяться до класу дистанційних систем та передбачають віддалений доступ, не відносяться до класу розподіленої інтелімедійної автоматизованої системи (PIMAC), оскільки орієнтовані на надання доступу до ресурсів для віддалених користувачів без використання обчислювальних можливостей мережі.

Таким чином, основною характеристикою PIMAC є використання обчислювальних можливостей мережі для вирішення задач об'єктів тренінгу та обміну даними між ними.

Особливості PIMAC визначаються їх функціональністю, що полягає в автоматизованому адаптивному переналаштуванні системи відповідно до рівня знань об'єкта тренінгу *KnowledgeLevel* і його преференцій, вимог і потреб

(*TrainingObject Preferences*, *TrainingObject Demands*, *TrainingObject Requirements*); підтримці модуля зворотного зв'язку *TrainingObject Feedbacks*; збереженні та представленні контенту предметної області *SubjectDomainContent* у вигляді бази даних, бази знань або сховища даних; веденні протоколу цілей тренінгу *TrainingGoals*, а також визначенні умов їх задоволення *GoalSatisfactionCriteria*.

Крім функціональності, на архітектуру РІМАС впливають також вимоги необхідності організації ефективного використання обчислювальних можливостей мережі, а саме: підтримка доступу до розподілених даних *SharedDataAccess*; необхідності уніфікації й організації спільної роботи гетерогенних програмних засобів, що забезпечують розподіл та обслуговування технологічних проблем та задач, визначених об'єктами тренінгу; організації та підтримки взаємодії користувачів (об'єктів тренінгу) між собою *TraineeObjectsInteraction* і їх взаємодії з системою *SystemInteraction* на основі обчислювальних ресурсів мережі.

Для імплементації виділених архітектурних рішень РІМАС необхідно забезпечити реалізацію та підтримку різних форм *TrainingForms* і видів тренінгу *TrainingModes*. Класичними формами та видами тренінгу, що застосовуються, є електронні лекції, лабораторні роботи та тестовий контроль знань. В ІМАС, крім класичних засобів, основний акцент приділяється вирішенню послідовності технологічних проблем, що описуються певною множиною обмежень *ConstrainedLearningProblem*.

Необхідно використовувати засоби контролю процесу тренінгу *TrainingProcessControl*, що дозволяють створювати та використовувати стратегії тренінгу *TrainingStrategy* і вибирати оптимальну стратегію тренінгу відповідно до поточної моделі об'єкта тренінгу *TrainingObjectModel*. Ефективність застосування технологічних стратегій залежить від наявності процедур визначення рівня знань об'єктів тренінгу *KnowledgeLevel*, а також адаптації системи до поточного рівня знань об'єкта тренінгу. Ефективними реалізаціями РІМАС є предметно незалежні реалізації, наповнені прикладними знаннями *AppliedKnowledges* виділеної предметної області *SubjectDomain* і, відповідно, реалізують технологічний процес,

орієнтований на цю предметну область. Крім того, необхідно забезпечення віддаленого доступу об'єктів тренінгу до системного середовища використання системи "логіп-пароль", створення та ведення системи профілів користувачів *TrainingObjectProfile* і подальшої їх ідентифікації та перевірки (верифікації) на всіх етапах роботи з системою. Відповідно до профілів користувачів та їх моделей, як об'єктів тренінгу, система виконує розділення доступу до наявних ресурсів технологічного контенту, представлених у формі баз даних контенту *ContentDataBases*, баз знань контенту *ContentKnowledgeBases* і сховищ контенту *ContentRepository*.

Описані властивості РІМАС дозволили розглядати таку систему, як систему з відкритою архітектурою, з наявними можливостями налаштування, реконфігурації та розширення, шляхом підключення нових баз даних, баз знань та модулів інференції.

Важливою є також функціональність РІМАС з точки зору динаміки, циклу технологічного процесу *TechnologicalCycle*, що полягає в імплементації доступних форм організації тренінгу *TrainingForms* та їх адаптації до початкового рівня знань об'єктів тренінгу *InitialKnowledgeLevel*, поточного рівня знань *CurrentKnowledgeLevel*, вимог щодо підсумкового контролю знань *SummarizedKnowledgeControl* і підсумкового рівня знань *SummarizedKnowledgeLevel*. Ефективне використання обчислювальних можливостей мережі дозволяє створювати на основі РІМАС змішані та гібридні веб-орієнтовані технологічні системи, наприклад, з використанням експертних модулів й інтеграцією з існуючими системами та платформами дистанційного навчання.

Інтелектуальна автоматизована система *IntelligentAutomatedSystem*, на відміну від звичайної ІМАС, може організовувати технологічний процес без попередньої підготовки та верифікації послідовності технологічних проблем *TechnologicalProblemSequence*, множини їх ідеальних рішень *IdealSolutionSequence*, послідовності технологічних завдань *TechnologicalTasksSequence* і множини ідеальних відповідей на них *IdealAnswersSequence*. Тобто інтелектуальна система генерує технологічні проблеми та технологічні завдання в режимі виконання

(*RuntimeTechnologicalProblems*, *RuntimeTechnologicalTasks*). Контроль коректності рішень і відповідей на етапі виконання забезпечується окремим модулем автоматизованої системи, який називається вирішувачем проблем *ProblemSolver*. Проте даний підхід має суттєвий недолік, оскільки вирішення проблем є предметно- та доменнозалежним.

Прикладом предметно незалежних універсальних ІМАС є селективні автоматизовані системи. Тренінгова діяльність *TrainingActivityRoutine* для кожного з об'єктів *TrainingObjects* ідентична і є результатом застосування одних і тих же тренінгових стратегій *TrainingStrategies*. Тобто значення характеристики *TrainingObjectsAdaptivness* є низьким, а система не адаптується до вимог і потреб конкретного об'єкта або груп об'єктів. Управління тренінгом *TrainingControl* (визначення форми контенту *ContentForm*, тестів, завдань, технологічних проблем, інтерактивної та контекстної допомоги тощо) здійснюється автором курсу *ContentExpert*. Система складається з трьох основних блоків: блоку генерації завдань та технологічних проблем, блоку верифікації та блоку вибору стратегії тренінгу.

Схематично таку автоматизовану систему можна представити таким чином:

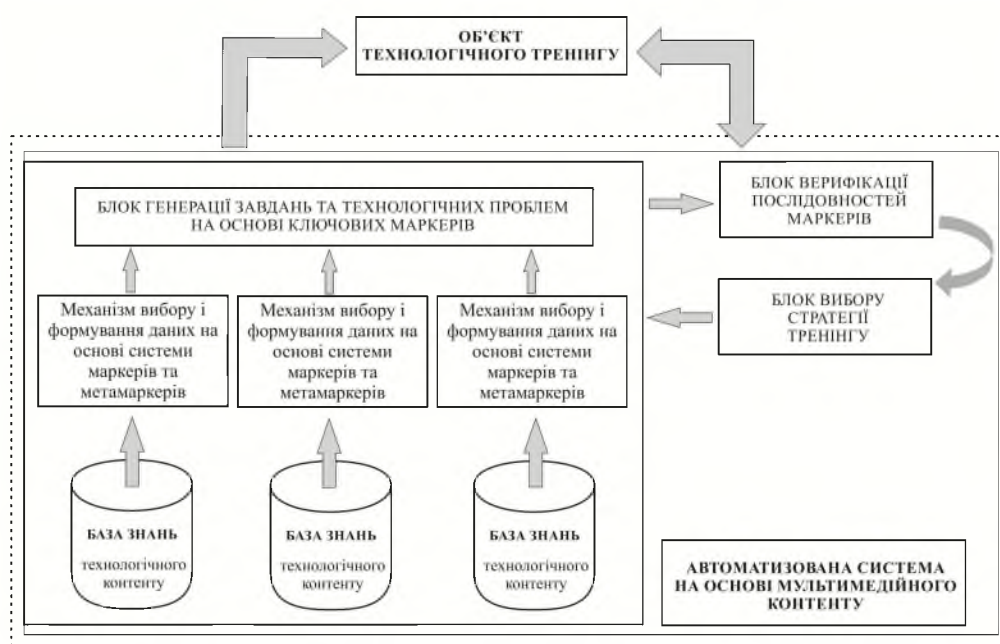


Рисунок 4.7 – Схематична структура інтелемедійної автоматизованої системи

Автоматизовані системи тренінгу використовують методи дистанційної технології тренінгу *DistanceTraining*, які використовують засоби телекомунікаційних мереж і технологій для пропагації контенту тренінгового матеріалу в середовищі інтерактивної взаємодії *TraineeInteractionEnvironment* об'єктів і суб'єктів тренінгової діяльності *TrainingActivities*.

Блок верифікації в ІМАС призначений для контролю знань об'єкта тренінгу *KnowledgeLevelControl*. Цей блок на вході одержує відповідь об'єкта тренінгу *TrainingObject* на технологічне завдання *TechnologicalTask* або його варіант розв'язку тренінгової проблеми *TrainingProblemSolution*, порівнює його з правильною відповіддю *CorrectAnswer* або коректним рішенням *CorrectSolution* і виконує оцінювання результатів тренінгової діяльності. Результатом оцінювання є прийняття рішення щодо застосування коректної тренінгової стратегії *TrainingStrategy* до об'єкта тренінгу, а також сумаризації відповідних показників його профілю *TrainingObjectProfile*.

ІМАС орієнтовані на роботу з даними, які представляються та зберігаються в базах даних і сховищах даних. Тому цей тип систем є орієнтований на дані *DataBasedSystem*. Відповідно, інтелектуальні автоматизовані системи *IntelligentAutomatedSystem* орієнтовані на знання. Тобто вони є системами класу *KnowledgesBasedSystem*, оскільки у своїй роботі використовують методи штучного інтелекту, зокрема експертних систем, і оперують з такими концепціями як база знань, у якій представляється логічно впорядкований контент технологічного матеріалу та механізм логічного висновку як спосіб генерації нового контенту та логічного аналізу існуючого.

Одним із основних елементів ІМАС є блок вибору стратегії тренінгу. Цей блок призначений для формування та вибору технологічних стратегій *TechnologicalStrategys*. На вході він отримує інформацію про результат тренінгу *TrainingResults*, на виході – видає зважене рішення щодо вибору тієї чи іншої

імплементції певної стратегії тренінгу відповідно до поточного профілю *CurrentProfile* об'єкта тренінгу *TrainingObject*.

Блок генерації завдань і технологічних проблем, як складова автоматизованої технологічної системи, призначений для генерації технологічних завдань *TechnologicalTasks* і технологічних проблем *TechnologicalProblems* щодо виділеної предметної області відповідно до поточного рівня знань об'єкта тренінгу *CurrentKnowledgeLevel*. На рівні реалізації такий блок є базою даних тренінгового контенту *ContentDataBase*, базою знань контенту *ContentKnowledgeBase*, сховищем контенту *ContentRepositotory*, де містяться завдання та технологічні проблеми, які вибираються або формуються (генеруються) відповідним механізмом відбору і формування даних *DataEngine*. Результат роботи механізму *DataEngine* – згенеровані завдання *TaskGeneration* і згенеровані технологічні проблеми *TechnologicalProblemGeneration* – виводиться об'єктом тренінгу, а також передається у блок верифікації.

В ІМАС використовуються такі характеристики, як складність контенту тренінгового матеріалу *TrainingContentComplexity*, складність технологічних завдань *TechnologicalTaskComplexity* і складність технологічних проблем *TechnologicalProblemComplexity*. Ця характеристика залежить від двох основних параметрів: рівня представлення контенту тренінгового матеріалу *ContentPresentationLevel* і кількості кроків, необхідних для досягнення успішного результату при вирішенні технологічних завдань *TechnologicalTaskSolution* і технологічних проблем *TechnologicalProblemSolution*. Основний градієнт цієї характеристики є інкрементальним (за зростанням).

Відкритість архітектурних рішень у більшості автоматизованих системах дозволяє включати в них модулі генерації, довідки з контенту *ContentReferencesModule* і блоку генерації, консультації з контенту й інференції *ContentConsultingModule*. Ці модулі предметно залежні, стосуються виділеної предметної області *SelectedSubjectDomain*, і дозволяють оперувати з послідовністю завдань

TrainingTasksSequence і послідовностями технологічних проблем *TechnologicalProblemsSequence*.

Ефективність роботи ІМАС оцінюється показником степеня автоматизації засвоєння *MasteringAutomationDegree*, який описує вміння об'єкта тренінгу *TrainingObjectSkills* як його навички *TrainingObjectExperience* у роботі з контентом тренінгового матеріалу *TrainingContent*, як рівень засвоєння контенту *ContentMasteringSkills* і ефективність практичного застосування знань, отриманих на основі контенту *ContentBasedKnowledges*, у формі застосування контентних знань *ContentKnowledgesApplication* під час вирішення завдань *SubjectDomainTasks* і технологічних проблем предметної області *SubjectDomainProblems*. Цей показник обчислюється на основі такої формули: $MasteringAutomationDegree = ExpertTime/TrainingObjectTime$, де *ExpertTime* – час успішного виконання тесту експертом предметної області (укладачем курсу), *TrainingObjectTime* – час успішного виконання тесту об'єктом тренінгу.

Ранні ІМАС використовували систему телебачення як частину та вид дистанційної технології тренінгу *DistanceTraining*, що полягає в подачі об'єкту тренінгу *TrainingObject* аудіовізуального контенту технічного курсу в режимі реального часу. Це дозволяло одержувати значну цільову аудиторію об'єктів тренінгу в віддаленому режимі.

Сьогодні новітні ІМАС використовують цю технологію у вигляді аудіо-, відео- та презентаційних файлів завантаження, які вирішують ті ж задачі, але дозволяють працювати індивідуально або в режимі офлайн. Суттєвою перевагою є можливість багаторазового повторного використання аудіовізуального контенту *AudioVisualContent*. Презентаційні файли (файли презентації), як правило, містять короткий структурований опис контенту, що дозволяє акцентовано засвоювати основний аудіовізуальний матеріал.

Згідно з цією теорією, процес тренінгу планується у вигляді схеми, що складається з таких етапів:

1. Створення мотивації об'єкта тренінгу *TrainingObject* для вивчення контенту технологічного матеріалу в вигляді системних (мультимедійних) бонусів.

2. Визначення орієнтованості та концептуальності технологічної діяльності *TechnologicalActivities* з освоєння контенту тренінгового матеріалу *TrainingContent*.

3. Використання абстрагованих представлень предметної галузі у формі обмежень високого рівня.

Одним із важливих питань побудови ІМАС є визначення складності контенту тренінгового матеріалу *TrainingContentDifficulty*, технологічного завдання *TechnologicalTaskDifficulty*, технологічної проблеми *TechnologicalProblemDifficulty* та їх співвіднесення до здібностей об'єкта тренінгу *TrainingObjectAbilities*. Ця характеристика пов'язана також з рівнем засвоєння контенту тренінгового матеріалу *ContentMasteringLevel* і залежить від рівня попередньої підготовки об'єкта тренінгу *InitialKnowledgesLevel*.

Важливим питанням при створенні ІМАС є оцінка якості оволодіння об'єктом тренінгу тренінгового матеріалу *ContentMasteringLevel*. Згідно із класичним підходом, що застосовується в ІМАС, ми виділили такі рівні засвоєння контенту тренінгового матеріалу:

1. *ComprehensiveLevel* – рівень розуміння, який характеризується тим, що об'єкт тренінгу розуміє й осмислено сприймає технологічний контент. У класичних ІМАС цей рівень визначає необхідні знання студента *KnowledgeLevel*, які дозволяють йому розуміти структуру технологічного контенту та взаємозв'язки між його складовими. Водночас характеристики цього рівня не визначають фактичне засвоєння тренінгового матеріалу у формі знань, умінь та навичок (*KnowledgesMastering*, *SkillsMastering*, *ExperiencesMastering*) і є характеристиками порогових значень знань *ThresholdKnowledges*, необхідних для просування у ланцюгу технологічних проблем. В ІМАС такий рівень визначається на основі контролю засвоєння (успішного задоволення множини обмежень) найвищого рівня.

2. *CognitiveLevel* – рівень пізнання, що полягає в розрізненні об'єктів, процесів і явищ предметної області, розуміння взаємозв'язків між ними та способів

оперування на основі вивченого контенту. Найбільш ефективним є застосування характеристик цього рівня при роботі з об'єктами, їх класифікації й онтологізації.

3. *ReproductiveLevel* – рівень відтворення, що полягає в реплікаційному відтворенні знань, отриманих на основі вивчення технологічного контенту (*ContentDeducedKnowledges*), а також їх застосуванні для типових проблем предметної області *ProblemOrientedApplication* або на основі шаблонів *PatternOrientedApplication*.

4. *ApplicationalLevel* – рівень застосування, що характеризується здатністю об'єкта тренінгу відтворювати (реплікувати) технологічний контент і модифікувати його при роботі з множиною відомих та описаних об'єктів предметної області, а також застосовувати знання, отримані на основі контенту, для нестандартних і нових ситуацій предметної області. Таким чином, цей рівень дозволяє застосовувати знання для дослідження множини об'єктів і генерувати нові знання в цьому процесі.

5 *CreativeLevel* – творчий рівень, що описує степінь засвоєння контенту тренінгового курсу *TrainingCourse*, технологічної проблеми *TechnologicalProblem* або частини тренінгового курсу *TrainingCoursePart*, коли об'єкт технологічного процесу здатний створювати новий контент відповідного рівня, розробляти нові алгоритми, методики, формальні моделі й оригінальні способи розв'язання типових і нетипових проблем предметної області.

6. *ConstraintsLevel* – рівень обмежень, характерний тільки для ІМАС. Полягає в описі структури контенту, способів коректного рішення технологічних проблем предметної області на основі введення множини звичайних обмежень або ранжованих обмежень з ваговими коефіцієнтами.

Розглянуто підхід до оцінювання рівня засвоєння контенту тренінгового матеріалу на кожному з рівнів 1-5 *ContentMasteringLevel*, що застосовується в класичних ІМАС. Оцінювання здійснено за формулою: $CML = (K_1 / K_2) \cdot 100$, де K_1 – кількість успішних коректних спроб вирішення проблем із запропонованої послідовності (або кількість правильних відповідей на тестові запитання із запропонованого переліку), K_2 – загальна кількість спроб. Згідно із загальноприйнятою шкалою рейтингового оцінювання знань, значення коефіцієнта

CML визначає такі оцінки: “відмінно” $-90 \leq CML \leq 100, ECTS = A$, “добре” $-82 \leq CML \leq 89, ECTS = B$, “добре” $-75 \leq CML \leq 81, ECTS = C$, “задовільно” $-67 \leq CML \leq 74, ECTS = D$, “задовільно” $-60 \leq CML \leq 66, ECTS = E$, “незадовільно” $-35 \leq CML \leq 59, ECTS = FX$, “незадовільно” $-0 \leq CML \leq 34, ECTS = F$.

Стратегія класичної ІМАС щодо визначення рівня засвоєння контенту тренінгового матеріалу – тестування за зростанням рівня складності. Якщо об’єкт тренінгу показує результат, більший за порогове значення успішності, тоді система виконує тестування рівня засвоєння на наступному рівні. Або навпаки: об’єкт продовжує роботу з системою на попередньому рівні.

Для ІМАС процес оцінювання здійснюється за формулою: $CML^{Constr} = (ConstrS/ConstrR) \cdot 100$, де *ConstrS* – сумарна кількість задоволених обмежень при вирішенні сукупності технологічних проблем певного рівня, *ConstrR* – сумарна кількість релевантних обмежень до сукупності технологічних проблем виділеного рівня. Шкалу оцінювання застосовують аналогічної структури.

Таким чином, спосіб оцінювання, що застосовується в ІМАС, є якісним, на відміну від кількісного оцінювання в класичних ІМАС. Крім того, якісна складова може бути посилена за допомогою введення вагових коефіцієнтів обмежень та їх ранжування.

Інтелектуальність ІМАС полягає в організації тренінгової діяльності *TrainingActivities*, яка імплементується системою відповідно до налаштувань тьютора-викладача, що є укладачем контенту тренінгових курсів; визначає структуру технологічних проблем і послідовність їх руху та виступає суб’єктом тренінгової діяльності *TrainingSubject*. Результат тренінгової діяльності об’єкта тренінгу або об’єктів тренінгу оцінюється системою та переноситься як властивість об’єкта *TrainingObjectProperty* у його системному профілі *TrainingObjectProfile*. Досягнення мети тренінгової діяльності *TrainingActivitiesGoal* забезпечується контентним наповненням, що подається у формі електронних тренінгових підручників *Training - E - Textbook*, з викладом наповнення тренінгового курсу *TrainingCourse*, його розділів *TrainingCourseChapters* та його частин *TrainingCourseParts* відповідно до вибраної

тренінгової програми (стратегії) *TrainingStrategy*. Окремі технологічні проблеми, що потребують детальних специфікацій способу розв'язання *DetailedSolution*, подано в електронних навчальних посібниках *Training - E - Manual*.

Технологія фасетів в ІМАС дозволяє записувати кілька варіантів одного й того ж завдання *TrainingTask* на основі єдиної форми *NotationForm*, що дозволяє поєднувати ряд завдань з одними і тим ж варіантами відповіді *AnswerVariants*. Система генерує для кожного ініціалізованого об'єкта тренінгу відповідний варіант і відповідний перелік правильних відповідей *CorrectSolutions*.

Отже, система генерує множину завдань *TaskSet* і множину фасетів *FacetSet*. Відповідність між множиною завдань і множиною фасетів є відповідністю один до багатьох.

При роботі з ІМАС використовувались класичні показники кількісної та якісної оцінки технологічного процесу, а саме: *ContentPresentationLevel* – рівень представлення контенту, *TrainingContentMastering* – рівень засвоєння контенту, *ContentMasteringAutomation* – степінь автоматизації засвоєння контенту, *TrainingContentPerceiving* – степінь усвідомлення тренінгового контенту, *TrainingContentComplexity* – складність тренінгового матеріалу, *TrainingContentDifficulties* – складність тренінгового матеріалу.

В результаті роботи з ІМАС (як із будь-якою іншою автоматизованою системою) об'єкт тренінгу отримує знання *Knowledges*, уміння *Skills* і навички *Experiences* діяльності в певній предметній області *SubjectDomain*. Важливо застосовувати одержані знання, вміння та навички для вирішення технологічних проблем (*ApplicationSkills, SolutionSkills*) і коректно оцінювати набуті знання, уміння та навички (*KnowledgesEvaluation, SkillsEvaluation, ExperiencesEvaluation*).

В ІМАС використовується метод представлення послідовності технологічних проблем та тестових завдань у ланцюговій формі. Суть такого представлення така: якщо об'єкт тренінгу неправильно вирішує поточну технологічну проблему *CurrentTechnologicalProblem*, то ІМАС розуміє, що він не зможе вирішити наступну технологічну проблему (тобто діяльність об'єкта тренінгу розглядається як процес

слідування ланцюгом технологічних проблем – *PreviousTechnologicalProblem*, *CurrentTechnologicalProblem*, *NextTechnologicalProblem*). Особливість реалізації ланцюгової технології представлення технологічних проблем *ChainedTechnologicalProblems* в ІМАС полягає в тому, що (хоча проблеми і зв'язані в один ланцюг для зручності функціонування механізму інференції) контроль діяльності об'єкта тренінгу не прив'язується до його поточних рішень технологічних проблем *CurrentSolution*, а переноситься на процес сумаризації порушених і задоволених обмежень, які є основним критерієм оцінки тренінгової діяльності.

ВИСНОВКИ ДО РОЗДІЛУ 4

1. Здійснено аналіз існуючого мультимедійного програмного забезпечення. Розглянуто основні пакети програмного забезпечення, їх сильні та слабкі сторони, і взаємодію з іншим програмним забезпеченням.

2. Охарактеризовано варіанти розробки інтелектуальної системи. Визначено, що оболонки експертних системи типу “Esta/Stress” є оптимальними оболонками для створення інтелектуальної системи. За певними характеристиками проаналізовано бажані можливості та функції для системи-прототипу. Здійснено аналіз існуючих оболонок експертних систем, виявлені переваги та недоліки кожної з оболонок.

3. Описано суть розробки інтелектуальної системи представлення знань у формі мультимедійного контенту. Вказано на її корисність в процесі функціонування та визначено основну мету створення.

4. Визначено доцільність розробки інтелімедійної експертної системи. Виявлено особливості предметної області та її відповідність поставленим вимогам.

5. Запропоновано опис структури бази знань для прототипу інтелімедійної інформаційної системи. Визначено види фактичної інформації та правила для розробки ефективної інтелімедійної інформаційної системи. Визначено основний

засіб розробки інтелімедійної інформаційної системи й описано процес її функціонування.

6. Описано процес проектування мультимедійного інтерфейсу в системі інтелімедійного класу. Визначено, як ефективно та практично представляти дані в такій системі. Описано джерела даних і знань про технологічне обладнання та процес буріння в цілому, що дозволило структурувати та виділити кожен із елементів бурового обладнання та побудувати відповідну систему маркерів. Представлено приклади функціонування та описано процес роботи і використання інтелімедійної системи як засобу підтримки прийняття рішень в процесі буріння. Окреслено основні функції інтелімедійної системи та пояснено спосіб її реалізації.

7. Проаналізовано процес інтеграції експертної системи та мультимедійного інтерфейсу.

8. Охарактеризовано проблеми, що виникли при тестуванні прототипу. Визначено причини та шляхи вирішення проблем, що виникають.

9. Здійснено тестування інтелімедійної системи підтримки прийняття рішень у процесі буріння свердловин. Проведено апробацію системи. Суб'єктами тестування було оцінено дві версії системи, базованої на знаннях (із мультимедійним та звичайним інтерфейсом). Для рейтингу та порівняння кожної з систем суб'єктам тестування було задано ряд питань щодо продуктивності кожної з версій. Здійснено аналіз результатів для оцінки кожної версії системи.

10. Представлено результати випробувань мультимедійної та немультимедійної версій створеної інтелімедійної системи в режимі інтелектуального тьютора-тренажера. Порівняння мультимедійної та немультимедійної версій системи виконано на основі введених показників, що беззаперечно визначило переваги мультимедійної версії системи.

ОСНОВНІ РЕЗУЛЬТАТИ РОБОТИ

1. Запропоновано основну ідею функціональності автоматизованої інтелектуальної системи на основі обмежень як ідею створення середовища підтримки прийняття рішень в умовах невизначеності при вирішенні оператором технологічних проблем на основі інтелектуальних засобів виявлення помилок і генерації зворотного зв'язку на основі мультимедійного контенту.

2. У результаті дослідження практичної діяльності операторів технологічних процесів у нафтогазовій предметній області, визначено, що більшість знань операторів мають декларативний характер (в той час як реальні задачі прийняття рішень вимагають процедурних алгоритмічних знань), що є потенційним джерелом помилок.

3. Запропоновано стратегію функціонування **інтелімедійної автоматизованої системи**, яка полягає в тому, що при рішенні технологічних проблем і відповідному виявленні помилки виконується модифікація процедурних знань шляхом включення порушеного правила, що визначає динаміку знань в орієнтованому перетворенні декларативних знань у процедурні. Основним елементом автоматизованої інтелектуальної системи на основі обмежень виділено обробку помилок у процесах виявлення, класифікації та корекції.

4. Запропонований формально-логічний апарат дозволяє моделювати технологічний процес на основі помилок. Відповідно, коректні рішення розглянуто як такі, що не порушують обмеження в накладеній на домен ієрархії обмежень. При рішенні технологічної проблеми об'єкт прийняття рішень переміщується певною послідовністю станів із множини можливих і в кожен момент часу знаходиться в певному виділеному стані, що є підмножиною однієї з технологічних проблем, які виникають в виділеному технологічному процесі. Накладання множини обмежень дозволило виділити класи еквівалентностей для множини станів технологічної проблеми. В межах класу еквівалентності система використовує одну й ту саму структуру та наповнення зворотного зв'язку, тому всі стани проблеми в межах класу еквівалентності розглядалися як тотожні. Технологічно повідомлення зворотного

зв'язку прикріпляється безпосередньо до обмежень на рівні множини, системи або ієрархії.

5. Представлено модель домену технологічної проблеми розглядається як сукупність правил виду: якщо умова релевантності з коефіцієнтом впевненості, тоді умова задоволення з коефіцієнтом впевненості. Коефіцієнт впевненості, в даному випадку інтерпретовано як коефіцієнт релевантності, як ваговий, імовірнісний, можливісний або коефіцієнт преференції.

6. Представлено системи на основі обмежень, які виконують оцінку рішень, запропонованих об'єктом прийняття рішень, шляхом співставлення їх із відповідною доменною моделлю, що визначається накладеною множиною обмежень. Представлена процедура виконується як послідовність кроків: виділення шаблонів умов релевантності і шаблонів умов задоволення; співставлення шаблонів релевантності з відповідним станом проблеми; перевірка умови задоволення релевантних обмежень; перевірка задоволеності (порушення) обмеження, шляхом співставлення шаблону його умови задоволення стану проблеми.

7. Представлено процедуру побудови моделей об'єкта прийняття рішень у поточній, короткотривалій і довготривалій формах.

8. Основним завданням **інтелектуальної автоматизованої системи** на основі обмежень визначено підтримку рішення технологічних проблем шляхом реалізації системи у вигляді доменно та предметно незалежної інтелектуальної оболонки на основі інтерфейсної реалізації; забезпечення підтримки рішення технологічних проблем засобами зворотного зв'язку з об'єктом; підтримку рішення технологічних проблем на основі ведення контекстозалежних та релевантних діалогів з користувачем.

9. Визначено, що представлення декларативних знань у формі обмежень високого рівня абстракції дозволяє виконувати ідентифікацію помилок, відповідно таких знань робить неможливим розрізнення хибних, коректних та релевантних знань з метою корекції помилок. Таким чином, у запропонованому підході роль системи обмежень до технологічних проблем зводилась до класифікації та аналізу помилок і формування зворотного зв'язку. Основною ідеєю механізму

функціонування автоматизованої інтелектуальної системи на основі обмежень визначено співставлення рішення об'єкту з множинами, системами й ієрархіями обмежень, що описують технологічну проблему.

10. Реалізовано інтелектуальні функції автоматизованої інтелектуальної системи на основі обмежень шляхом взаємодії системи з об'єктом прийняття рішень, з активацією релевантних доменних знань для об'єкту відповідно до профілю. Відповідно, ефективність зворотного зв'язку оцінювалася шляхом визначення місця виникнення помилки; ідентифікації та класифікації помилки, співвіднесення її з певною ієрархією помилок, виділення складових помилки та її можливих рівнів; виділення релевантних концепцій предметної області як основи побудови коректного рішення.

11. Виконано класифікацію рівнів зворотного зв'язку виду: 1) "так", "ні", "так:cf", "ні:cf", $cf \in [0..1]$; 2) прапорець помилок; 3) виведення повідомлення про суть помилки; 4) виведення частини правильного рішення; 5) виведення повного правильного рішення; 6) виведення всіх помилок; 7) візуалізація ієрархій помилок та порушених обмежень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Odegard S.I., Risvik B.T., Bjørkevoll, K., Mehus, O. and others. Advanced Dynamic Training Simulator For Drilling As Well As Related Experience From Training Of Drilling Teams With Focus On Realistic Downhole Feedback. *Distributed Computing*. 2013. URL: <https://www.semanticscholar.org/paper/Advanced-Dynamic-Training-Simulator-For-Drilling-As-Odegard-Risvik/48ee0a46a069f9fcb8d2eb202d2a991ea0ed9630>. (Date of access: 20.02.2021).
2. Li G., Müller M., Casser V., Smith N. and others. OIL: Observational Imitation Learning. URL: <https://arxiv.org/pdf/1803.01129.pdf>. (Date of access: 20.02.2021).
3. Alkamil E., Abbas A., Flori R., da Silva B. and others. Learning from Experience: Real-Time H2S Monitoring System Using Fuzzy ART Unsupervised Learning. 2018. URL: https://www.researchgate.net/publication/327405895_Learning_from_Experience_Real-Time_H2S_Monitoring_System_Using_Fuzzy_ART_Unsupervised_Learning. (Date of access: 20.02.2021).
4. Nwachukwu A., Jeong H., Sun A., Pyrcz M. Machine Learning-Based Optimization of Well Locations and WAG Parameters under Geologic Uncertainty. 2018. URL: https://www.researchgate.net/publication/324118445_Machine_Learning-Based_Optimization_of_Well_Locations_and_WAG_Parameters_under_Geologic_Uncertainty. (Date of access: 20.02.2021).
5. Asala H., Chebeir J., Zhu W., Taleghani A. A Machine Learning Approach to Optimize Shale Gas Supply Chain Networks. 2017. URL: https://www.researchgate.net/publication/320177437_A_Machine_Learning_Approach_to_Optimize_Shale_Gas_Supply_Chain_Networks. (Date of access: 20.02.2021).
6. Teixeira A.F., Secchi A.R. Machine learning models to support reservoir production optimization. *IFAC-PapersOnLine*. 2019. Volume 52. Issue 1. P. 498-501. URL: <https://www.sciencedirect.com/science/article/pii/S2405896319301971>. (Date of access: 20.02.2021).

7. Мырзахметов Б.А., Айтореева Г.К. Особенности применения «интеллектуальных» тренажоров-имитаторов в подготовке кадров для нефтегазовой промышленности. *Вестник КазНТУ*. 2012. №1(89). С. 140-147. URL: <https://official.satbayev.university/download/document/7162/ВЕСТНИК-2012%20№1.pdf>. (Дата звернення: 20.02.2021).
8. Горбійчук М.І., Кропивницька В.Б. Моделювання та ідентифікація процесу заглиблення свердловин. *Науковий вісник Івано-Франківського національного технічного університету нафти і газу*. 2004. №1(7). С. 9-13.
9. Чудик І.І., Бабій Р.Б. Оптимальна подача промивальної рідини на вибій при бурінні свердловини. *Нафтогазова енергетика*. 2007. №3(4). С.71-75.
10. Шавранський М.В. Система контролю для запобігання прихоплень бурильної колони в процесі буріння : дис. ... канд. техн. наук : 05.11.13. Івано-Франківськ: ІФНТУНГ, 2003. 168 с.
11. Вовк. Р.Б. Система інтелектуальної підтримки прийняття рішень для запобігання нештатних ситуацій в процесі буріння свердловин : дис. ... канд. техн. наук : 05.13.07. Івано-Франківськ: ІФНТУНГ, 2012. 265 с.
12. Семенцов Г.Н., Когуч Я.Р., Куровець Я.В., Дранчук М.М. Автоматизація технологічних процесів у нафтовій та газовій промисловості. Івано-Франківськ: ІФНТУНГ Факел, 2009. 300 с.
13. Семенцов Г.Н. Інтелектуальні системи керування технологічними процесами. Івано-Франківськ: ІФНТУНГ, 2012. 173 с.
14. Семенцов Г., Горбійчук М., Чигур І. Математичний аналіз критеріїв відпрацювання доліт. *Нафтова і газова промисловість*. 2001. №6. С. 25-28.
15. Семенцов Г.Н., Горбійчук М.І., Чигур І.І. Визначення часу відпрацювання долота за оснащенням. *Нафтова і газова промисловість*. 2001. №5. С. 20-25.
16. Крижанівський Є.І., Яким Р.С., Шмандровський Л.Є., Петрина Д.Ю. Критерії підвищення довговічності тришарошкових бурових доліт з опорами типу ВУ.

Науковий вісник Івано-Франківського національного технічного університету нафти і газу. 2011. №1(27). С. 31-38.

17. Семенцов Г.Н., Гутак О.В. Моделювання функції мети для системи адаптивного оптимального керування процесом буріння нафтових і газових свердловин долотами нового покоління. *Вісник Хмельницького нац. ун-ту*. 2010. №2. С. 133-140.
18. Дудля М.А., Карпенко В.М. Алгоритм адаптивної стабілізації навантаження на долото. *Науковий вісник: зб. наук. пр. Національної гірничої академії*. 2000. №4. С. 81-88.
19. Горбійчук М.І. Інформаційне забезпечення адаптивної системи керування процесом буріння. *Методи та прилади контролю якості*. 2002. №8. С. 86-89.
20. Горбійчук М.І., Кропивницька В.Б. Аналіз алгоритмів ідентифікації процесу заглиблення свердловин. *Нафтова і газова промисловість*. 2005. №2. С. 24-26.
21. Кропивницька В.Б. Комп'ютерна система оптимального керування процесом буріння нафтових і газових свердловин. *Науковий вісник Івано-Франківського національного технічного університету нафти і газу*. 2008. №1. С. 105-108.
22. Герасимов Б.М., Дивизнюк М.М., Субач І.Ю. Системы поддержки принятия решений: проектирование, применение, оценка / Севастополь: Научно-исследовательский центр вооруженных сил Украины «Государственный океанариум», 2004. 320 с.
23. Алексеев А.В., Борисов А.Н., Вилюмс Э.Р. Интеллектуальные системы принятия проектных решений. Рига: Зинатне, 1997. 206 с.
24. Алтунин А.Е., Семухин М.В. Модели и алгоритмы принятия решений в нечетких условиях. Тюмень: Изд-во Тюменского государственного университета, 2000. 352 с.
25. Ларичев О.И. Теория и методы принятия решений. М.: Логос, 2000. 296 с.
26. Фролов Ю.В. Интеллектуальные системы и управленческие решения. М., 2000. 122 с.

27. Исмаков Р.А., Рахматуллин Д.В., Мухаметгалиев И.Д. Применение виртуальной программы-тренажера для ЭВМ «Слайд Мастер 1.18» для обучения практическим навыкам бурения нефтяных и газовых скважин с использованием забойных телеметрических систем. Сборник материалов всероссийской научно-практической конференции «Современные технологии извлечения нефти и газа. Перспективы развития минерально-сырьевого комплекса (российский и мировой опыт)» (г. Ижевск, 26-27 мая 2016). Ижевск: Издательский центр «Удмуртский университет», 2016. С. 93-102. URL: <https://core.ac.uk/download/pdf/235148605.pdf#page=93>. (Дата звернення: 20.02.2021).
28. Исмаков Р.А., Хафизов А.Р., Мухаметгалиев И.Д., Гуменников С.Г. Анализ работы имитационных тренажерных комплексов для обучения практическим навыкам бурения. *Нефтегазовое дело*. 2016. Т. 14. №4. С. 9-13. URL: <http://ngdelo.ru/files/ngdelo/2016/4/ngdelo-4-2016-p9-13.pdf>. (Дата звернення: 20.02.2021).
29. Krasnov F., Glavnov N., Sitnikov A. A Machine Learning Approach to Enhanced Oil Recovery Prediction. *Analysis of Images, Social Networks and Texts*. 2018. Vol. 10716. P. 164-171. URL: https://link.springer.com/chapter/10.1007/978-3-319-73013-4_15. (Date of access: 20.02.2021).
30. Цветанская Д.С., Дружинская Е.В. Симулятор виртуальной реальности работника камеры пуска приема средств очистки и диагностики. *Наука настоящего будущего*. 2019. Том 1. С. 66-67. URL: <https://www.elibrary.ru/item.asp?id=41177334>. (Дата звернення: 20.02.2021).
31. Ащанулов А.В., Шумилина Н.А. VR-тренажёр в подготовке и обучении персонала для решения производственных задач. Материалы научно-практической конференции «Нефтегазовое производство – основа научно-технического прогресса и экономической стабильности (г. Оренбург, 22 января 2020)». Саратов: Общество с ограниченной ответственностью «Амирит», 2020. С. 28-32. URL: <https://www.elibrary.ru/item.asp?id=44009249>. (Дата звернення: 20.02.2021).

32. Котелева Н.И., Шабловский И.Е., Кошкин А.В. Компьютерные тренажеры для обучения операторов технологических процессов нефтегазовой отрасли: анализ существующих решений и пути их усовершенствования. *Записки Горного института*. 2011. Т. 192. С. 212-215. URL: [https://cyberleninka.ru/article/n/kompyuternye-trenazhery-dlya-obucheniya-operatorov-tehnologicheskikh-protsessov-neftegazovoy-otrasli-analiz-suschestvuyuschih/viewer](https://cyberleninka.ru/article/n/kompyuternye-trenazhery-dlya-obucheniya-operatorov-tehnologicheskikh-protsessov-neftegazovoy-otrasli-analiz-suschestvuyuschih-resheniy-i-puti-ih-usovershenstvovaniya). (Дата звернення: 20.02.2021).
33. Хафизов Ф.Ш., Арсланов А.Р., Шевченко Д.И., Кудрявцев А.А. Цели, требования и практические аспекты разработки современных технических средств обучения для специалистов трубопроводного транспорта. *Трубопроводный транспорт: теория и практика*. 2011. № 1(23). С. 12-19. URL: <https://www.elibrary.ru/item.asp?id=23465698>. (Дата звернення: 20.02.2021).
34. Берштейн Л.С., Боженюк А.В. Нечеткие модели принятия решений: дедукция, индукция, аналогия. Таганрог : Изд-во ТРТУ, 2001. 110 с.
35. Ногин В.Д. Принятие решений в многокритериальной среде. М.: Физматлит, 2002. 162 с.
36. Гаврилова Т.А., Червинская К.Р. Извлечение и структурирование знаний для экспертных систем. М.: Радио и связь, 1992. 200 с.
37. Шекета В.І., Демчина М.М., Процюк Г.Я. Класифікація об'єктів нафтогазової предметної області на основі шаблонів у формі правил . *Інтелектуальний аналіз інформації* : матеріали XII міжн. наук. конф. ім. Т.А.Таран (м. Київ, 16-18 травня 2012 р.). К., 2012. С. 233-239.
38. Семенцов Г.Н. Контроль технологічних параметрів і показників процесу буріння на базі системних технологій Data Mining DMPC. *Приладобудування - 2006 (стан і перспективи)* : зб. наук. праць п'ятої наук.-тех. конф. (м. Київ, 25-26 квітня 2006 р.). К., 2006. С. 254-255.
39. Юрчишин В.М., Шекета В.І., Юрчишин О.В. Інформаційне моделювання нафтогазових об'єктів : монографія. Івано-Франківськ: Вид-во Івано-Франківського нац. техн. ун-ту нафти і газу, 2010. 196 с.
40. Семенцов Г.Н., Чигур І.І., Шавранський М.В., Борин В.С. Фазі-логіка в системах

керування. Івано-Франківськ : ІФНТУНГ, 2002. 84 с.

41. Шекета В.І., Демчина М.М., Гобир Л.М. Управління процесом інтерпретації інформації про нафтогазові об'єкти на основі нечітких моделей. *Математичне та імітаційне моделювання систем*: матеріали XII міжн. наук.-практ. конф. (25-28 червня 2012 р.). Чернігів-Жукин, 2012. С. 43-46.
42. Горбійчук М.І., Кропивницька В.Б. Оптимальне керування процесом механічного буріння. *Нафтова і газова промисловість*. 2005. №3. С.20-22.
43. Семенцов Г.Н., Чигур І.І., Когутяк М.І., Чигур Л.Я. Інтелектуальний пристрій на нечіткій логіці для розпізнавання образів у бурінні. *Нафтогазова енергетика*. 2009. №1(10). С. 75-77.
44. Семенцов Г.Н., Кузь Т.Я. Інформаційна модель контролю питомих енерговитрат на поглиблення свердловини. *Науковий вісник Івано-Франківського національного технічного університету нафти і газу*. 2001. №1(1). С. 76-80.
45. Семенцов Г.Н., Чигур І.І. Нечіткі моделі технологічних процесів і їх ідентифікація. *Розвідка і розробка нафтових і газових родовищ*. Івано-Франківськ: ІФДТУНГ. 1996. Вип. 33(6). С. 12-17. (Серія: Технічна кібернетика та електрифікація об'єктів паливно-енергетичного комплексу).
46. Семенцов Г.Н., Матієвський Ю.В. Автоматизація технологічних процесів на базі методів нечіткої логіки. *Наука і газ України – 2004* : матеріали VIII міжнар. конф. (м. Львів, 29 вересня 2004 р.) Львів, 2004. С. 280-282.
Вовк Р.Б. Моделювання структури та функціональності технологічних проблем на основі обмежень. *Математичні машини та системи*. 2011. № 2. С. 153-161.
47. Гончаров А.С. Проектирование и разработка алгоритмического и программного обеспечения симулятора состояния подземного нефтяного резервуара в режиме реального времени: магистерская диссертация / Национальный исследовательский Томский политехнический университет (ТПУ), Инженерная школа информационных технологий и робототехники (ИШИТР), Отделение информационных технологий (ОИТ). Томск, 2018. 115 с. URL: <http://earchive.tpu.ru/handle/11683/47655>. (Дата звернення: 20.02.2021).

48. Мочалов Д.О., Законьшек Я.В., Шамис М.А. Применение комплексов моделирования в реальном времени для современных энергосистем нефтегазовых предприятий. *Экспозиция Нефть Газ*. 2014. №7 (39). С. 79-82. URL: <https://cyberleninka.ru/article/n/primenenie-kompleksov-modelirovaniya-v-realnom-vremeni-dlya-sovremennyh-energосistem-neftegazovyh-predpriyatiy>. (Дата звернення: 20.02.2021).
49. Юхин Е.Г., Кошелев Н.А., Хафизов А.М., Малышева О.С. Разработка виртуального тренажера-имитатора работы трубчатой печи для повышения профессиональных навыков сотрудников предприятий нефтегазовой отрасли. *Фундаментальные исследования*. 2015. № 12 (часть 5) С. 970-974. URL: <https://www.fundamental-research.ru/ru/article/view?id=39661>. (Дата звернення: 20.02.2021).
50. Григорьев Л.И., Кузьмицкий И.Ф., Санжаров В.В. Системный и синергетический анализ управления непрерывными технологическими процессами в нештатных ситуациях. *XII всероссийское совещание по проблемам управления*. М.: ВСПУ, 2014. С. 4285-4296. URL: <https://scholar.google.ru/citations?user=ya0fuSUAAAAJ&hl=ru>. (Дата звернення: 20.02.2021).
51. Хафизов Ф.Ш., Кудрявцев А.А., Шевченко Д. И. Общая концепция интегрированной обучающей системы для трубопроводного транспорта нефти. *Электронный научный журнал «Нефтегазовое дело»*. 2011. № 5. С. 476-489. URL: http://ogbus.ru/files/ogbus/authors/KhafizovFSh/KhafizovFSh_6.pdf. (Дата звернення: 20.02.2021).
52. Хафизов Ф.Ш., Кудрявцев А.А., Шевченко Д.И. Интегрированные обучающие системы для специалистов трубопроводного транспорта нефти. *Электронный научный журнал «Нефтегазовое дело»*. 2011. № 3. С. 356-371. URL: http://ogbus.ru/files/ogbus/authors/KhafizovFSh/KhafizovFSh_4.pdf. (Дата звернення: 20.02.2021).
53. Anderson J.R., Boyle C. F., Reiser B. J. Intelligent Tutoring Systems. *Science*. 26 Apr 1985. Vol. 228. Issue 4698. Pp. 456-462. URL:

<https://science.sciencemag.org/content/228/4698/456.abstract> (Date of access: 16.02.2021).

54. Vanlenh Kurt. The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*. 46:4. Pp. 197-221. URL: <https://www.tandfonline.com/doi/abs/10.1080/00461520.2011.611369> (Date of access: 16.02.2021).
55. Corbett A.T., Koedinger K.R., Anderson J.R. Intelligent Tutoring Systems // Handbook of Human-Computer Interaction. 1997. Pp. 849-874. URL: <https://www.sciencedirect.com/science/article/pii/B9780444818621501035> (Date of access: 16.02.2021).
56. Burns H.L., Capps G.C. Foundations of Intelligent Tutoring Systems / Edited By Martha C. Polson, J. Jeffrey Richardson. 1988. 292 p. URL: https://books.google.com.ua/books?hl=uk&lr=&id=yDm-6Xdtz_MC&oi=fnd&pg=PA1&dq=tutoring+systems&ots=iR7oCxqpz6&sig=uk_FbBgxOvdhOdTog-ZaQB9-wnE&redir_esc=y#v=onepage&q=tutoring%20systems&f=false (Date of access: 16.02.2021).
57. Шутый С.А., Харькова И.В. Информационные образовательные технологии: интеллектуальные системы обучения. *Электронное обучение в непрерывном образовании*. 2017. №1. С. 191-1193. URL: <https://www.elibrary.ru/item.asp?id=29120666> (дата звернення 16.02.2021).
58. Алешева Л.Н. Интеллектуальные обучающие системы. *Вестник университета*. 2018. №1. С. 149-155. URL: <https://cyberleninka.ru/article/n/intellektualnye-obuchayuschie-sistemy/viewer> (дата звернення 16.02.2021).
59. Басалин П.Д., Белоусова И.И. Интерактивные формы обучения в образовательном процессе. *Инновации в образовании. Вестник Нижегородского университета им. Н.И. Лобачевского*. 2014. № 3(4). С. 18-21. URL: <https://cyberleninka.ru/article/n/interaktivnye-formy-obucheniya-v-obrazovatelnom-protssesse/viewer> (дата звернення 16.02.2021).

60. Рыбина Г.В. Интеллектуальные обучающие системы на основе интегрированных экспертных систем: опыт разработки и использования. *Информационно-измерительные и управляющие системы*. 2011. Том 9. №10. С. 4-16. URL: <https://elibrary.ru/item.asp?id=17256995> (дата звернения 16.02.2021).
61. Попова Ю.Б. Интеллектуальная составляющая автоматизированной системы обучения CATS. *Образовательные технологии и общество*. 2019. Вып. 4. Том 22. С. 24-37. URL: <https://cyberleninka.ru/article/n/intellektualnaya-sostavlyayuschaya-avtomatizirovannoy-sistemy-obucheniya-cats/viewer> (дата звернения 16.02.2021).
62. Manton E., Turner C.T., English D. Testing the level of student knowledge. *Education*. 2004. Vol. 124. Issue 4. P. 682-687. URL: <https://web.b.ebscohost.com/abstract?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=00131172&AN=14053938&h=iB28IYq058KSNOb%2b36L%2bp33Z2iUeSjSuAqdWQeyfzkE9yf7Quc10jWVgW7XJo0hcX3QkSEl6T7j31v2VeoEROg%3d%3d&crl=c&resultNs=AdminWebAuth&resultLocal=ErrCrlNotAuth&crlhashurl=logi.n.aspx%3fdirect%3dtrue%26profile%3dehost%26scope%3dsite%26authtype%3dcrawler%26jrnl%3d00131172%26AN%3d14053938>. (Date of access: 18.02.2021).
63. Čisar S.M., Radosav D., Markoski B., Pinter R. Computer Adaptive Testing of Student Knowledge. *Acta Polytechnica Hungarica*. 2010. Vol. 7. No. 4. P. 139-152. URL: http://acta.uni-obuda.hu/Cisar_Radosav_Markoski_Pinter_Cisar_25.pdf. (Date of access: 18.02.2021).
64. Heinrich C. Virtual Case Studies in the Classroom Improve Student Knowledge. *Clinical Simulation in Nursing*. October 2012. Vol. 8. Issue 8. P. e353-e361. URL: <https://www.sciencedirect.com/science/article/abs/pii/S1876139911000284>. (Date of access: 18.02.2021).
65. Morehead K., Rhodes M.G., DeLozier S. Instructor and student knowledge of study strategies. *Memory*. 2016. Vol. 24. P. 257-271. URL: <https://www.tandfonline.com/doi/abs/10.1080/09658211.2014.1001992>. (Date of access: 18.02.2021).

66. Corbett A., McLaughlin M., Scarpinato K.C. Modeling Student Knowledge: Cognitive Tutors in High School and College. *User Modeling and User-Adapted Interaction*. 2000. 10. P. 81-108. URL: <https://link.springer.com/article/10.1023/A:1026505626690>. (Date of access: 18.02.2021).
67. Guzman E., Conejo R., Perez-de-la-Cruz J. Improving Student Performance Using Self-Assessment Tests. *Intelligent Systems*. July-Aug. 2007. Vol. 22. No. 4. P. 46-52. URL: <https://ieeexplore.ieee.org/abstract/document/4287273>. (Date of access: 18.02.2021).
68. Thomas Rist. Intellimedia Systems: Researc Vol.at the Intersection of Multimedia and Artificial Intelligence. *Pacific Rim International Conference on Artificial Intelligence*. 2002. Vol. 2417. Pp. 9-18. https://link.springer.com/chapter/10.1007/3-540-45683-X_4 (Date of access: 16.02.2021)
69. Brøndsted T., Dalsgaard P., Larsen L. B., Manthey M., Kevitt P. M., Moeslund T. B., Olesen K. G. The IntelliMedia WorkBench-An Environment for Building Multimodal Systems. *International Conference on Cooperative Multimodal Communication*. 2001. Vol. 2155. Pp. 217-233. https://link.springer.com/chapter/10.1007/3-540-45520-5_13 (Date of access: 16.02.2021)
70. Brøndsted T., Larsen L. B., Manthey M., Kevitt P. M., Moeslund T. B., Olesen K. G. Developing Intelligent Multimedia Applications. *Multimodality in Language and Speech Systems*. 2002. Vol. 19. Pp. 19-171. https://link.springer.com/chapter/10.1007/978-94-017-2367-1_7 (Date of access: 16.02.2021)
71. McQuiggan S. W., Mott B. W., Lester J. C. Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User Modeling and User-Adapted Interaction*. 2008. Pp. 81-123. <https://link.springer.com/article/10.1007/s11257-007-9040-y> (Date of access: 16.02.2021)
72. Kevitt P. Mc. Intelligent Multimedia at the Imagineering Quarter. *International Conference on Contemporary Computing*. 2011. Vol. 168. Pp. 3-3.

https://link.springer.com/chapter/10.1007/978-3-642-22606-9_3 (Date of access: 16.02.2021)

73. Beun Robbert-Jan, Bunt H. Multimodal Cooperative Communication. *International Conference on Cooperative Multimodal Communication*. 2001. Vol. 2155. Pp. 1-10. https://link.springer.com/chapter/10.1007/3-540-45520-5_1
https://link.springer.com/chapter/10.1007/978-3-642-22606-9_3 (Date of access: 16.02.2021)
74. Devedžić V. Teaching Knowledge Modeling at the Graduate Level – A Case Study. *Innovative Teaching and Learning*. 2000. Vol. 36. Pp. 135-188. https://link.springer.com/chapter/10.1007/978-3-7908-1868-0_5 (Date of access: 16.02.2021)
75. Юрчишин В.М., Шекета В.І., Юрчишин О.В. Інформаційне моделювання нафтогазових об'єктів: монографія. Івано-Франківськ: Вид-во Івано-Франківського нац. техн. ун-ту нафти і газу. 2010. 196 с.
76. Bulatov A., Krokhin A., Larose B. Dualities for Constraint Satisfaction Problems. *Complexity of Constraints, LNCS 5250*. 2008. Pp. 93-124. URL: <http://community.dur.ac.uk/andrei.krokhin/papers/dualsurvey.pdf> (Date of access: 16.02.2021)
77. Dundua B., Florindo M., Kutsia T., Marin M. CLP(H): Constraint logic programming for hedges. *Theory and Practice of Logic Programming*. Vol.16. Issue 2. March 2016. Pp. 141–162. URL: <https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/abs/clph-constraint-logic-programming-for-hedges/CF9E9B13FA39D4A5CB501FD0AA4FD39C> . (Date of access: 18.02.2021).
78. Nasserini S.H., Behmanesh E., Taleshian F., Abdolalipoor M., Taghi Nezhad N.A. Fully fuzzy linear programming with inequality constraints. *International Journal of Industrial Mathematics*. Fall 2013. Vol. 5. Number 4. P. 309-316. URL: https://scholar.googleusercontent.com/scholar?q=cache:UXffQ4zxLgYJ:scholar.google.com/+fuzzy+constraints&hl=uk&as_sdt=0,5&as_ylo=2012&as_yhi=2016. (Date of access: 18.02.2021).

79. Mula J., Poler R., Garcia-Sabater J.P. Material Requirement Planning with fuzzy constraints and fuzzy coefficients. *Fuzzy Sets and Systems*. 2007. Volume 158. Issue 7. P. 783-793. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0165011406004970>. (Date of access: 18.02.2021).
80. Гуляницький Л.Ф., Рясна І.І. До формалізації задач оптимізації на нечітких множинах. Теорія оптимальних рішень: Зб. наук. пр. 2016. №2016. С. 17-25. URL: http://scholar.googleusercontent.com/scholar?q=cache:pvwBR0K8ZrgJ:scholar.google.com/+нечіткі+обмеження&hl=uk&as_sdt=0,5&as_ylo=2012&as_yhi=2016. (Дата звернення: 18.02.2021).
81. Jiménez M., Arenas M., Bilbao A., Rodri'guez V.M. Linear programming with fuzzy parameters: An interactive method resolution. *European Journal of Operational Research*. 2007. Volume 177. Issue 3. P. 1599-1609. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0377221705006454>. (Date of access: 18.02.2021).
82. Hofstedt P., Pepper P. Integration of declarative and constraint programming. *Theory and Practice of Logic Programming*. 2007. Vol. 7. Issue 1-2. P. 93-121. URL: <https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/abs/integration-of-declarative-and-constraint-programming/8FEA4C998E36AFDDBB8283CAA446DE0E1>. (Date of access: 18.02.2021).
83. Вовк Р.Б., Шекета В.І. Automated Intelligent System for Unscheduled Situations Control in Drilling. *Oil and Gas Power Engineering*. No. 3(16). P. 108-120. URL: <https://nge.nung.edu.ua/index.php/ngc/article/view/78>. (Дата звернення: 18.02.2021).
84. Kuipers F., Korkmaz T., Krunz M., Mieghem P. Van. Performance evaluation of constraint-based path selection algorithms. *Network*. Sept.-Oct. 2004. Vol. 18. No. 5. P. 16-23. URL: <https://ieeexplore.ieee.org/abstract/document/1337731>. (Date of access: 18.02.2021).
85. Vu X., O'Sullivan B. Semiring-Based Constraint Acquisition. *19th IEEE International Conference on Tools with Artificial Intelligence (Patras, Greece, 2007)*. ICTAI' 2007.

- P. 251-258. URL: <https://ieeexplore.ieee.org/abstract/document/4410291>. (Date of access: 18.02.2021).
86. Sachenbacher M., Williams B. Diagnosis as Semiring-based Constraint Optimization. *Proceedings of the 16th European Conference on Artificial Intelligence (Valencia, Spain, August 22-27, 2004). ECAI' 2004*. URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.3034&rep=rep1&type=pdf>. (Date of access: 18.02.2021).
87. Bistarelli S., Montanari U., Rossi F. Semiring-based constraint logic programming: syntax and semantics. *ACM Transactions on Programming Languages and Systems*. January 2001. Vol. 23. No. 1. P. 1-29. URL: <https://dl.acm.org/doi/abs/10.1145/383721.383725>. (Date of access: 18.02.2021).
88. Wang Y.-M., Chin K.-S. Fuzzy analytic hierarchy process: A logarithmic fuzzy preference programming methodology. *International Journal of Approximate Reasoning*. 2011. Volume 52. Issue 4. P. 541-553. URL: <https://www.sciencedirect.com/science/article/pii/S0888613X10001611>. (Date of access: 18.02.2021).
89. Chen Y.-Y., Lin J.T. Hierarchical multi-constraint production planning problem using linear programming and heuristics. *Journal of the Chinese Institute of Industrial Engineers*. 2008. Vol. 25. No. 5. P. 347-357. URL: <https://www.tandfonline.com/doi/abs/10.1080/10170660809509098>. (Date of access: 18.02.2021).
90. Sundström O., Ambühl D. and Guzzella L. On Implementation of Dynamic Programming for Optimal Control Problems with Final State Constraints. *Oil & Gas Science and Technology*. 2010. Vol. 65. No. 1. P. 91-102. URL: <https://ogst.ifpenergiesnouvelles.fr/articles/ogst/abs/2010/01/ogst09017/ogst09017.html>. (Date of access: 18.02.2021).
91. Семко В.В. Логіко-математична модель опису простору рішень. *Вісник Чернігівського державного технологічного університету*. 2013. № 2 (65). С. 147-156. URL: <http://scholar.googleusercontent.com/scholar?q=cache:oELSNvmTEYYJ:scholar.google.com/+прост>

ip+рiшень&hl=uk&as_sdt=0,5&as_ylo=2008&as_yhi=2015&as_vis=1. (Дата звернення: 18.02.2021).

92. Borges M., Filieri A., d'Amorim M., Păsăreanu C.S., Visser W. Compositional solution space quantification for probabilistic software analysis. *35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, June 2014. PLDI' 14. P. 123-132. URL: <https://dl.acm.org/doi/abs/10.1145/2594291.2594329>. (Date of access: 18.02.2021).
93. De Raedt L., Tias G., Siegfried N. Constraint programming for itemset mining. *14th ACM SIGKDD international conference on Knowledge discovery and data mining*, (Las Vegas, Nevada, USA, August 2008). KDD' 2008. P. 204-212. URL: <https://dl.acm.org/doi/abs/10.1145/1401890.1401919>. (Date of access: 18.02.2021).
94. Bessiere C. Constraint Propagation. *Foundations of Artificial Intelligence*. 2006. Vol. 2. Pp. 29-83. URL: <https://www.sciencedirect.com/science/article/pii/S1574652606800076>. (Date of access: 18.02.2021).
95. Вовк Р.Б. Контроль поиска решений технологических проблем процесса бурения, формализованных наложением иерархии ограничений. *Вісник Кременчуцького національного університету ім. Михайла Остроградського*. 2014. № 1. С. 76-82. URL: <https://www.elibrary.ru/item.asp?id=21451664>. (Дата звернення: 18.02.2021).
96. Hebrard E., Hnich B., O'Sullivan B., Walsh T. Finding Diverse and Similar Solutions in Constraint Programming. *The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, (Pittsburgh, Pennsylvania, USA, July 9-13, 2005). 2005. URL: <https://www.aaai.org/Papers/AAAI/2005/AAAI05-059.pdf>. (Date of access: 18.02.2021).
97. Crain. T., Gramoli V., Raynal M. A Contention-Friendly Binary Search. *Euro-Par 2013 Parallel Processing*. Euro-Par 2013. Lecture Notes in Computer Science. 2013. Vol 8097. Springer, Berlin, Heidelberg Tree. URL: https://link.springer.com/chapter/10.1007/978-3-642-40047-6_25. (Date of access: 18.02.2021).
98. Hladik P.-E., Cambazard H., Déplanche A.-M., Jussien N. Solving a real-time allocation problem with constraint programming. *Journal of Systems and Software*. 2018. Volume 81. Issue 1. P. 132-149. URL:

<https://www.sciencedirect.com/science/article/abs/pii/S0164121207000672>. (Date of access: 18.02.2021).

99. Apt, K., Monfroy E. Constraint programming viewed as rule-based programming. *Theory and Practice of Logic Programming*. 2001. Vol.1. Issue 6. P. 713-750. URL: <https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/abs/constraint-programming-viewed-as-rulebased-programming/5FC292D8AE02AC85ECDF650CC7EE5DB>. (Date of access: 18.02.2021).
100. Zhu Li. A Decision Method for Over-constraint of Parametric Model. URL: https://en.cnki.com.cn/Article_en/CJFDTotol-JZGC201020101.htm. (Date of access: 18.02.2021).
101. Jagadeesan R., Marrero W., Pitcher C., Saraswat V. Timed constraint programming: a declarative approach to usage control. *7th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming*, (Lisbon, Portugal, July 11-13 2005). PPDP' 2005. URL: <https://dl.acm.org/doi/abs/10.1145/1069774.1069790>. (Date of access: 18.02.2021).
102. Gelfond M., Mellarkod V.S., Zhang Y. Systems integrating answer set programming and constraint programming. *Annals of Mathematics and Artificial Intelligence*. August 2008. Vol. 53. Issue 1-4. P. 251-287. URL: <http://redwood.cs.ttu.edu/~yuazhang/publications/aspclp-lash08.pdf>. (Date of access: 18.02.2021).
103. Вовк Р.Б., Процюк Г.Я. Моделювання класів технологічних проблем на основі обмежень. *Математичні машини і системи*. 2011. №2. С. 153-161. URL: http://scholar.googleusercontent.com/scholar?q=cache:32CXjZycm4oJ:scholar.google.com/+часткове+впорядкування+змінних+&hl=uk&as_sdt=0,5&as_ylo=2007&as_yhi=2015. (Дата звернення: 18.02.2021).
104. Вовк Р., Процюк В., Шекета В. Формальний опис процесу контролю задоволення та порушення обмежень в інтелектуальних системах. *Вісник Національного університету «Львівська політехніка»*. 2011. №694 Комп'ютерні

науки та інформаційні технології. С. 189-199. URL: <http://ena.lp.edu.ua/handle/ntb/10505>. (Дата звернення: 18.02.2021).

105. Rossi F., Venable K.B., Walsh T. Preferences in Constraint Satisfaction and Optimization. *AI Magazine*. Vol. 29. No. 4. P. 58-68. URL: https://scholar.googleusercontent.com/scholar?q=cache:sy_FYhi1FFoJ:scholar.google.com/+constraint+based+preferential+optimization&hl=uk&as_sdt=0,5&as_ylo=2007&as_yhi=2015. (Date of access: 18.02.2021).
106. Thiele L., Miettinen K., Korhonen P.J., Molina J. A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization. *Evolutionary Computation*. 2009. Vol. 17. No. 3. P. 411-236. URL: <https://www.mitpressjournals.org/doi/abs/10.1162/evco.2009.17.3.411>. (Date of access: 18.02.2021).
107. Junker U. Preferred Explanations and Relaxations for Over-Constrained Problems. P. 167-172. URL: <https://www.aaai.org/Papers/AAAI/2004/AAAI04-027.pdf>. (Date of access: 18.02.2021).
108. De Raedt L., Kersting K. Probabilistic Inductive Logic Programming. *Lecture Notes in Computer Science*. 2008. Vol 4911. P. 1-27. URL: https://link.springer.com/chapter/10.1007/978-3-540-78652-8_1. (Date of access: 18.02.2021).
109. Hong Yan, Zhenxin Yu, T.C. Edwin Cheng. A strategic model for supply chain design with logical constraints: formulation and solution. *Computers & Operations Research*. 2003. Volume 30. Issue 14. P. 2135-2155. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0305054802001272>. (Date of access: 18.02.2021).
110. Hooker J.N. Logic, Optimization, and Constraint Programming. *INFORMS Journal on Computing*. 2002. Vol. 14. No. 4. P. 293-420. URL: <https://pubsonline.informs.org/doi/abs/10.1287/ijoc.14.4.295.2828>. (Date of access: 18.02.2021).
111. Schimpf O., Shen K. ECLiPSe – From LP to CLP. *Theory and Practice of Logic Programming*. Vol.12. Special Issue 1-2: Prolog Systems. 2012. Pp. 127-156. URL:

<https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/abs/eclipse-from-lp-to-clp/3AA1A4B6D1553CA2976C9D0856C16200> (Date of access: 18.02.2021).

112. Swift T., D. S. Warren D. S. XSB: Extending Prolog with Tabled Logic Programming. *Theory and Practice of Logic Programming*. Vol.12. Special Issue 1-2: Prolog Systems. January 2012. Pp. 157–187. URL: <https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/abs/xsb-extending-prolog-with-tabled-logic-programming/260AA88A3FFE3EAEB6A5CF6B01FC7EB5>. (Date of access: 18.02.2021).
113. Zhu H., Liu D., Zhang S., Teng S., Zhu Y. Solving the Group Multirole Assignment Problem by Improving the ILOG Approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. Vol. 47. Issue: 12. Dec. 2017. Pp. 3418 – 3424. URL: <https://ieeexplore.ieee.org/abstract/document/7480954>. (Date of access: 18.02.2021).
114. Lienhardt M., Lanese I., Mezzina C. A., Stefani J.-B. A Reversible Abstract Machine and Its Space Overhead. *International Conference on Formal Techniques for Distributed Systems*. *Formal Techniques for Distributed Systems*. Vol. 7273. 2012. Pp. 1-17. URL: https://link.springer.com/chapter/10.1007/978-3-642-30793-5_1. (Date of access: 18.02.2021).
115. CHARAF M. EL H., BENATTOU M., AZZOUZIA S. JESS AGENT Based Architecture for Testing Distributed Systems. *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING*. 2014. Pp. 1619-1634. URL: https://www.researchgate.net/profile/My-El-Hassan-Charaf/publication/273141924_A_JESS_AGENT_based_architecture_for_testing_distributed_systems/links/551c22480cf2909047ba24e1/A-JESS-AGENT-based-architecture-for-testing-distributed-systems.pdf. (Date of access: 18.02.2021).
116. Heinz S., Schulz J. Explanations for the Cumulative Constraint: *An Experimental Study*. *Experimental Algorithms*. 2011. Vol 6630. P. 400-409. URL: https://link.springer.com/chapter/10.1007/978-3-642-20662-7_34. (Date of access: 18.02.2021).

117. Fischetti M., Salvagnin D. An In-Out Approach to Disjunctive Optimization. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. 2010. Vol 6140. P. 136-140. URL: https://link.springer.com/chapter/10.1007/978-3-642-13520-0_17. (Date of access: 18.02.2021).
118. Schutt, A., Feydy, T., Stuckey, P.J. Explaining the cumulative propagator. *Constraints*. 2011. Vol. 16. P. 250-282. URL: <https://link.springer.com/article/10.1007%2Fs10601-010-9103-2>. (Date of access: 18.02.2021).
119. Rach S., Ufer S., Heinze A. Learning from errors: effects of teachers training on students' attitudes towards and their individual use of errors. *PNA*. 2013. Vol. 8. No. 1. P. 21-30. URL: https://scholar.googleusercontent.com/scholar?q=cache:Ay3YzM7bzMJ:scholar.google.com/+learning+from+errors&hl=uk&as_sdt=0,5&as_ylo=2007&as_yhi=2015. (Date of access: 18.02.2021).
120. Piteira M., Costa C. Learning computer programming: study of difficulties in learning programming. *International Conference on Information Systems and Design of Communication*, July 2013. P. 75-80. URL: <https://dl.acm.org/doi/abs/10.1145/2503859.2503871>. (Date of access: 18.02.2021).
121. Lewis F.L., Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control. *Circuits and Systems Magazine*. 2009. Vol. 9. No. 3. P. 32-50. URL: <https://ieeexplore.ieee.org/abstract/document/5227780>. (Date of access: 18.02.2021)
122. Molzahn D. K., Roald L.A. Grid-Aware versus Grid-Agnostic Distribution System Control: A Method for Certifying Engineering Constraint Satisfaction. *Proceedings of the 52nd Hawaii International Conference on System Sciences*. 2019. Pp. 3445 – 3454. URL: <http://128.171.57.22/bitstream/10125/59780/1/0340.pdf>. (Date of access: 18.02.2021).

123. Calafiore G.C., El Ghaoui L. Linear Programming with Probability Constraints - Part 1. 2007 *American Control Conference*. 2007. P. 2636-2641. URL: <https://ieeexplore.ieee.org/abstract/document/4282190>. (Date of access: 18.02.2021).
124. Kaňková V. A Note on Multistage Stochastic Programming with Individual Probability Constraints. *Operations Research Proceedings*. 2001. Vol. 2000. P. 91-96. URL: https://link.springer.com/chapter/10.1007/978-3-642-56656-1_15. (Date of access: 18.02.2021).
125. Walsh T. Stochastic Constraint Programmin. 2000. P. 129-135. URL: <https://www.aaai.org/Papers/Symposia/Fall/2001/FS-01-04/FS01-04-020.pdf>. (Date of access: 18.02.2021).
126. Zou L., Xu K., Yu J.X., Chen L. etc. Efficient processing of label-constraint reachability queries in large graphs. *Information Systems*. 2014. Volume 40. P. 47-66. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0306437913001257>. (Date of access: 18.02.2021).
127. Chwatal A.M., Raidl G.R. Solving the Minimum Label Spanning Tree Problem by Mathematical Programming Techniques. *Advances in Operations Research*. 2011. URL: https://downloads.hindawi.com/journals/aor/2011/143732.pdf?ev=pub_ext_btn_xdl. (Date of access: 18.02.2021).
128. Jin R., Hong H., Wang H., Ruan N. ect. Computing label-constraint reachability in graph databases. *International Conference on Management of Data*, (Indianapolis, Indiana, USA, June 6-10, 2010). ACM SIGMOD' 2010. P. 123-134. URL: <https://dl.acm.org/doi/abs/10.1145/1807167.1807183>. (Date of access: 18.02.2021).
129. Leone M., Sumedha, Weigt M., Clustering by soft-constraint affinity propagation: applications to gene-expression data. *Bioinformatics*. 15 October 2007. Volume 23. Issue 20. P. 2708-2715. URL: <https://academic.oup.com/bioinformatics/article/23/20/2708/230273?login=true>. (Date of access: 18.02.2021).
130. Berrada I., Ferland J.A., Michelon P. A multi-objective approach to nurse scheduling with both hard and soft constraints. *Socio-Economic Planning Sciences*. 1996. Volume

30. Issue 3. P. 183-193. URL: <https://www.sciencedirect.com/science/article/abs/pii/0038012196000109>. (Date of access: 18.02.2021).
131. Guadarrama S., Muñoz S., Vaucheret C. Fuzzy Prolog: a new approach using soft constraints propagation. *Fuzzy Sets and Systems*. 2004. Volume 144. Issue 1. P. 127-150. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0165011403004494>. (Date of access: 18.02.2021).
132. Balduccini M. Representing Constraint Satisfaction Problems in Answer Set Programming. 2009. URL: https://www.researchgate.net/profile/Marcello-Balduccini/publication/228379886_Representing_Constraint_Satisfaction_Problems_in_Answer_Set_Programming/links/02e7e528e01b59a90f000000/Representing-Constraint-Satisfaction-Problems-in-Answer-Set-Programming.pdf. (Date of access: 18.02.2021).
133. Barto L., Kozik M. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *Journal of the ACM*. January 2014. Article No.: 3. URL: <https://dl.acm.org/doi/abs/10.1145/2556646>. (Date of access: 18.02.2021).
134. Koster A. van Hoesel S., Kolen A. Solving partial constraint satisfaction problems with tree decomposition. *Networks*. 2002. Vol. 40(3). P. 170-180. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.10046>. (Date of access: 18.02.2021).
135. Le. N.-T., Wolfgang M. Using Weighted Constraints to Diagnose Errors in Logic Programming – The Case of an Ill-defined Domain. *International Journal of Artificial Intelligence in Education*. 2009. Vol. 19. No. 4. P. 381-400. URL: <https://content.iospress.com/articles/international-journal-of-artificial-intelligence-in-education/jai19-4-04>. (Date of access: 18.02.2021).
136. Gómez-López M.T., Gasca R.M. Using Constraint Programming in Selection Operators for Constraint Databases. *Expert Systems with Applications*. 2014. Vol. 41. Issue 15. P. 6773-6785. URL: <https://www.sciencedirect.com/science/article/abs/pii/S095741741400270X>. (Date of access: 18.02.2021).

137. Bartak R. A Theoretical Framework for Constraint Hierarchy Solvers. *Proceedings of the 15th European Conference on Artificial Intelligence* (Lyon, France, July 2002). ECAI' 2002. P. 146-150. URL: <https://books.google.com.ua/books?id=5ZuuF0ogxU4C&pg=PA147&lpg=PA147&dq=global+comparator+set+solves&source=bl&ots=e1hm4mO4KT&sig=ACfU3U0iLiF5qSLhPUrALa0VySsAnUiJ3w&hl=uk&sa=X&ved=2ahUKEwjHxNzgyYLVAhWwtIsKHe4hCZoQ6AEwBHoECAyQAw#v=onepage&q=global%20comparator%20set%20solves&f=false>. (Date of access: 18.02.2021).
138. Понкин И.В., Редькина А.И. Методология науки: абдукция; гипотеза. *Копирайт*. 2019. №3. С. 32-58. URL: <https://www.elibrary.ru/item.asp?id=41245495>. (Дата звернення: 18.02.2021).
139. Гребенников В.В., Понкин И.В., Редькина А. И. Метод абдукции как метод научного исследования. *Образование и право*. 2017. №5. URL: <https://cyberleninka.ru/article/n/metod-abduktsii-kak-metod-nauchnogo-issledovaniya>. (Date of access: 18.02.2021).
140. Rahmani F., Leifels K. Abductive Grounded Theory: a worked example of a study in construction management. *Construction Management and Economics*. 2018. Vol. 36. Issue 10. P. 565-583. URL: <https://www.tandfonline.com/doi/abs/10.1080/01446193.2018.1449954>. (Date of access: 18.02.2021).
141. Philipsen K. Theory Building: Using Abductive Search Strategies. *Collaborative Research Design*. 2018. P. 45-71. URL: https://link.springer.com/chapter/10.1007/978-981-10-5008-4_3. (Date of access: 18.02.2021).
142. Timmermans S. Abductive analysis. *Kwalon*. 2016. Vol. 21. P. 66-67. URL: https://www.tijdschriftkwalon.nl/inhoud/tijdschrift_artikel/KW-21-2-16/Abductive-analysis. (Date of access: 18.02.2021).
143. Awuzie B., McDermott P. An abductive approach to qualitative built environment research: A viable system methodological exposé. *Qualitative Research Journal*. Vol.

17. Issue 4. P. 356-372. URL: <https://www.emerald.com/insight/content/doi/10.1108/QRJ-08-2016-0048/full/html>. (Date of access: 18.02.2021).
144. Soler-Toscano F., Fernandez-Duque D., Nepomuceno-Fernandez A. A modal framework for modelling abductive reasoning. *Logic Journal of the IGPL*. Apr. 2012. Vol. 20. No. 2. P. 438-444. URL: <https://academic.oup.com/jigpal/article-abstract/20/2/438/828386>. (Date of access: 18.02.2021).
145. Alrajeh A., Fearfull A., Monk E. Qualitative Research Process Using Abductive Approach. 2012. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2276609. (Date of access: 18.02.2021).
146. Caroprese L., Zumpano E. Indefinite abductive explanations. 2019. P. 233-254. URL: <https://www.tandfonline.com/doi/abs/10.1080/11663081.2019.1624349>. (Date of access: 18.02.2021).
147. Komosinski M., Kups A., Urbanski M. Multi-criteria Evaluation of Abductive Hypotheses: Towards Efficient Optimization in Proof Theory. *18th International Conference on Soft Computing*, January 2012. P. 320-325. URL: https://www.researchgate.net/profile/Mariusz-Urbanski/publication/245022811_Multi-criteria_evaluation_of_abductive_hypotheses_Towards_efficient_optimization_in_proof_theory/links/02e7e51d593c9c9a0a000000/Multi-criteria-evaluation-of-abductive-hypotheses-Towards-efficient-optimization-in-proof-theory.pdf. (Date of access: 18.02.2021).
148. Hobbs J., Stickel M.E., Appelt D. et al. Interpretation as abduction. *Artificial Intelligence*. 1993. 63(1-2). P. 69-142.
149. Kakas A.C., Kowalski R.A., Toni F. Abductive logic programming. *Journal of Logic and Computation*. 1993. V. 2(6). P. 719-770,
150. Вагин В.Н., Хотимчук К.Ю. Абдукция в задачах планирования работ в сложных объектах. *Искусственный интеллект и принятие решений*. М.: Ленанд, 2011. Т.1. С. 3-13.
151. Kakas A.C., Kowalski R.A., Toni F. The role of Abduction in Logic Programming. *Handbook in Artificial Intelligence and Logic Programming*. 1998. V. 5. P. 235-324.

152. Kakas A.C., Michael A., Mourlas C. Aclp: Abductive constraint logic programming. *Journal of Logic Programming*. 2000. V. 44 (1-3). P. 129-177.
153. Kakas A.C., Riguzzi F. Abductive concept learning. *New Generation Computing*. 2000. V. 18(3). P. 243-294.
154. Hobbs J., Stickel M.E., Appelt D. et al. Interpretation as abduction. *Artificial Intelligence*. 1993. 63(1-2). P. 69-142.
155. Kakas A.C., Abductive logic programming. *Journal of Logic and Computation*. 1993. V. 2(6). P. 719-770.
156. Вагин В.Н., Хотимчук К.Ю. Абдукция в задачах планирования работ в сложных объектах. *Искусственный интеллект и принятие решений*. М.: Ленанд, 2011. Т.1. С. 3-13.
157. Kakas A.C., Kowalski R.A., Toni F. The role of Abduction in Logic Programming. *Handbook in Artificial Intelligence and Logic Programming*. 1998. V. 5. P. 235-324.
158. Kakas A.C., Michael A., Mourlas C. Aclp: Abductive constraint logic programming. *Journal of Logic Programming*. 2000. V. 44 (1-3). P. 129-177.
159. Kakas A.C., Riguzzi F. Abductive concept learning. *New Generation Computing*. 2000. . V. 18(3). .P. 243-294.
160. Mancarella P., Terreni G. An abductive proof procedure handling active rules. *AI*IA 2003: Advances in Artificial Intelligence*. SV of LNAI, 2003. P. 105-117.
161. Мельник В. Д. Абдуктивне виведення знань про процес буріння на основі мультимедійних даних про бурове обладнання. *Науковий вісник Івано-Франківського національного технічного університету нафти і газу*. 2016. №1(40). С.80-91.
162. Берштейн Л.С., Боженюк А.В. Нечеткие модели принятия решений: дедукция, индукция, аналогия. Таганрог : Изд-во ТРТУ, 2001. 110 с.
163. Шекета В.І., Демчина М.М., Гобир Л.М. Abduction Application in Problems of Classification of Data About Oil and Gas Facilities. *Oil and Gas Power Engineering*. 2014. No. 2(22). P. 86-97. URL: <https://nge.nung.edu.ua/index.php/ngc/article/view/323>. (Date of access: 18.02.2021).

164. Vanderhaegen F., Caulier P. A multi-viewpoint system to support abductive reasoning. *Information Sciences*. 2011. Volume 181. Issue 24. P. 5349-5363. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0020025511004129>. (Date of access: 18.02.2021).
165. Romdhane L.B., Ayeb B. An evolutionary algorithm for abductive reasoning. *Journal of Experimental & Theoretical Artificial Intelligence*. 2011. Vol. 23. Issue 4. P.529-544. URL: <https://www.tandfonline.com/doi/abs/10.1080/0952813X.2010.505824>. (Date of access: 18.02.2021).
166. Garcez A.S.d., Gabbay D.M., Ray O. et al. Abductive reasoning in neural-symbolic systems. *Topoi*. 2007. Vol. 26. P. 37-49. URL: <https://link.springer.com/article/10.1007%252Fs11245-006-9005-5>. (Date of access: 18.02.2021).
167. Juba B. Learning Abductive Reasoning Using Random Examples. *Thirtieth AAAI Conference on Artificial Intelligence*. 2016. Vol. 30. No. 1. P. 999-1007. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/10099>. (Date of access: 18.02.2021).
168. Zhou Z.-H. Abductive learning: towards bridging machine learning and logical reasoning. *Sci China Inf Sci*. 2019. Vol 62(7). URL: <http://scis.scichina.com/en/2019/076101.pdf>. (Date of access: 18.02.2021).
169. Dong A., Lovallo D., Mounarath R. The effect of abductive reasoning on concept selection decisions. *Design Studies*. 2015. Volume 37. P. 37-58. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0142694X14000805>. (Date of access: 18.02.2021).
170. Hwang M.-Y., Hong J.-C., Ye J.-H., Wu Y.-F. et al. Practicing abductive reasoning: The correlations between cognitive factors and learning effects. *Computers & Education*. 2019. Vol. 138. P. 33-45. URL: <https://www.sciencedirect.com/science/article/pii/S0360131519301010>. (Date of access: 18.02.2021).
171. Lombrozo T. Explanation and abductive inference. *Oxford library of psychology. The Oxford handbook of thinking and reasoning*. 2012. P. 260-276. URL: <https://psycnet.apa.org/record/2012-08871-014>. (Date of access: 18.02.2021).

172. Bridewell W., Langley P. A Computational Account of Everyday Abductive Inference. *Proceedings of the Annual Meeting of the Cognitive Science Society*. 2011. Vol 33(33). P. 2289-2294. URL: <https://escholarship.org/content/qt34n5277r/qt34n5277r.pdf>. (Date of access: 18.02.2021).
173. Ma J. Distributed Abductive Reasoning: Theory, Implementation and Application. 2011. URL: <https://spiral.imperial.ac.uk/handle/10044/1/9163>. (Date of access: 18.02.2021).
174. Пелин Н., Пелин С. Вопросы теории и практики логического программирования. *Acta et commentationes (Științe Exacte și ale Naturii)*. 2016. № 2(2). С. 101.113. URL: https://scholar.googleusercontent.com/scholar?q=cache:ewamYWuenYQJ:scholar.google.com/+формальная+логика+Хорн&hl=uk&as_sdt=0,5&as_ylo=2010&as_yhi=2019&scioq=Абдукция. (Date of access: 18.02.2021).
175. Вострикльва Е.В. Логическая форма и обыденное мышление. *Epistemology & Philosophy of Science*. 2016. №4 (50). С. 244-251. URL: <https://cyberleninka.ru/article/n/logicheskaya-forma-i-obydennoe-myshlenie>. (Date of access: 18.02.2021).
176. Nasser S.H., Yazdani A., Darvishi Salokolaei D. A Primal Simplex Algorithm for Solving Linear Programming Problem with Grey Cost Coefficients. *Journal of New Research in Mathematics*. 2016. Vol. 1. Issue 4. P. 115-135. URL: http://jnrm.srbiau.ac.ir/?_action=articleInfo&article=9081. (Date of access: 18.02.2021).
177. De P.K., Yadav B. A General Approach for Solving Assignment Problems Involving with Fuzzy Cost Coefficients. *Modern Applied Science*. 2012. Vol. 6. No. 3. P. 2-10. URL: <https://pdfs.semanticscholar.org/5aec/b0d81c83f8fc6743cb718b1525a2fa286183.pdf>. (Date of access: 18.02.2021).
178. Morano P., Tajani F., Salvo F., De Ruggiero M. Weight Coefficients in the Appraisal System Approach. *Computational Science and Its Applications – ICCSA 2019*. 2019.

Vol 11622. P. 3-12. URL: https://link.springer.com/chapter/10.1007/978-3-030-24305-0_1. (Date of access: 18.02.2021).

179. Sakama C. Abduction in argumentation frameworks. *Journal of Applied Non-Classical Logics*. 2018. Vol. 28. Issue 2-3. P. 218-239. URL: <https://www.tandfonline.com/doi/abs/10.1080/11663081.2018.1487241?journalCode=tnc120>. (Date of access: 18.02.2021).
180. Hobbs J.R., Stickel M., Martin P., Edwards D. Interpretation as Abduction. *Artificial Intelligence*. 1993. Vol. 63. Issue 1-2. P. 95-103. URL: <https://www.aclweb.org/anthology/P88-1012.pdf>. (Date of access: 18.02.2021).
181. Bistarelli S., Martinelli F., Santini F. A semiring-based framework for the deduction/abduction reasoning in access control with weighted credentials. *Computers & Mathematics with Applications*. 2012. Vol. 64. Issue 4. P. 447-462. URL: <https://www.sciencedirect.com/science/article/pii/S0898122111010728>. (Date of access: 18.02.2021).
182. Dai W.-Z., Xu Q., Yu Y., Zhou Z.-H. Bridging Machine Learning and Logical Reasoning by Abductive Learning. 33rd Conference on Neural Information Processing Systems (Vancouver, Canada, NeurIPS 2019). 2019. URL: <https://openreview.net/forum?id=ryxA8NHeLB>. (Date of access: 18.02.2021).
183. Kakas A.C., Riguzzi F. Abductive concept learning. *New Gener Comput*. 2000. Vol. 18. P. 243-294. URL: <https://link.springer.com/article/10.1007%2FBF03037531>. (Date of access: 18.02.2021).
184. El-Alfy E.M., Abdel-Aal R.E. Construction and analysis of educational tests using abductive machine learning. *Computers & Education*. 2008. Volume 51. Issue 1. P. 1-16. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0360131507000218>. (Date of access: 18.02.2021).
185. Mutter S. Classification using Association Rules. 2004. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.8785&rep=rep1&type=pdf>. (Date of access: 18.02.2021).
186. Liu B., Ma Y., Wong C.K. et al. Scoring the Data Using Association Rules. *Applied Intelligence*. 2003. Vol. 18. P. 119-135. URL:

- <https://link.springer.com/article/10.1023/A:1021931008240>. (Date of access: 18.02.2021).
187. Dass R. Classification Using Association Rules. Research and Publications. 2008. P. 1-52. URL: <http://vslir.iima.ac.in:8080/jspui/handle/11718/156>. (Date of access: 18.02.2021).
188. McGregor A., Hall M., Lorier P., Brunskill J. Flow Clustering Using Machine Learning Techniques. *Passive and Active Network Measurement*. 2004. Vol 3015. P. 205-214. URL: https://link.springer.com/chapter/10.1007/978-3-540-24668-8_21. (Date of access: 18.02.2021).
189. Domingos P., Hulten G. A General Method for Scaling Up Machine Learning Algorithms and its Application to Clustering. *Eighteenth International Conference on Machine Learning*. 2001. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.2571>. (Date of access: 18.02.2021).
190. Wagstaff K., Cardie C. Clustering with Instance-level Constraints. *Seventeenth International Conference on Machine Learning*, 2000. P. 1103-1110. URL: <https://www.aaai.org/Papers/AAAI/2000/AAAI00-180.pdf>. (Date of access: 18.02.2021).
191. Al-Omary A.Y., Jamil M.S. A new approach of clustering based machine-learning algorithm. *Knowledge-Based Systems*. 2006. Volume 19. Issue 4. P. 248-258. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0950705106000189>. (Date of access: 18.02.2021).
192. Park, Y.W., Klabjan, D. An aggregate and iterative disaggregate algorithm with proven optimality in machine learning. *Mach Learn*. 2016. Vol. 105. P. 199-232. <https://link.springer.com/article/10.1007/s10994-016-5562-z>. (Date of access: 18.02.2021).
193. Glassman E.L. Clustering and Visualizing Solution Variation in Massive Programming Classes. 2016. URL: <http://up.csail.mit.edu/other-pubs/elg-thesis.pdf>. (Date of access: 18.02.2021).
194. Мельник В. Д. Абдуктивне виведення знань про процес буріння на основі мультимедійних даних про бурове обладнання. *Науковий вісник Івано-Франківського національного технічного університету нафти і газу*. 2016. №1(40). С.80-91.

195. Шекета В. І., Мельник В. Д., Гобир Л. М. Інтелімедійна інформаційна система підтримки прийняття рішень в процесі буріння. *Проблеми інформаційних технологій*. 2016. № 1. С. 96-116.
196. Мельник В. Д., Шекета В. І. Інтелімедійні засоби підтримки прийняття рішень в нафтогазовій справі. *Молодий вчений: науковий журнал*. № 5 (69) травень 2019. С. 268-271.
197. Мельник В.Д., Якубів Т.Р. Побудова інтелектуальних навчальних систем на основі мультиагентних технологій. *Інформаційні технології в освіті, науці, техніці та промисловості* : зб. тез доповідей Всеукр.наук.-практ. конференції аспірантів, молодих учених та студентів (м. Івано-Франківськ, 8-11 жовтня 2013 р.). Івано-Франківськ, 2013. С. 98-100.
198. Мельник В. Д. Проблемы интеграции мультимедийных данных в интеллектуальные информационные технологии. *Наука, техника и технологии в контексте глобализации: парадигматические характеристики и проблемы интеграции = Science, engineering and technology in the context of globalization: paradigmatic characteristics and problems of integration* : мат. межд. науч. конф. (28-29 окт., 2015 г.). Прага : Sociosféra-CZ, 2015. С. 42-44.
199. Мельник В. Д., Шекета В. І., Романишин Ю. Л., Гургула О. Б. Застосування інтелімедійних інформаційних технологій в навчальному процесі. *Інформаційні технології та комп'ютерне моделювання*: матеріали Міжнародної науково-практичної конференції (м. Івано-Франківськ, 23-28 травн. 2016 р.) / Прикарпатський національний університет імені Василя Стефаника. Івано-Франківськ, 2016. С. 62-64.
200. Мельник В. Д., Демчина М. М., Шекета В. І. Технології інтерпретації документів в інтелектуальних системах дистанційного навчання. *Вісник Кременчуцького нац. ун-ту ім. М. Остроградського*. 2011. № 3 (68). С.18-22.
201. Мельник В. Д., Шекета В. І., Демчина М. М. Виведення значень логічних параметрів в інтелектуальних системах дистанційного навчання. *Вісник*

Хмельницького нац. ун-ту. Технічні науки : наук. журнал. Хмельницьк, 2011. № 1. С. 118-122.

202. Мельник В. Д., Вовк Р. Б., Шекета В. І. Система інтелектуальної підтримки прийняття рішень при контролі технологічних параметрів. *Методи та прилади контролю якості = Methods and devices of quality control: науково-технічний журнал* / За ред. І. С. Кісіля; ІФНТУНГ. 2012. № 2 (29). С. 129.
203. Вовк Р. Б., Мельник В. Д., Гобир Л. М. Подання та оброблення технологічних знань про процес буріння на основі обмежень. *Науковий вісник ІФНТУНГ*. 2013. №1(34). С. 73-81.
204. Шекета В. І., Демчина М. М., Мельник В. Д. Імплементация інтелектуальної стратегії прийняття рішень у процесі буріння. *Нафтогазова енергетика: Всеукраїнський науково-технічний журнал*. 2013. №2(20). С. 38-50.
205. Мельник В. Д. Формальна структура кейсів технологічних проблем в процесі побудови їх рішень на основі обмежень. *Математичні машини та системи: науковий журнал Інституту проблем математичних машин та систем НАН України*. №2. 2016. С. 116-127.
206. Мельник В. Д. Структуризація тьюторних складових навчальних систем. *Міжнародний науковий журнал «Інтернаука»*. 2020. № 3 (83), 1 т. С. 47-50.
207. Мельник В. Д. Використання продукційного підходу для інтелектуалізації діалогів з користувачем в системах дистанційного навчання. *Радиоэлектроника и молодежь в XXI веке* : матеріали XIII Міжнародного молодіжного форуму (г., Харків, 30 марта - 1 апреля 2009). Х. : ХНУРЭ, 2009. ч.2. С.168.
208. Мельник В. Д., Шекета В. І., Демчина М. М. Обробка суб'єктивних оцінок групи експертів в інтелектуальній системі дистанційного навчання. *Математичне і програмне забезпечення інтелектуальних систем* : матеріали VII Міжнар. наук.-практ. конф. (м. Дніпропетровськ, 18-20 листопада 2009 р.). Дніпропетровськ, ДНУ, 2009. С.184-185.
209. Мельник В. Д. Структуризація відповідей користувача в системі дистанційного навчання на основі експертних параметрів. *Системний аналіз та інформаційні*

технології : матеріали XI Міжнародної науково-технічної конференції (м. Київ, 26-30 травня 2009 р.). К.: НТУУ “КПІ”, 2009. С.348.

210. Мельник В. Д., Процюк В. Р., Процюк Г. Я. Представлення документів та їх класів в системах дистанційного навчання. *Сучасні проблеми прикладної математики та інформатики* : матеріали XVI Всеукраїнської наукової конференції (м. Львів, 8-9 жовтня 2009 р.). Львів, 2009. С.143.
211. Мельник В. Д., Шекета В. І., Демчина М. М. Прийняття рішень при модифікації запитів в системі дистанційного навчання. *Теоретичні та прикладні аспекти побудови програмних систем* : матеріали шостої Міжнародної конференції (м. Київ, 8-10 грудня 2009 р.). К., 2009. С. 149-156.
212. Мельник В. Д. Побудова контенту навчального матеріалу орієнтованого на знання. *Математичне і програмне забезпечення інтелектуальних систем* : матеріали VIII Міжнародної науково-практичної конференції (м. Дніпропетровськ, 10-12 листопада 2010 р.). Дніпропетровськ, 2010. С. 148.
213. Демчина М.М., Мельник В. Д., Гобир Л. М. Застосування функцій доцільності при прийнятті рішень щодо оптимізації технологічних параметрів. *Математичне та імітаційне моделювання систем МОДС 2013* : восьма міжн. наук.-практ. конференція: тези доповідей (Чернігів-Жукин, 24-28 черв. 2013 р.). Чернігів, 2013. С. 145-149.
214. Демчина М. М., Мельник В. Д., Бестильний М. Я. Інтелектуальний контроль параметрів буріння. *Методи та засоби неруйнівного контролю промислового обладнання* : збір. тез. доп. 4-ої наук.- практ. конф. (м. Івано-Франківськ, 26-27 лист. 2013 р.). Івано-Франківськ: ІФНТУНГ, 2013. С.41-43.
215. Демчина М. М., Юрчишин В. М., Мельник В. Д. Дослідження функціонування бази знань системи адаптивного тестування. *Сучасні інформаційні технології в дистанційній освіті* : зб. тез доповідей III Всеукраїнської науково-практичного семінару (м. Івано-Франківськ, 22-24 вер. 2014 р.). Івано-Франківськ.: вид-тво ІФНТУНГ, 2014. С.97-100.

216. Юрчишин В. М., Чесановський М. С., Стисло Т.Р., Мельник В. Д. Побудова знання-орієнтованого управління технологічними процесами нафтогазової галузі. *Інформаційні технології в освіті, техніці та промисловості* : матеріали III всеукраїнської науково-практичної конференції молодих учених і студентів (м. Івано-Франківськ, 10-13 жовтня 2017р.). Івано-Франківськ: вид-тво ІФНТУНГ, 2017. С. 15-17.
217. Юрчишин В.М., Стисло Т. Р., Стисло О. В., Гобир Л. М. Інтелімедійні методи контролю якості бурового обладнання. *Сучасні прилади, матеріали і технології для неруйнівного контролю і технічної діагностики машинобудівного і нафтогазопромислового обладнання* : зб. матеріалів доп. 8-ої наук.-техн. конф. Івано-Франківськ : ІФНТУНГ, 2017. С. 175-177.
218. Vitaliy Melnyk, Roman Vovk, Mykola Demchyna. Frame Based Approach to Construction of Intelligent System for Student Knowledge Control. 2010 IEEE Xth International Conference Modern Problems Of Radio Engineering, Telecommunications And Computer Science: Proceedings, Lviv-Slavske, February 23-27, 2010. P.287.
219. Мельник В. Д. Особливості представлення навчального контенту в інтелектуальній тьюторній системі на основі обмежень. *Математичне та програмне забезпечення інтелектуальних систем* : тези доповідей XI міжнародної наук.-практ. конф. (м. Дніпропетровськ, 20-22 лист. 2013 р.). Дніпропетровськ, 2013. С. 157-159.

ДОДАТОК А

Класифіковані лістинги проектованої моделі інтелімедійної інформаційної системи підтримки прийняття рішень в нафтогазовій справі

/КЛАС ifntung.im.Main.as

```

package ifntung.im {
    import com.greensock.plugins.GlowFilterPlugin;
    import com.greensock.plugins.TweenPlugin;
    import com.greensock.TweenMax;
    import flash.display.Bitmap;
    import flash.display.MovieClip;
    import flash.display.Sprite;
    import flash.events.Event;
    import flash.events.MouseEvent;
    import flash.geom.Point;
    import flash.geom.Rectangle;
    import flash.ui.ContextMenu;
    import flash.utils.getDefinitionByName;
    import flash.utils.getQualifiedClassName;
    import ifntung.im.problem.AbstractProblem;
    import ifntung.im.problem.FlagsProblem;
    import ifntung.im.problem.PuzzleProblem;
    import ifntung.im.problem.RigProblem;
    import ifntung.im.problem.TestsProblem;
    import ifntung.im.util.MathUtils;

    /**
     * Базовий клас проекту.
     * Відповідає за логіку відображення всіх типів
     * задач, вирішуваних ППІ та їх менеджмент.
     *
     * @author Roman
     */
    public class Main extends Sprite {
        ///Розміри екрану
        public static const WIDTH:Number = 800;
        public static const HEIGHT:Number = 600;
        ///Поточна задача користувача
        private var currentProblem:AbstractProblem;
        private var problemsViewClassName:Array;
        ///Синглтонне посилання на об'єкт даного класу
        public static var instance:Main;
        /**
         * Статичне підключення класів із зовнішньої бібліотеки.
         */
        private static function linking():void {
            _PuzzleProblem1, _PuzzleProblem2, _FlagsProblem1, _TestsProblem1, _RigProblem;
            TweenPlugin.activate([GlowFilterPlugin]); //activation is permanent in the SWF, so this line only needs to be run once.
        }

        public function Main() {
            instance = this;
            if (stage) init();
            else addEventListener(Event.ADDED_TO_STAGE, init);
            linking();
        }

        /**
         * Точка входу
         */
        private function init(e:Event = null):void {
            removeEventListener(Event.ADDED_TO_STAGE, init);
            scrollRect = new Rectangle(0, 0, 800, 620);

            var my_menu:ContextMenu = new ContextMenu();
            my_menu.builtInItems.forwardAndBack = false;
            my_menu.builtInItems.loop = false;
            my_menu.builtInItems.play = false;
            my_menu.builtInItems.print = false;
            my_menu.builtInItems.quality = false;
            my_menu.builtInItems.rewind = false;
            my_menu.builtInItems.save = false;
    
```

```

my_menu.builtInItems.zoom = true;
    contextMenu = my_menu;

    var intro:MovieClip = new _Intro();
    intro.y = HEIGHT / 2;
    intro.x = WIDTH + WIDTH / 2;
    addChild(intro);

TweenMax.to(intro, 0.75, { delay:0.5, x: WIDTH / 2, onComplete: function():void {
    TweenMax.to(intro, 0.75, {delay:2, x: -WIDTH, onComplete: function():void {
        removeChild(intro);

        showProblemTypeSelectionWindow();

    }});
    });

}
//Карта відображення типу задч на множину назв його екземплярів
private var problemTypeToViewClassNamesMap:Object = {
    "_rigs_":["_RigProblem"],
    "_test_":["_TestsProblem1"],
    "_flags_":["_FlagsProblem1"],
    "_puzzle_":["_PuzzleProblem1", "_PuzzleProblem2"]
};
//Обробник події вибору типу задачі
private function onTsmClick(e:MouseEvent):void {
    problemsViewClassName = problemTypeToViewClassNamesMap[e.target.name];
    tsm.mouseEnabled = false;
    TweenMax.to(tsm, 0.25, {x:-Main.WIDTH / 2, onComplete:function(){
        tsm.mouseEnabled = true;
        removeChild(tsm);
        showNextProblem();
    }});
}
//Вікно вибору типу задачі
private var tsm: _TypeSelectionWindow = null;
private function showProblemTypeSelectionWindow():void {
    if (tsm == null) {
        tsm = new _TypeSelectionWindow();
        tsm["_rigs_"].addEventListener(MouseEvent.CLICK, onTsmClick);
        tsm["_test_"].addEventListener(MouseEvent.CLICK, onTsmClick);
        tsm["_flags_"].addEventListener(MouseEvent.CLICK, onTsmClick);
        tsm["_puzzle_"].addEventListener(MouseEvent.CLICK, onTsmClick);
    }

    tsm.y = Main.HEIGHT / 2;
    tsm.x = Main.WIDTH / 2 + Main.WIDTH;
    tsm.mouseEnabled = false;
    TweenMax.to(tsm, 0.25, {x:Main.WIDTH / 2, onComplete:function(){
        tsm.mouseEnabled = true;
    }});
    addChild(tsm);
}
/**
 * Створення задачі за її іменем в бібліотеці
 * @paramviewClassName імя в бібліотеці
 * @return обект створеної задачі
 */
public function createProblem(viewClassName:String):AbstractProblem {
    var _type:String = viewClassName.substring(0, viewClassName.indexOf("Problem"));
    var _class:Class = getDefinitionByName(viewClassName) as Class;
    var view:MovieClip = new _class() as MovieClip;
    switch (_type) {
        case "_Puzzle":
            return new PuzzleProblem(view);
        case "_Flags":
            return new FlagsProblem(view);
        case "_Tests":
            return new TestsProblem(view);
        case "_Rig":
            return new RigProblem(view);
    }
}

```



```

}

/**
 * @parammin мінімальне значення (лівий край проміжка)
 * @parammax максимальне значення (правий край проміжка)
 * @paramnumbers.Amount кількість випадкових чисел
 * @return повертає множину випадкових чисел в конкретному проміжку
 */
public static function getUniqueRandomNumbersBetween(min:int, max:int, numbers.Amount:int):Array {
    if (min > max) return null;

    if (numbers.Amount > max - min + 1) return null;

    var arr:Array = new Array(max - min + 1);
    for (var i:int = 0; i < arr.length; i++) {
        arr[i] = i + min;
    }

    var retArr:Array = new Array(numbers.Amount);
    for (var j:int = 0; j < numbers.Amount; j++) {
        var rndArrInd:int = Math.floor(Math.random() * arr.length);
        retArr[j] = arr[rndArrInd];
        arr.splice(rndArrInd, 1);
    }

    return retArr;
}

/**
 * @paramvertices вершини опуклого багатокутника
 * @return повертає площу опуклого багатокутника за заданими вершинами
 */
public static function calcPolygonArea(vertices:Array):Number {
    var total = 0;

    for (var i = 0, l = vertices.length; i < l; i++) {
        var addX = vertices[i].x;
        var addY = vertices[i == vertices.length - 1 ? 0 : i + 1].y;
        var subX = vertices[i == vertices.length - 1 ? 0 : i + 1].x;
        var subY = vertices[i].y;

        total += (addX * addY * 0.5);
        total -= (subX * subY * 0.5);
    }

    return Math.abs(total);
}

/**
 * Реалізація Алгоритму Джарвіса
 * @parampoints точки
 * @return масив точок, які утворюють опуклий багатокутник максимального розміру
 */
public static function getConvexPolygonVertices(points:Array):Array {
    function cross(o:Point, a:Point, b:Point) {
        return (a.x - o.x) * (b.y - o.y) - (a.y - o.y) * (b.x - o.x)
    }

    /**
     * @param points An array of [X, Y] coordinates
     */
    function convexHull(points:Array):Array {
        points.sort(function(a:Point, b:Point) {
            return a.x == b.x ? a.y - b.y : a.x - b.x;
        });

        var lower:Array = [];
        for (var i:int = 0; i < points.length; i++) {
            while (lower.length >= 2 && cross(lower[lower.length - 2], lower[lower.length - 1], points[i]) <= 0) {
                lower.pop();
            }
            lower.push(points[i]);
        }

        var upper:Array = [];
        for (var i:int = points.length - 1; i >= 0; i--) {
            while (upper.length >= 2 && cross(upper[upper.length - 2], upper[upper.length - 1], points[i]) <= 0) {
                upper.pop();
            }
        }
    }
}

```

```

upper.push(points[i]);
    }

    upper.pop();
    lower.pop();
    return lower.concat(upper);
}

return convexHull(points);
}

/**
 * @return Повертає площу трикутника за трьома вершинами
 */
public static function calculateTriangleArea(pointA:Point, pointB:Point, pointC:Point):Number {
    var a:Number = distance(pointB.x, pointB.y, pointC.x, pointC.y);
    var b:Number = distance(pointA.x, pointA.y, pointC.x, pointC.y);
    var c:Number = distance(pointA.x, pointA.y, pointB.x, pointB.y);
    var p:Number = (a + b + c) / 2;
    var s:Number = Math.sqrt(p * (p - a) * (p - b) * (p - c));
    return s;
}

/**
 * Визначення чи знаходиться точка всередині опуклого багатокутника
 */
public static function isPointInsidePolygon(polygonVertices:Array, point:Point, polygonArea:Number = NaN):Boolean {
    var s:Number = 0;
    for (var i:int = 0; i < polygonVertices.length - 1; i++)
    {
        s += calculateTriangleArea(polygonVertices[i], polygonVertices[i + 1], point);
    }
    s += calculateTriangleArea(polygonVertices[polygonVertices.length - 1], polygonVertices[0], point);

    var pS:Number = isNaN(polygonArea) ? calcPolygonArea(polygonVertices) : polygonArea;
    trace(s - pS);
    return (pS >= s) || (s - pS < 0.01);
}

/**
 *
 * @param array1 перший масив
 * @param array2 другий масив
 * @return повертає масив, який є об'єднанням масивів без повторень однакових елементів
 */
public static function concatArraysNoDuplicate(array1:Array, array2:Array):Array {
    var array:Array = array1.concat(array2);

    for(var i = 0; i < array.length; ++i) {
        for(var j = i + 1; j < array.length; ++j) {
            if(array[i] === array[j])
                array.splice(j--, 1);
        }
    }

    return array;
}

/**
 *
 * @param arr1 перший масив
 * @param arr2 перший масив
 * @return повертає масив, який містить спільні елементи двох масивів
 */
public static function getCommonElements(arr1:Array, arr2:Array):Array {
    var commonEl:Array = new Array();
    for (var i:int = 0; i < arr1.length; i++)
    {
        if (arr2.indexOf(arr1[i]) >= 0) {
            commonEl.push(arr1[i]);
        }
    }
    return commonEl;
}
}
}
}

```

//КЛАС ifntung.im.util.TextPrintAnimator.as

```

package ifntung.im.util
{
    import flash.events.TimerEvent;
    import flash.text.StaticText;
    import flash.text.TextField;
    import flash.utils.Timer;

    /**
* Клас, який дозволяє анімувати текстове поле
* процесом набору його текстових символів
* @author Roman
*/
    public class TextPrintAnimator
    {

        private static var INSTANCES:Array = new Array();

        private var textField:TextField;
        private var charDelay:Number = 0.07;
        private var onCompleteCallback:Function;

        private var str:String;
        private var currCharInd:int;

        private var self:TextPrintAnimator;
        private var currTimer:Timer = null;
        private var broken:Boolean = false;
        private var _ndl:int = 2;

        function TextPrintAnimator(textField:TextField, charDelay:Number = 0.07, onCompleteCallback:Function = null, _ndl:int = 16) {
            this.textField = textField;
            this.charDelay = charDelay;
            this.onCompleteCallback = onCompleteCallback;
            this.self = this;
            this._ndl = _ndl;

            str = textField.text;
            currCharInd = 0;

            var sndWasStopped:Boolean = false;
            for (var i:int = INSTANCES.length - 1; i > -1; i--) {
                var inst:TextPrintAnimator = INSTANCES[i] as TextPrintAnimator;
                if (inst.textField == textField) {
                    if (inst.currTimer) inst.currTimer.stop();
                    inst.broken = true;
                    sndWasStopped = true;
                    INSTANCES.splice(i, 1);
                }
            }

            INSTANCES.push(this);

            function f():void {
                var timer:Timer = new Timer(self.charDelay, 1);
                timer.addEventListener(TimerEvent.TIMER_COMPLETE, function(e:TimerEvent):void {
                    if (self.currCharInd >= self.str.length) {
                        if (self.onCompleteCallback) self.onCompleteCallback();

                        var ind:int = INSTANCES.indexOf(self);
                        if (ind > -1) INSTANCES.splice(ind, 1);

                        return;
                    }
                    if (self.broken) return;

                    var dl:int = currCharInd + _ndl <= str.length ? _ndl : str.length - currCharInd;
                    textField.text += str.substr(currCharInd, dl);
                    currCharInd += dl;
                    f();
                });
                timer.start();
            }
        }
    }
}

```

```

f());

        textField.text = "";

        if (!sndWasStopped) {
            var j:int = 0
            for (; j < INSTANCES.length; j++) {
                var inst:TextPrintAnimator = INSTANCES[j] as TextPrintAnimator;
                if (inst != this && inst.lostDrukTime() > this.lostDrukTime()) break;
            }
        }

        private function lostDrukTime():int {
            return (str.length - currCharInd) * charDelay;
        }

24):void {
        new TextPrintAnimator(textField, charDelay, onCompleteCallback, _ndl);

        private function kill():void {
            var ind:int = INSTANCES.indexOf(self);
            if (ind > -1) {
                if (currTimer) currTimer.stop();
                breaked = true;
                INSTANCES.splice(ind, 1);
            }
        }

        public static function killTextAnimationOf(textField:TextField):void {
            for each (var inst:TextPrintAnimator in INSTANCES)
            {
                if (inst.textField == textField) inst.kill();
            }
        }
    }
}

```

//КЛАС ifnung.im.problem.**AbstractProblem.as**

```

package ifnung.im.problem {
    import flash.display.MovieClip;
    import flash.utils.getQualifiedClassName;

    /**
     * Абстрактна задача.
     * Інкапсулює основну логіку та дані задачі.
     * @author Roman
     */
    public class AbstractProblem {
        //Графічний контент із бібліотеки
        public var view:MovieClip;
        //Назва графічного контенту із бібліотеки
        public var viewClassName:String;
        /**
         * Конструктор із ініціалізацією
         */
        public function AbstractProblem(view:MovieClip) {
            this.view = view;
            this.viewClassName = getQualifiedClassName(view);
        }
        /**
         * Створення задачі
         */
        public function create():void {

        }
        /**
         * Знищення та очищення ресурсів
         */
        public function destroy():void {

        }
    }
}

```

```

/**
 * Перевірка чи вирішена задача чи ні
 * @return стан вирішення задачі
 */
public function isSolved():Boolean {
    return false;
}
}

}
//ІНТЕРФЕЙС ifitung.im.problem.IRandomizeable.as
package ifitung.im.problem
{
    /**
     * Інтерфейс, який надає можливість внести бажану випадковість у
     * логіку класу реалізатора
     * @author Roman
     */
    public interface IRandomizeable
    {
        function randomize():void;
    }
}
//КЛАС ifitung.im.problem.FlagsProblem.as
package ifitung.im.problem {
    import com.greensock.TweenMax;
    import flash.display.DisplayObject;
    import flash.display.MovieClip;
    import flash.display.Sprite;
    import flash.events.MouseEvent;
    import flash.geom.Point;
    import flash.geom.Rectangle;
    import ifitung.im.Main;
    import ifitung.im.util.MathUtils;
    import ifitung.im.util.TextPrint.Animator;

    /**
     * Задача, яка реалізує логіку перевірки знань на базі виділення
     * конкретної затребуваної області опуклим багатокутником.
     * Імплементує випадкову генерацію контенту.
     * @author Roman
     */
    public class FlagsProblem extends AbstractProblem implements IRandomizeable {
        //Емпірично визначене оптимальне відношення площ виділеного та базового багатокутників
        const MAX_AREA_DIV:Number = 3.5;
        //Масив точок-вершин опуклого багатокутника
        var flags:Array;
        //Рендер-об'єкт розставлених вершин
        var drawing:Sprite;
        //Площа виділеного опуклого багатокутника
        var objectsArea:Object;
        var objectsVertices:Object;
        //Кількість підзадач
        var tasksNumber:int;
        var tasksIdToSolving:Array;
        //ІД поточної підзадачі
        var currentTaskId:int;
        /**
         * Конструктор ініціалізації
         */
        public function FlagsProblem(view:MovieClip) {
            super(view);
        }
        /**
         * Створення задачі
         */
        override public function create():void {
            tasksNumber = view["theProblem"].numChildren / 2;
            randomize();

            objectsArea = new Object();

```



```

objectsVertices = new Object();

for (var k:int = 0; k < tasksIdToSolving.length; k++) {
    var _mc:MovieClip = view["theProblem"]["_object_" + tasksIdToSolving[k]] as MovieClip;

    objectsVertices[_mc.name] = new Array();
    var _points:Array = new Array();
    for (var j:int = 0; j < _mc.numChildren; j++) {
        var __ch:DisplayObject = _mc.getChildAt(j);
        if (__ch is _Flag) {
            var _p:Point = new Point(__ch.x, __ch.y);
            __p = __ch.parent.localToGlobal(_p);
            _points.push(_p);
        }
    }
    _points = MathUtils.getConvexPolygonVertices(_points);
    objectsVertices[_mc.name] = _points;
    objectsArea[_mc.name] = MathUtils.calcPolygonArea(_points);
}

for (var k:int = 1; k <= tasksNumber; k++) {
    var _mc:MovieClip = view["theProblem"]["_object_" + k] as MovieClip;
    for (var j:int = 0; j < _mc.numChildren; j++) {
        var __ch:DisplayObject = _mc.getChildAt(j);
        if (__ch is _Flag) {
            __ch.visible = false;
        }
    }
}

flags = new Array();
view["theProblem"].mouseChildren = false;
view["theProblem"].addEventListener(MouseEvent.CLICK, onTheProblemClick);

view["_clearBtn"].addEventListener(MouseEvent.CLICK, onClearBtnClick);

showNextTaskIfExist();
}
/**
 * Випадкова зміна стану задачі
 */
public function randomize():void {
    tasksIdToSolving = MathUtils.getUniqueRandomNumbersBetween(1, tasksNumber,
Math.ceil(Math.random()*tasksNumber));
}
/**
 * Знищення та очищення ресурсів
 */
override public function destroy():void {
    view["_clearBtn"].visible = false;
    view["theProblem"].removeEventListener(MouseEvent.CLICK, onTheProblemClick);
    view["_clearBtn"].removeEventListener(MouseEvent.CLICK, onClearBtnClick);

    Main.instance.stage.removeChild(drawing);
}
/**
 * Перевірка чи вирішена задача чи ні
 * @return стан вирішення задачі
 */
override public function isSolved():Boolean {
    if (flags.length < 3) return false;

    var objectView:MovieClip = view["theProblem"]["_object_" + currentTaskId] as MovieClip;

    for (var i:int = flags.length - 1; i >= 0; i--) {
        if (!MathUtils.isPointInsidePolygon(objectsVertices["_object_" + currentTaskId], flags[i],
objectsArea["_object_" + currentTaskId])
        ) {
            trace("! is point inside");
            return false;
        }
    }

    var div:Number = objectsArea["_object_" + currentTaskId] / MathUtils.calcPolygonArea(flags);
    trace(div);
    if (div < MAX_AREA_DIV) {
        return true;
    }
}

```



```

}

/**
 * Показує підзадачу, якщо така існує
 * @return true якщо існує підзадача, в іншому випадку else
 */
private function showNextTaskIfExist():Boolean {
    if (tasksIdToSolving.length == 0) return false;

    clear();

    var taskId:int = tasksIdToSolving.pop();

showTask(taskId);

        return true;
    }
    /**
     * Перерисувати опуклий багатокутник
     */
    private function redraw():void {
        if (flags.length < 1) return;

        drawing.graphics.clear();
        drawing.graphics.lineStyle(1, 0xFF0000);
        drawing.graphics.beginFill(0xFF0000, 0.25);
        drawing.graphics.moveTo(flags[0].x, flags[0].y);
        for (var i:int = 0; i < flags.length; i++) {
            drawing.graphics.lineTo(flags[i].x, flags[i].y);
        }
        drawing.graphics.endFill();
    }
}

}

//КЛАС ifntung.im.problem.PuzzleProblem.as
package ifntung.im.problem {
    import flash.display.MovieClip;
    import flash.events.MouseEvent;
    import flash.utils.getQualifiedClassName;
    import ifntung.im.Main;
    import ifntung.im.util.MathUtils;

    /**
     * Задача, яка реалізує логіку інтерактивного відновлення графічного
     * контенту методом переміщення попередньо диференційованих частин на
     * відповідні позиції
     * @author Roman
     */
    public class PuzzleProblem extends AbstractProblem implements IRandomizeable{
        //Емпірично встановлена максимально-допустима відстань від об'єкта
        //до його правильної кінцевої позиції
        private static const ALIGN_MAX_DISTANCE:Number = 60;
        private static const MAX_DISTANCE_EPS:Number = 0.00001;
        //Кількість об'єктів для відновлення їх правильної позиції
        var tilesNumber:int;
        /**
         * Конструктор ініціалізації
         */
        public function PuzzleProblem(view:MovieClip) {
            super(view);
        }
        /**
         * Створення задачі
         */
        override public function create():void {
            tilesNumber = view["theProblem"].numChildren / 2;
            for (var i:int = 1; i <= tilesNumber; i++) { var cn:String = getQualifiedClassName(view["theProblem"]["_user_" + i]);
                trace(cn);

                view["theProblem"]["_user_" + i].mouseChildren = false;
                view["theProblem"]["_user_" + i].addEventListener(MouseEvent.CLICK, onTileMouseDown);
                view["theProblem"]["_user_" + i].addEventListener(MouseEvent.CLICK, onTileMouseUp);
                view["theProblem"]["_base_" + i].visible = false;
            }

            randomize();
        }
    }
}

```

```

}

/**
 * Випадкова зміна стану задачі
 */
public function randomize():void {
    var rndSolvedTileIndexes:Array = MathUtils.getUniqueRandomNumbersBetween(1, tilesNumber, int(tilesNumber / 2));
    if (rndSolvedTileIndexes) {
        for (var j:int = 0; j < rndSolvedTileIndexes.length; j++) {
            view["theProblem"]["_user_" + rndSolvedTileIndexes[j]].x = view["theProblem"]["_base_" +
rndSolvedTileIndexes[j]].x;

view["theProblem"]["_user_" + rndSolvedTileIndexes[j]].y = view["theProblem"]["_base_" + rndSolvedTileIndexes[j]].y;
            checkAlign(view["theProblem"]["_user_" + rndSolvedTileIndexes[j]]);
        }
    }
}

/**
 * Витягнення індекса із підзадачі
 * @paramtile графічний контент підзадачі
 * @return індекс
 */
private function parseIndex(tile:MovieClip):int {
    return parseInt(tile.name.substring(tile.name.lastIndexOf("_") + 1, tile.name.length));
}

/**
 * Перевірка відстані прилягання диференційованого об'єкта до його бази
 * позиції
 * @paramtile диференційований об'єкт
 */
private function checkAlign(tile:MovieClip):void {
    var parsedIndex:int = parseIndex(tile);
    if (!tile.hitTestObject(view["theProblem"]["_base_" + parsedIndex])
        MathUtils.distanceBetween(tile, view["theProblem"]["_base_" + parsedIndex]) < ALIGN_MAX_DISTANCE) {
        tile.x = view["theProblem"]["_base_" + parsedIndex].x;
        tile.y = view["theProblem"]["_base_" + parsedIndex].y;

        view["theProblem"].setChildIndex(tile, tilesNumber);
        tile.filters = null;
        tile.removeEventListener(MouseEvent.CLICK, onTileMouseDown);
        tile.removeEventListener(MouseEvent.CLICK, onTileMouseUp);

        if (isSolved()) {
            trace("PUZZLE PROBLEMS HAS BEEN SOLVED");

            Main.instance.showNextProblem();
        }
    }
}

private function onTileMouseUp(e:MouseEvent):void {
    var mc:MovieClip = e.target as MovieClip;
    mc.stopDrag();
    checkAlign(mc);
}

private function onTileMouseDown(e:MouseEvent):void {
    var mc:MovieClip = e.target as MovieClip;
    view["theProblem"].setChildIndex(mc, view["theProblem"].numChildren - 1);
    mc.startDrag();
}

/**
 * Знищення та очищення ресурсів
 */
override public function destroy():void {
    if (view.parent) view.parent.removeChild(view);
}

/**
 * Перевірка чи вирішена задача чи ні
 * @return стан вирішення задачі
 */
override public function isSolved():Boolean {
    for (var i:int = 1; i <= tilesNumber; i++) {
        if (MathUtils.distanceBetween(view["theProblem"]["_user_" + i], view["theProblem"]["_base_" + i]) >
MAX_DISTANCE_EPS) {

```

```

return false;
        }
        }
        return true;
    }
}

//КЛАС ifntung.im.problem.TestsProblem.as
package ifntung.im.problem
{

import com.greensock.easing.Back;
import com.greensock.easing.BackOut;
import com.greensock.TweenMax;
import fl.controls.CheckBox;
import flash.display.MovieClip;
import flash.events.Event;
import ifntung.im.Main;
import ifntung.im.util.MathUtils;
import ifntung.im.util.TextPrintAnimator;

/**
 * Задача, яка реалізує логіку тестового завдання на базі графічного
 * контенту, тобто встановлення правильної відповідності між текстовою
 * та графічною інформацією.
 * @authorRoman
 */
public class TestsProblem extends AbstractProblem implements IRandomizeable
{
    private var idElementNameMap:Object;
    private var currentElementNameQuestion:String;
    private var minId:int = 1;
    private var maxId:int;
    //кількість правильно вирішених тестів
    private var correctSolvedNum:int;
    //кількість неправильно вирішених тестів
    private var incorrectSolvedNum:int;
    /**
     * Конструктор ініціалізації
     */
    public function TestsProblem(view:MovieClip)
    {
        super(view);
    }
    /**
     * Створення задачі
     */
    override public function create():void {
        view["_theProblem_"]["id_name_map"].visible = false;
        view["_theProblem_"]["_correct_"].visible = view["_theProblem_"]["_incorrect_"].visible = false;
        correctSolvedNum = incorrectSolvedNum = 0;
        view["_theProblem_"]["_correctAnswNum_"].text = ""+correctSolvedNum;
        view["_theProblem_"]["_incorrectAnswNum_"].text = ""+incorrectSolvedNum;

        idElementNameMap = new Object();
        var txt:String = view["_theProblem_"]["id_name_map"].text;
        var idDelimNameS:Array = txt.split(String.fromCharCode(13));
        var maxId:int = 1;
        for (var i:int = 0; i < idDelimNameS.length; i++)
        {
            var splited:Array = (idDelimNameS[i] as String).split(".");
            idElementNameMap[splited[0]] = splited[1];
            var sp0:int = parseInt(splited[0]);
            if (sp0 > maxId) maxId = sp0;
        }

        this.maxId = maxId;

        for (var i:int = 1; i <= 4; i++)
        {
            var rb:CheckBox = view["_theProblem_"]["_label" + i + "_"] as CheckBox;
            rb.addEventListener(Event.CHANGE, function(e:Event):void {
                if (isSolved()) {

```

```

view.mouseEnabled = view.mouseChildren = false;

60;

view["_theProblem_"]["_correct_"].visible = true;
view["_theProblem_"]["_correct_"].scaleX = view["_theProblem_"]["_correct_"].scaleY =

TweenMax.to(view["_theProblem_"]["_correct_"], 0.25, { scaleX:1, scaleY:1 });
view["_theProblem_"]["_correctAnswNum_"].text = ""+ ++correctSolvedNum;
TweenMax.delayedCall(2, function():void {
    view.mouseEnabled = view.mouseChildren = true;
    Main.instance.showNextProblem();
});
} else if(anyIncorrectAnswerSelected()) {
    view.mouseEnabled = view.mouseChildren = false;
    view["_theProblem_"]["_incorrect_"].visible = true;

view["_theProblem_"]["_incorrect_"].scaleX = view["_theProblem_"]["_incorrect_"].scaleY = 60;
TweenMax.to(view["_theProblem_"]["_incorrect_"], 0.25, { scaleX:1, scaleY:1 });
view["_theProblem_"]["_incorrectAnswNum_"].text = ""+ ++incorrectSolvedNum;

TweenMax.delayedCall(2, function():void {
    view.mouseEnabled = view.mouseChildren = true;
    randomize();
});

}

});

}

randomize();
}

/**
 * Випадкова зміна стану задачі
 */
public function randomize():void {
    var rndArr:Array = MathUtils.getUniqueRandomNumbersBetween(minId, maxId, 4);
    currentElementNameQuestion = idElementNameMap["" + rndArr[0]];
    view["_questionTxt_"].text = currentElementNameQuestion + " це:";
    TextPrintAnimator.animateText(view["_questionTxt_"], 0.15, null, 1);

    for (var i:int = 1; i <= 4; i++)
    {
        (view["_theProblem_"]["_label" + i + "_"] as CheckBox).selected = false;
        (view["_theProblem_"]["_label" + i + "_"] as CheckBox).label = rndArr[i - 1];
    }

    view["_theProblem_"]["_correct_"].visible = view["_theProblem_"]["_incorrect_"].visible = false;
}
/**
 * Знищення та очищення ресурсів
 */
override public function destroy():void {
}
/**
 * Перевірка на наявність вибору хоча б одної неправильної відповіді
 * @return true якщо хоча б одна неправильна відповідь обрана, інакше false
 */
private function anyIncorrectAnswerSelected():Boolean {
    for (var i:int = 1; i <= 4; i++)
    {
        var cb:CheckBox = view["_theProblem_"]["_label" + i + "_"] as CheckBox;
        if (cb.selected) {
            if (idElementNameMap[cb.label] != currentElementNameQuestion) return true;
        }
    }

    return false;
}
/**
 * Перевірка чи вирішена задача чи ні
 * @return стан вирішення задачі
 */
override public function isSolved():Boolean {
    for (var i:int = 1; i <= 4; i++)
    {
        var cb:CheckBox = view["_theProblem_"]["_label" + i + "_"] as CheckBox;

```

```

if (idElementNameMap[cb.label] == currentElementNameQuestion){
    if (!cb.selected) return false;
    } else {
        if (cb.selected) return false;
    }
}
return true;
}
}
}

```

//КЛАС iftung.im.problem.RigProblem.as

```

package iftung.im.problem
{
    import com.greensock.easing.Linear;
    import com.greensock.plugins.GlowFilterPlugin;
    import com.greensock.TweenMax;
    import flash.display.DisplayObject;
    import flash.display.MovieClip;
    import flash.events.MouseEvent;
    import flash.filters.DropShadowFilter;
    import flash.filters.GlowFilter;
    import iftung.im.Main;
    import iftung.im.util.MathUtils;
    import iftung.im.util.TextPrintAnimator;

    /**
     * Задача, яка реалізує логіку інтелемедійної підтримки прийняття рішення
     * про вибір типу бурильної установки.
     * @author Roman
     */
    public class RigProblem extends AbstractProblem
    {
        //Дані про класифікацію бурильних установок
        private static var DATA:Object = {
            "_1A": [
                "БУ 3200/200 ЭУК-2М2",
                "БУ 3200/200 ЭУК-2М2У",
                "БУ 3200/200 ЭУК-2МЯ",
                "БУ 3200/200 ЭУК-3МА",
                "БУ 3200/200 ДГУ-1М",
                "БУ 3200/200 ДГУ-1У",
                "БУ 3200/200 ДГУ-1Т",
                "БУ 3200/200 ЭУ-1М",
                "БУ 3200/200 ЭУ-1У",
                "БУ 4200/250 ЭК-БМЧ",
                "БУ 4500/270 ЭК-БМЧ",
                "БУ 5000/320 ЭУК-Я",
                "БУ 5000/320 ДГУ-1Т",
                "БУ 5000/320 ДГУ-1",
                "БУ 5000/320 ЭР-0",
                "БУ 5000/320 ЭР",
                "БУ 5000/450 ЭР-Т",
                "БУ 5000/320 ЭК-БМЧ",
                "БУ 6000/400 ЭК-БМЧ",
                "БУ 6500/400 ЭР",
                "БУ 8000/500 ЭР",
                "БУ UNOC 320 ДЕ",
                "БУ UNOC 500 ДЕ",
                "НБО-1К",
                "НБО-Э",
                "НБО-Д",
                "ЗД86-1",
                "ЗД86-2",
                "БУ 1600/100 ДГУ",
                "БУ 1600/100 ЭУ",
                "БУ 2000/125 ЭП",
                "БУ 2000/125 ДЭП ",
                "БУ 2500/160 ДГУ М1",
                "БУ 2900/175 ДЭП-2",
                "БУ 2900/175 ЭПК",
            ]
        }
    }
}

```

"БУ 2900/175 ЕПКМ1",
 "БУ 2900/200 ЕПК-БМ",
 "БУ 2900/200 ДЭП-БМ",
 "БУ 3900/225 ЕПК-БМ",
 "БУ 3900/225 ДЭП-БМ",
 "БУ 4200/250 ЭЧК-БМ",
 "БУ 4500/270 ЭЧК-БМ"

],
 "_1В":[],
 "_2А":[
 "БУ 3200/200 ДГУ-1М",

"БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1",
 "БУ UNOC 320 ДЕ",

"БУ UNOC 500 ДЕ",
 "НБО-Д",
 "ЗД86-1",
 "ЗД86-2",
 "БУ 1600/100 ДГУ",
 "БУ 2000/125 ДЭП ",
 "БУ 2500/160 ДГУ М1",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/200 ДЭП-БМ",
 "БУ 3900/225 ДЭП-БМ"

],
 "_2В":[
 "БУ 2500/160 ЭСК-БМ",

"БУ 3200/200 ЭУК-2М2",
 "БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 4200/250 ЭК-БМЧ",
 "БУ 4500/270 ЭК-БМЧ",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ЭР-0",
 "БУ 5000/320 ЭР",
 "БУ 5000/450 ЭР-Т",
 "БУ 1600/100 ЭУ",
 "БУ 2000/125 ЭП",
 "БУ 2000/125 ДЭП ",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЕПК",
 "БУ 2900/175 ЕПКМ1",
 "БУ 2900/200 ЕПК-БМ",
 "БУ 2900/200 ДЭП-БМ",
 "БУ 3900/225 ЕПК-БМ",
 "БУ 3900/225 ДЭП-БМ",
 "БУ 4200/250 ЭЧК-БМ",
 "БУ 4500/270 ЭЧК-БМ",
 "БУ 5000/320 ЭК-БМЧ",
 "БУ 6000/400 ЭК-БМЧ",
 "БУ 6500/400 ЭР",
 "БУ 8000/500 ЭР"

],
 "_3А":[
 "БУ 2500/160 ЭСК-БМ",

"БУ 3200/200 ЭУК-2М2",
 "БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 3200/200 ДГУ-1М",
 "БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 4200/250 ЭК-БМЧ",
 "БУ 4500/270 ЭК-БМЧ",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1",
 "БУ 5000/320 ЭР-0",

"БУ 5000/320 ЭР",
 "БУ 5000/450 ЭР-Т",
 "БУ 5000/320 ЭК-БМЧ",
 "БУ 6000/400 ЭК-БМЧ",
 "БУ 6500/400 ЭР",
 "БУ 8000/500 ЭР",
 "БУ UNOC 320 ДЕ",
 "БУ UNOC 500 ДЕ",
 "НБО-1К",
 "НБО-Э",
 "НБО-Д",
 "ЗД86-1",
 "ЗД86-2",
 "БУ 1600/100 ДГУ",
 "БУ 1600/100 ЭУ",
 "БУ 2000/125 ЭП",
 "БУ 2000/125 ДЭП "

"БУ 2500/160 ДГУ М1",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПКМ1",
 "БУ 2900/200 ЭПК-БМ",
 "БУ 2900/200 ДЭП-БМ",
 "БУ 3900/225 ЭПК-БМ",
 "БУ 3900/225 ДЭП-БМ",
 "БУ 4200/250 ЭЧК-БМ",
 "БУ 4500/270 ЭЧК-БМ"

],
 "_3В":[],
 "_4А":[,
 "БУ 3200/200 ДГУ-1М",

"БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 5000/320 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1",
 "БУ 5000/320 ЭР-0",
 "БУ 5000/320 ЭР",
 "БУ 5000/450 ЭР-Т",
 "БУ 6500/400 ЭР",
 "БУ 8000/500 ЭР",
 "БУ UNOC 320 ДЕ",
 "БУ UNOC 500 ДЕ",
 "НБО-1КНБО-Э",
 "НБО-Д",
 "ЗД86-1",
 "ЗД86-2",
 "БУ 1600/100 ДГУ",
 "БУ 1600/100 ЭУ",
 "БУ 2000/125 ЭП",
 "БУ 2000/125 ДЭП ",
 "БУ 2500/160 ДГУ М1",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЭПК",
 "БУ 2900/200 ДЭП-БМ",
 "БУ 3900/225 ДЭП-БМ"

],
 "_4В":[,
 "БУ 2500/160 ЭСК-БМ",

"БУ 3200/200 ЭУК-2М2",
 "БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 4200/250 ЭК-БМЧ",
 "БУ 4500/270 ЭК-БМЧ",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ЭК-БМЧ",
 "БУ 6000/400 ЭК-БМЧ",
 "НБО-1К",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПКМ1",
 "БУ 2900/200 ЭПК-БМ",
 "БУ 3900/225 ЭПК-БМ",
 "БУ 4200/250 ЭЧК-БМ",

"БУ 4500/270 ЭЧК-БМ"

],
" _5А":[
"БУ 3200/200 ЭУК-2МЯ",

"БУ 5000/320 ЭУК-Я"

],
" _5В":[
"БУ 2500/160 ЭСК-БМ",

"БУ 3200/200 ЭУК-2М2",
"БУ 3200/200 ЭУК-2М2У",
"БУ 3200/200 ЭУК-3МА",
"БУ 3200/200 ДГУ-1М",
"БУ 3200/200 ДГУ-1У",
"БУ 3200/200 ЭУ-1М",
"БУ 3200/200 ЭУ-1У",
"БУ 4200/250 ЭК-БМЧ",
"БУ 4500/270 ЭК-БМЧ",
"БУ 5000/320 ДГУ-1",
"БУ 5000/320 ЭР-0",
"БУ 5000/320 ЭР",
"БУ 5000/320 ЭК-БМЧ",
"БУ 6000/400 ЭК-БМЧ",
"БУ 6500/400 ЭР",
"БУ 8000/500 ЭР",
"БУ UNOC 320 ДЕ",
"БУ UNOC 500 ДЕ",
"НБО-1К",
"НБО-Э",
"НБО-Д",
"ЗД86-1",
"ЗД86-2",
"БУ 1600/100 ДГУ",
"БУ 1600/100 ЭУ",
"БУ 2000/125 ЭП",
"БУ 2000/125 ДЭП ",
"БУ 2500/160 ДГУ М1",
"БУ 2900/175 ДЭП-2",
"БУ 2900/175 ЭПК",
"БУ 2900/175 ЭПКМ1",
"БУ 2900/200 ЭПК-БМ",
"БУ 2900/200 ДЭП-БМ",
"БУ 3900/225 ЭПК-БМ",
"БУ 3900/225 ДЭП-БМ",
"БУ 4200/250 ЭЧК-БМ",
"БУ 4500/270 ЭЧК-БМ"

],
" _5С":[
"БУ 3200/200 ДГУ-1Т",

"БУ 5000/320 ДГУ-1Т",
"БУ 5000/450 ЭР-Т"

],
" _6А":[
"БУ 4200/250 ЭК-БМЧ",

"БУ 4500/270 ЭК-БМЧ",
"БУ 5000/320 ЭК-БМЧ",
"БУ 6000/400 ЭК-БМЧ",
"БУ 2900/200 ЭПК-БМ",
"БУ 2900/200 ДЭП-БМ",
"БУ 3900/225 ЭПК-БМ",
"БУ 3900/225 ДЭП-БМ",
"БУ 4200/250 ЭЧК-БМ",
"БУ 4500/270 ЭЧК-БМ"

],
" _6В":[
"БУ 3200/200 ЭУК-2М2",

"БУ 3200/200 ЭУК-2М2У",
"БУ 3200/200 ЭУК-2МЯ",
"БУ 3200/200 ЭУК-3МА",
"БУ 3200/200 ДГУ-1М",
"БУ 1600/100 ДГУ",
"БУ 1600/100 ЭУ",
"БУ 2000/125 ЭП",
"БУ 2000/125 ДЭП ",
"БУ 2500/160 ДГУ М1",
"БУ 3200/200 ДГУ-1У",
"БУ 3200/200 ДГУ-1Т",

"БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1",
 "БУ 5000/320 ЭР-0",
 "БУ 5000/320 ЭР",
 "БУ 5000/450 ЭР-Т",
 "БУ 6500/400 ЭР",
 "БУ 8000/500 ЭР",
 "БУ UNOC 320 ДЕ",
 "БУ UNOC 500 ДЕ",
 "НБО-1К",
 "НБО-Э",
 "НБО-Д",
 "ЗД86-1",
 "ЗД86-2",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПК М1"

],
 "_6C":[
 "БУ 3200/200 ЭУК-2М2",

"БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 3200/200 ДГУ-1М",
 "БУ 1600/100 ДГУ",
 "БУ 1600/100 ЭУ",
 "БУ 2000/125 ЭП",
 "БУ 2000/125 ДЭП",
 "БУ 2500/160 ДГУ М1",
 "БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1",
 "БУ 5000/320 ЭР-0",
 "БУ 5000/320 ЭР",
 "БУ 5000/450 ЭР-Т",
 "БУ 6500/400 ЭР",
 "БУ 8000/500 ЭР",
 "БУ UNOC 320 ДЕ",
 "БУ UNOC 500 ДЕ",
 "НБО-1К",
 "НБО-Э",
 "НБО-Д",
 "ЗД86-1",
 "ЗД86-2",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПК М1"

],
 "_6D":[
 "БУ 3200/200 ЭУК-2М2",

"БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 3200/200 ДГУ-1М",
 "БУ 1600/100 ДГУ",
 "БУ 1600/100 ЭУ",
 "БУ 2000/125 ЭП",
 "БУ 2000/125 ДЭП",
 "БУ 2500/160 ДГУ М1",
 "БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1",
 "БУ 5000/320 ЭР-0",
 "БУ 5000/320 ЭР",
 "БУ 5000/450 ЭР-Т",

Продовження Додатку А

"БУ 6500/400 ЭР",
 "БУ 8000/500 ЭР",
 "БУ UNOC 320 ДЕ",
 "БУ UNOC 500 ДЕ",
 "НБО-1К",
 "НБО-Э",
 "НБО-Д",
 "ЗД86-1",
 "ЗД86-2",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПК М1"

],
 " 7А":[
 "БУ 2500/160 ЭСК-БМ",

"БУ 3200/200 ЭУК-2М2",
 "БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 3200/200 ДГУ-1М",
 "БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 4200/250 ЭК-БМЧ",
 "БУ 4500/270 ЭК-БМЧ",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1",
 "БУ 5000/320 ЭР-0",
 "БУ 5000/320 ЭР",
 "БУ 5000/450 ЭР-Т",
 "БУ 5000/320 ЭК-БМЧ",
 "БУ 6000/400 ЭК-БМЧ",
 "БУ 6500/400 ЭР",
 "БУ 8000/500 ЭР",
 "БУ UNOC 320 ДЕ",
 "БУ UNOC 500 ДЕ",
 "НБО-1К",
 "НБО-Э",
 "НБО-Д",
 "ЗД86-1",
 "ЗД86-2",
 "БУ 1600/100 ДГУ",
 "БУ 1600/100 ЭУ",
 "БУ 2000/125 ЭП",
 "БУ 2000/125 ДЭП",
 "БУ 2500/160 ДГУ М1",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПКМ1",
 "БУ 2900/200 ЭПК-БМ",
 "БУ 2900/200 ДЭП-БМ",
 "БУ 3900/225 ЭПК-БМ",
 "БУ 3900/225 ДЭП-БМ",
 "БУ 4200/250 ЭЧК-БМ",
 "БУ 4500/270 ЭЧК-БМ"

],
 " 7В":[
 "БУ 2500/160 ЭСК-БМ",

"БУ 3200/200 ЭУК-2М2",
 "БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 3200/200 ДГУ-1М",
 "БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 4200/250 ЭК-БМЧ",
 "БУ 4500/270 ЭК-БМЧ",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1",
 "БУ 5000/320 ЭР-0",
 "БУ 5000/320 ЭР",

"БУ 5000/450 ЭР-Т",
 "БУ 5000/320 ЭК-БМЧ",
 "БУ 6000/400 ЭК-БМЧ",
 "БУ 6500/400 ЭР",
 "БУ 8000/500 ЭР",
 "БУ UNOC 320 ДЕ",
 "БУ UNOC 500 ДЕ",
 "НБО-1К",
 "НБО-Э",
 "НБО-Д",
 "ЗД86-1",
 "ЗД86-2",
 "БУ 1600/100 ДГУ",
 "БУ 1600/100 ЭУ",
 "БУ 2000/125 ЭП",
 "БУ 2000/125 ДЭП ",
 "БУ 2500/160 ДГУ М1",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПКМ1",
 "БУ 2900/200 ЭПК-БМ",
 "БУ 2900/200 ДЭП-БМ",
 "БУ 3900/225 ЭПК-БМ",
 "БУ 3900/225 ДЭП-БМ",
 "БУ 4200/250 ЭЧК-БМ",
 "БУ 4500/270 ЭЧК-БМ"

],
 "_7С":[
 "БУ 2500/160 ЭСК-БМ",

"БУ 3200/200 ЭУК-2М2",
 "БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 3200/200 ДГУ-1М",
 "БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 4200/250 ЭК-БМЧ",
 "БУ 4500/270 ЭК-БМЧ",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1",
 "БУ 5000/320 ЭР-0",
 "БУ 5000/320 ЭР",
 "БУ 5000/450 ЭР-Т",
 "БУ 5000/320 ЭК-БМЧ",
 "БУ 6000/400 ЭК-БМЧ",
 "БУ 6500/400 ЭР",
 "БУ 8000/500 ЭР",
 "БУ UNOC 320 ДЕ",
 "БУ UNOC 500 ДЕ",
 "НБО-1К",
 "НБО-Э",
 "НБО-Д",
 "ЗД86-1",
 "ЗД86-2",
 "БУ 1600/100 ДГУ",
 "БУ 1600/100 ЭУ",
 "БУ 2000/125 ЭП",
 "БУ 2000/125 ДЭП ",
 "БУ 2500/160 ДГУ М1",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПКМ1",
 "БУ 2900/200 ЭПК-БМ",
 "БУ 2900/200 ДЭП-БМ",
 "БУ 3900/225 ЭПК-БМ",
 "БУ 3900/225 ДЭП-БМ",
 "БУ 4200/250 ЭЧК-БМ",
 "БУ 4500/270 ЭЧК-БМ"

],
 "_8А":[
 "НБО-Д",

"ЗД86-1",
 "ЗД86-2"

],

"БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 5000/320 ДГУ-1Г",
 "БУ 5000/320 ДГУ-1",
 "БУ 1600/100 ДГУ",
 "БУ 2500/160 ДГУ М1"

],
 "_8В":[
 "БУ 3200/200 ДГУ-1М",

],
 "_8С":[
 "БУ UNOC 320 ДЕ",

"БУ UNOC 500 ДЕ",
 "БУ 2000/125 ДЭП",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/200 ДЭП-БМ",
 "БУ 3900/225 ДЭП-БМ"

],

"БУ 2500/160 ЭСК-БМ",

"_8D":[

"БУ 3200/200 ЭУК-2М2",
 "БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 4200/250 ЭК-БМЧ",
 "БУ 4500/270 ЭК-БМЧ",
 "БУ 5000/320 ЭУК-Я",
 "БУ 5000/320 ЭК-БМЧ",
 "БУ 6000/400 ЭК-БМЧ",
 "БУ 1600/100 ЭУ"

],

"_8E":[
 "БУ 5000/320 ЭР-0",

"БУ 5000/320 ЭР",
 "БУ 5000/450 ЭР-Т",
 "БУ 6500/400 ЭР",
 "БУ 2000/125 ЭП",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПКМ1",
 "БУ 2900/200 ЭПК-БМ",
 "БУ 8000/500 ЭР",
 "БУ 3900/225 ЭПК-БМ",
 "БУ 4200/250 ЭЧК-БМ",
 "БУ 4500/270 ЭЧК-БМ"

],

"_9A":[
 "БУ 1600/100 ДГУ",

"БУ 1600/100 ЭУ"

],

"_9B":[
 "БУ 1600/100 ДГУ",

"БУ 1600/100 ЭУ"

],

"_9C":[
 "БУ 2000/125 ЭП",

"БУ 2000/125 ДЭП",

],

"_9D":[
 "БУ 2500/160 ЭСК-БМ",

"БУ 2500/160 ДГУ М1"

],

"_9E":[
 "БУ 3200/200 ЭУК-2М2",

"БУ 3200/200 ЭУК-2М2У",
 "БУ 3200/200 ЭУК-2МЯ",
 "БУ 3200/200 ЭУК-3МА",
 "БУ 3200/200 ДГУ-1М",
 "БУ 3200/200 ДГУ-1У",
 "БУ 3200/200 ДГУ-1Т",
 "БУ 3200/200 ЭУ-1М",
 "БУ 3200/200 ЭУ-1У",
 "БУ 2900/175 ДЭП-2",
 "БУ 2900/175 ЭПК",
 "БУ 2900/175 ЭПКМ1",

```

"БУ 2900/200 ЭПК-БМ",
"БУ 2900/200 ДЭП-БМ"
    ],
    "_9F":[
        "БУ 4200/250 ЭК-БМЧ",
        "БУ 4500/270 ЭК-БМЧ ",
        "НБО-1К",
        "НБО-Э",
        "НБО-Д",
        "БУ 3900/225 ЭПК-БМ",
        "БУ 3900/225 ДЭП-БМ",
        "БУ 4200/250 ЭЧК-БМ",
        "БУ 4500/270 ЭЧК-БМ"
    ],
    "_9G":[
        "БУ 5000/320 ЭУК-Я",
        "БУ 5000/320 ДГУ-1Г",
        "БУ 5000/320 ДГУ-1",
        "БУ 5000/320 ЭР-0",
        "БУ 5000/320 ЭР",
        "БУ 5000/450 ЭР-Т",
        "БУ 5000/320 ЭК-БМЧ ",
        "БУ UNOC 320 ДЕ",
        "ЗД86-1",
        "ЗД86-2"
    ],
    "_9H":[
        "БУ 6000/400 ЭК-БМЧ",
        "БУ 6500/400 ЭР"
    ],
    "_9J":[
        "БУ 8000/500 ЭР ",
        "БУ UNOC 500 ДЕ"
    ]
}

};

//Обрані типи
private var selectedItems:Object;
//Поточне діалогове вікно
private var currDialogID:int = -1;
//Кількість діалогових вікон
private var dialogNum:int;
//Підходящі бурильні установки
private var allRigs:Array;
/**
 * Конструктор ініціалізації
 */
public function RigProblem(view:MovieClip)
{
    super(view);
}

/**
 * Створення задачі
 */
override public function create():void {
    dialogNum = 9;
    selectedItems = { };
    for (var i:int = 1; i <= dialogNum; i++)
    {
        selectedItems["_" + i] = null;
    }
    allRigs = [];
    for (var key:String in DATA) {
        allRigs = MathUtils.concatArraysNoDuplicate(allRigs, DATA[key]);
    }

    showDialog(1);

    view.addEventListener(MouseEvent.CLICK, onClick);
    view.addEventListener(MouseEvent.MOUSE_OVER, function(e:MouseEvent):void {
        if (e.target.parent && e.target.parent.name
            && new RegExp("_[1-9]").test(e.target.parent.name)
            && new RegExp("_" + (e.target.parent.name as String).substr(1, 1) + "[A-Z]").test(e.target.name)

```

```

    && e.target.mouseEnabled) {
        TweenMax.to(e.target, 0.25, { scaleX:0.95, scaleY:0.95 } );
    }
});
view.addEventListener(MouseEvent.MOUSE_OUT, function(e:MouseEvent):void {
    if (e.target.parent && e.target.parent.name
        && new RegExp("_[1-9]").test(e.target.parent.name)
        && new RegExp("_" + (e.target.parent.name as String).substr(1, 1) + "[A-Z]").test(e.target.name)
        && e.target.mouseEnabled) {
        TweenMax.to(e.target, 0.25, { scaleX:1, scaleY:1 } );
    }
});

//Додання обробників подій інтерактивної взаємодії
view["_btnClose_"].addEventListener(MouseEvent.CLICK, function(e:MouseEvent):void {
    Main.instance.showNextProblem();
});
view["_btnClose_"].visible = false;
view["_btnPrev_"].addEventListener(MouseEvent.CLICK, function(e:MouseEvent):void {

    if (currDialogID > 1) {

        var _sid:String = "_" + (currDialogID - 1);
        var _psi:String = selectedItems[_sid];
        selectedItems[_sid] = null;
        showDialog(currDialogID - 1);
        selectedItems[_sid] = _psi;

        TweenMax.to(view["_btnPrev_"], 0.2, { alpha:0, yoyo:true, repeat:1 } );

    }
});
view["_btnNext_"].addEventListener(MouseEvent.CLICK, function(e:MouseEvent):void {

    if (currDialogID < dialogNum) {
        var _sid:String = "_" + (currDialogID + 1);
        var _psi:String = selectedItems[_sid];
        selectedItems[_sid] = null;
        showDialog(currDialogID + 1);
        selectedItems[_sid] = _psi;

        TweenMax.to(view["_btnNext_"], 0.2, { alpha:0, yoyo:true, repeat:1 } );

    }
});

private function onClick(e:MouseEvent):void {
if (e.target.parent && e.target.parent.name
    && new RegExp("_[1-9]").test(e.target.parent.name)
    && new RegExp("_" + (e.target.parent.name as String).substr(1, 1) + "[A-Z]").test(e.target.name)) {

    var re:RegExp = new RegExp("_" + currDialogID + "[A-Z]");
    for (var i:int = 0; i < e.target.parent.numChildren; i++)
    {
        var ch:DisplayObject = e.target.parent.getChildAt(i);
        if (re.test(ch.name) && ch.hasOwnProperty("gITw")) {
            (ch["gITw"] as TweenMax).kill();
            ch.filters = null;
        }
    }
    selectedItems[e.target.parent.name] = e.target.name;

    e.target["gITw"] =
        TweenMax.to(e.target, 0.2, { glowFilter: { color:0x000000, blurX:4, blurY:4, strength:4,
alpha:1
            }, yoyo:true, repeat:-1});

    var arr:Array = DATA[e.target.name] as Array;
    var rigs:Array = getSuitableRigs();
    rigs = MathUtils.getCommonElements(rigs, arr);

    if (isSolved()) {
        trace("Solved:" + rigs);
        if (rigs.length == 0) {
            view["_suitableRigsTxt_"].text = "Підходящих бурильних установок немає у
бази";
        } else {

```



```

view["_suitableRigsTxt_"].text = (rigs.length > 1 ? "Підходящі БУ:" : "Підходяща БУ:");
    for (var j:int = 0; j < rigs.length - 1; j++)
    {
        view["_suitableRigsTxt_"].text += "{"+rigs[j] + "}";
    }
    view["_suitableRigsTxt_"].text += "{" + rigs[rigs.length - 1] + "}";
    }
    TextPrintAnimator.animateText(view["_suitableRigsTxt_"], 0.2, null, 1);
    view["_btnClose_"].visible = true;
    } else {
        showNextDialog();
    }
    }
}

/**
 * @return Повертає підходящі БУ
 */
private function getSuitableRigs():Array {
    var rigs:Array = allRigs; var c:int = 0;
    for (var i:int = 1; i <= dialogNum; i++)
    {
        if (selectedItems["_" + i] != null)
        {
            rigs = MathUtils.getCommonElements(DATA[selectedItems["_" + i]], rigs);
        }
    }
    return rigs;
}

/**
 * Фільтрувати типи БУ
 */
private function filterTypeRigs():void {
    var re:RegExp = new RegExp("_" + currDialogID + "[A-Z]");
    var existSuitableRigType:Boolean = false;
    var rigs:Array = getSuitableRigs();
    for (var key:String in DATA)
    {
        if (re.test(key)) {
            var b:Boolean = MathUtils.getCommonElements(DATA[key], rigs).length > 0;

            if (b) {
                existSuitableRigType = true;
                view["_" + currDialogID][key].alpha = 1;
                view["_" + currDialogID][key].mouseEnabled = true;
            } else {
                view["_" + currDialogID][key].alpha = 0.25;
                view["_" + currDialogID][key].mouseEnabled = false;
            }
        }
    }
}

/**
 * @return Повертає айді першого не ще обраного діалогового вікна
 */
private function getFirstNotSelectedDialogID():int {
    for (var i:int = 1; i <= dialogNum; i++)
    {
        if (selectedItems["_" + i] == null) return i;
    }
    return -1;
}

/**
 * Показати діалогове вікно
 * @param id айді діалогового вікна
 */
private function showDialog(id:int):void {
    if (currDialogID != -1) {
        TweenMax.to(view["_" + currDialogID], 0.25, { x: -Main.WIDTH, ease:Linear.easeNone });
    }
}

```

```

view["_" + id].x = Main.WIDTH;
view["_" + id].y = 0;
view.mouseChildren = view.mouseEnabled = false;
TweenMax.to(view["_" + id], 0.25, { x: 0, ease:Linear.easeNone, onComplete:function() {
    view.mouseChildren = view.mouseEnabled = true;

    for (var i:int = 1; i <= dialogNum; i++)
    {
        view["_" + i].visible = false;
    }
    view["_" + currDialogID].visible = true;
}});

currDialogID = id;
filterTypeRigs();

view["_btnPrev_"].visible = currDialogID > 1;
view["_btnNext_"].visible = currDialogID < dialogNum;
TweenMax.delayedCall(0.25, function() {
    TextPrintAnimator.animateText(view["_" + currDialogID]["_desc_"], 0.1, null, 1);
});
}
/**
 * Показати наступне діалогове вікно
 */
private function showNextDialog():void {
    var fid:int = getFirstNotSelectedDialogID();
    if (fid != -1) {
        showDialog(fid);
    }
}
/**
 * Знищення та очищення ресурсів
 */
override public function destroy():void {
    view.removeEventListener(MouseEvent.CLICK, onClick);

    for (var i:int = 1; i <= dialogNum; i++)
    {
        if (selectedItems["_" + i] != null) {
            (view["_" + i][selectedItems["_" + i]]["glTw"] as TweenMax).kill();
            view["_" + i][selectedItems["_" + i]].filters = null;
        }
    }
}
/**
 * Перевірка чи вирішена задача чи ні
 * @return стан вирішення задачі
 */
override public function isSolved():Boolean {
    return getFirstNotSelectedDialogID() == -1;
}
}
}

```

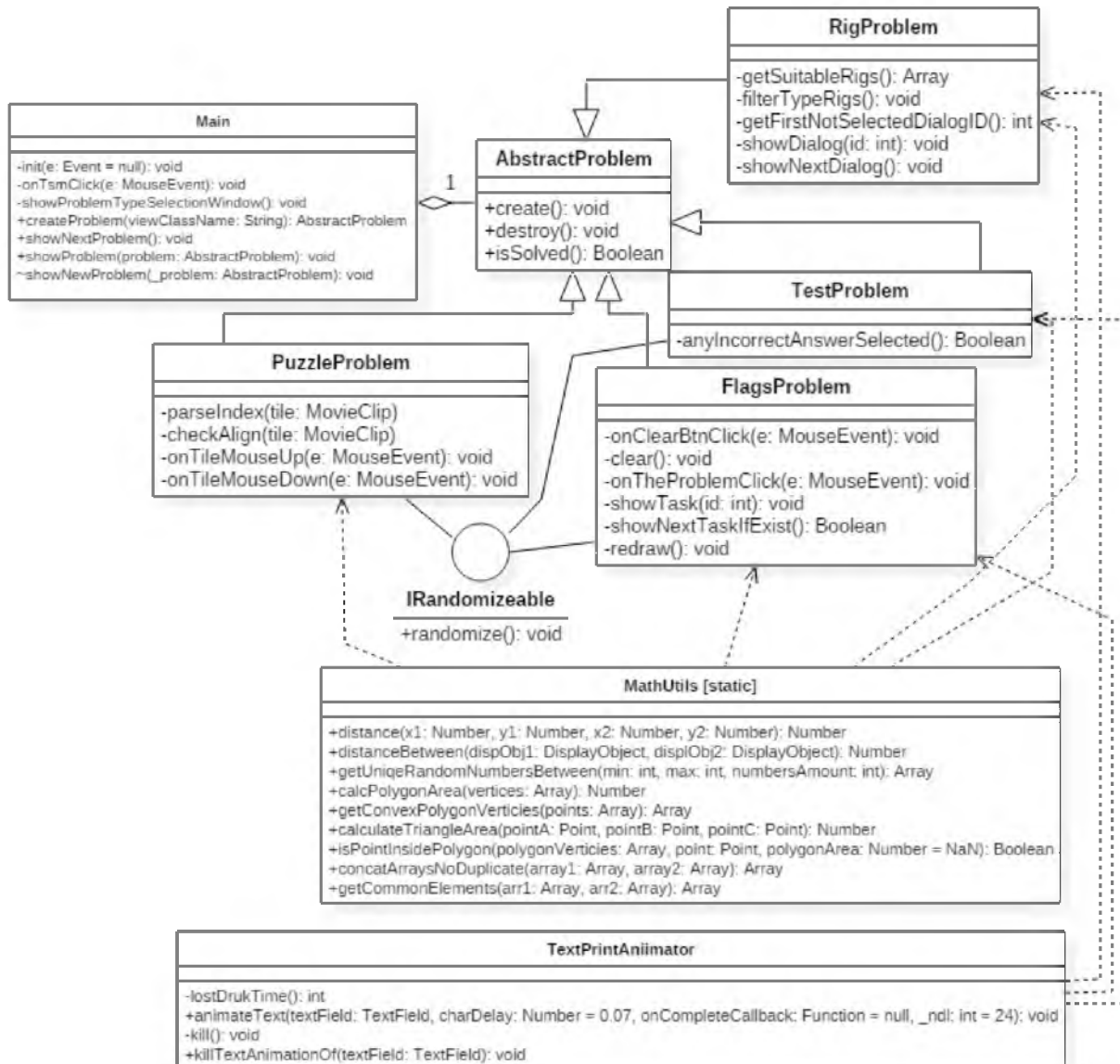
ДОДАТОК Б

Концептуальна схема реалізованого модуля інтелімедійної інформаційної системи підтримки прийняття рішень в нафтогазовій справі



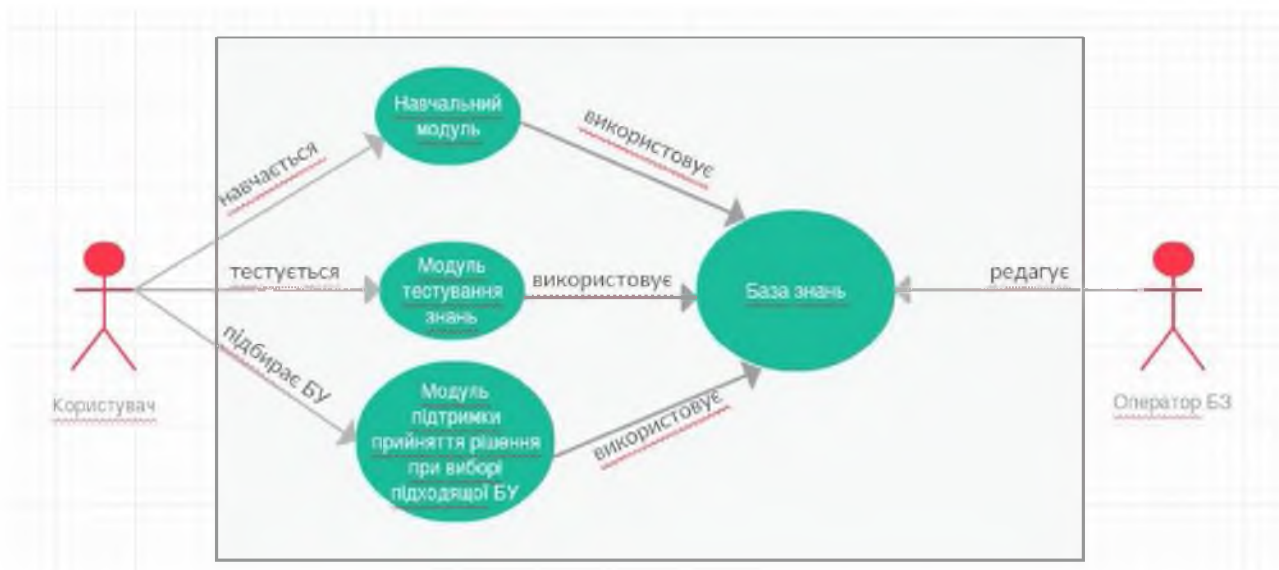
ДОДАТОК В

Діаграма класів реалізованого модуля інтелемедійної інформаційної системи підтримки прийняття рішень в нафтогазовій справі



ДОДАТОК Г

Діаграма прецедентів реалізованого модуля інтелімедійної інформаційної системи підтримки прийняття рішень в нафтогазовій справі



ДОДАТОК Д

Таблиця Д1

Класифікована таблиця назв бурових установок

1А	БУ 2500/160 ЭСК-БМ БУ 3200/200 ЭУК-2М2 БУ 3200/200 ЭУК-2М2У БУ 3200/200 ЭУК-2МЯ БУ 3200/200 ЭУК-3МА БУ 3200/200 ДГУ-1М БУ 3200/200 ДГУ-1У БУ 3200/200 ДГУ-1Т БУ 3200/200 ЭУ-1М БУ 3200/200 ЭУ-1У БУ 4200/250 ЭК-БМЧ БУ 4500/270 ЭК-БМЧ БУ 5000/320 ЭУК-Я БУ 5000/320 ДГУ-1Т БУ 5000/320 ДГУ-1 БУ 5000/320 ЭР-0 БУ 5000/320 ЭР БУ 5000/450 ЭР-Т БУ 5000/320 ЭК-БМЧ БУ 6000/400 ЭК-БМЧ БУ 6500/400 ЭР БУ 8000/500 ЭР БУ UNOC 320 ДЕ БУ UNOC500ДЕ НБО-1К НБО-Э НБО-Д ЗД86-1 ЗД86-2	БУ 1600/100 ДГУ БУ 1600/100 ЭУ БУ 2000/125 ЭП БУ 2000/125 ДЭП БУ 2500/160 ДГУ М1 БУ 2900/175 ДЭП-2 БУ 2900/175 ЭПК БУ 2900/175 ЭПКМ1 БУ 2900/200 ЭПК-БМ БУ 2900/200 ДЭП-БМ БУ 3900/225 ЭПК-БМ БУ 3900/225 ДЭП-БМ БУ 4200/250 ЭЧК-БМ БУ 4500/270 ЭЧК-БМ
2А	БУ 3200/200 ДГУ-1М БУ 3200/200 ДГУ-1У БУ 3200/200 ДГУ-1Т БУ 5000/320 ДГУ-1Т БУ 5000/320 ДГУ-1 БУ UNOC 320 ДЕ БУ UNOC500ДЕ НБО-Д ЗД86-1 ЗД86-2	БУ 1600/100 ДГУ БУ 2000/125 ДЭП БУ 2500/160 ДГУ М1 БУ 2900/175 ДЭП-2 БУ 2900/200 ДЭП-БМ БУ 3900/225 ДЭП-БМ

2Б	БУ 2500/160 ЭСК-БМ БУ 3200/200 ЭУК-2М2 БУ 3200/200 ЭУК-2М2У БУ 3200/200 ЭУК-2МЯ БУ 3200/200 ЭУК-3МА БУ 3200/200 ЭУ-1М БУ 3200/200 ЭУ-1У БУ 4200/250 ЭК-БМЧ БУ 4500/270 ЭК-БМЧ БУ 5000/320 ЭУК-Я БУ 5000/320 ЭР-0 БУ 5000/320 ЭР БУ 5000/450 ЭР-Т БУ 5000/320 ЭК-БМЧ БУ 6000/400 ЭК-БМЧ БУ 6500/400 ЭР БУ 8000/500 ЭР	БУ 1600/100 ЭУ БУ 2000/125 ЭП БУ 2000/125 ДЭП БУ 2900/175 ДЭП-2 БУ 2900/175 ЭПК БУ 2900/175 ЭПКМ1 БУ 2900/200 ЭПК-БМ БУ 2900/200 ДЭП-БМ БУ 3900/225 ЭПК-БМ БУ 3900/225 ДЭП-БМ БУ 4200/250 ЭЧК -БМ БУ 4500/270 ЭЧК -БМ
3А	БУ 2500/160 ЭСК-БМ БУ 3200/200 ЭУК-2М2 БУ 3200/200 ЭУК-2М2У БУ 3200/200 ЭУК-2МЯ БУ 3200/200 ЭУК-3МА БУ 3200/200 ДГУ-1М БУ 3200/200 ДГУ-1У БУ 3200/200 ДГУ-1Т БУ 3200/200 ЭУ-1М БУ 3200/200 ЭУ-1У БУ 4200/250 ЭК-БМЧ БУ 4500/270 ЭК-БМЧ БУ 5000/320 ЭУК-Я БУ 5000/320 ДГУ-1Т БУ 5000/320 ДГУ-1 БУ 5000/320 ЭР-0 БУ 5000/320 ЭР БУ 5000/450 ЭР-Т БУ 5000/320 ЭК-БМЧ БУ 6000/400 ЭК-БМЧ БУ 6500/400 ЭР БУ 8000/500 ЭР БУ UNOC 320 ДЕ БУ UNOC500ДЕ НБО-1К НБО-Э НБО-Д ЗД86-1	БУ 1600/100 ДГУ БУ 1600/100 ЭУ БУ 2000/125 ЭП БУ 2000/125 ДЭП БУ 2500/160 ДГУ М1 БУ 2900/175 ДЭП-2 БУ 2900/175 ЭПК БУ 2900/175 ЭПКМ1 БУ 2900/200 ЭПК-БМ БУ 2900/200 ДЭП-БМ БУ 3900/225 ЭПК-БМ БУ 3900/225 ДЭП-БМ БУ 4200/250 ЭЧК -БМ БУ 4500/270 ЭЧК -БМ

	3Д86-2	
4А	БУ 3200/200 ДГУ-1М БУ 3200/200 ДГУ-1У БУ 3200/200 ДГУ-1Т БУ 3200/200 ЭУ-1М БУ 3200/200 ЭУ-1У БУ 5000/320 ДГУ-1Т БУ 5000/320 ДГУ-1 БУ 5000/320 ЭР-0 БУ 5000/320 ЭР БУ 5000/450 ЭР-Т БУ 6500/400 ЭР БУ 8000/500 ЭР БУ UNOC 320 ДЕ БУ UNOC500ДЕ НБО-1КНБО-Э НБО-Д 3Д86-1 3Д86-2	БУ 1600/100 ДГУ БУ 1600/100 ЭУ БУ 2000/125 ЭП БУ 2000/125 ДЭП БУ 2500/160 ДГУ М1 БУ 2900/175 ДЭП-2 БУ 2900/175 ЭПК БУ 2900/200 ДЭП-БМ БУ 3900/225 ДЭП-БМ
4Б	БУ 2500/160 ЭСК-БМ БУ 3200/200 ЭУК-2М2 БУ 3200/200 ЭУК-2М2У БУ 3200/200 ЭУК-2МЯ БУ 3200/200 ЭУК-3МА БУ 4200/250 ЭК-БМЧ БУ 4500/270 ЭК-БМЧ БУ 5000/320 ЭУК-Я БУ 5000/320 ЭК-БМЧ БУ 6000/400 ЭК-БМЧ НБО-1К	БУ 2900/175 ЭПК БУ 2900/175 ЭПКМ1 БУ 2900/200 ЭПК-БМ БУ 3900/225 ЭПК-БМ БУ 4200/250 ЭЧК -БМ БУ 4500/270 ЭЧК -БМ
5А	БУ 3200/200 ЭУК-2МЯ БУ 5000/320 ЭУК-Я	
5Б	БУ 2500/160 ЭСК-БМ БУ 3200/200 ЭУК-2М2 БУ 3200/200 ЭУК-2М2У БУ 3200/200 ЭУК-3МА БУ 3200/200 ДГУ-1М БУ 3200/200 ДГУ-1У БУ 3200/200 ЭУ-1М БУ 3200/200 ЭУ-1У БУ 4200/250 ЭК-БМЧ БУ 4500/270 ЭК-БМЧ БУ 5000/320 ДГУ-1 БУ 5000/320 ЭР-0	БУ 1600/100 ДГУ БУ 1600/100 ЭУ БУ 2000/125 ЭП БУ 2000/125 ДЭП БУ 2500/160 ДГУ М1 БУ 2900/175 ДЭП-2 БУ 2900/175 ЭПК БУ 2900/175 ЭПКМ1 БУ 2900/200 ЭПК-БМ БУ 2900/200 ДЭП-БМ БУ 3900/225 ЭПК-БМ БУ 3900/225 ДЭП-БМ

	БУ 5000/320 ЭР БУ 5000/320 ЭК-БМЧ БУ 6000/400 ЭК-БМЧ БУ 6500/400 ЭР БУ 8000/500 ЭР БУ UNOC 320 ДЕ БУ UNOC500ДЕ НБО-1К НБО-Э НБО-Д ЗД86-1 ЗД86-2	БУ 4200/250 ЭЧК-БМ БУ 4500/270 ЭЧК-БМ
5В	БУ 3200/200 ДГУ-1Т БУ 5000/320 ДГУ-1Т БУ 5000/450 ЭР-Т	
6А	БУ 4200/250 ЭК-БМЧ БУ 4500/270 ЭК-БМЧ БУ 5000/320 ЭК-БМЧ БУ 6000/400 ЭК-БМЧ	БУ 2900/200 ЭПК-БМ БУ 2900/200 ДЭП-БМ БУ 3900/225 ЭПК-БМ БУ 3900/225 ДЭП-БМ БУ 4200/250 ЭЧК-БМ БУ 4500/270 ЭЧК-БМ
6Б 6В 6Г	БУ 3200/200 ЭУК-2М2 БУ 3200/200 ЭУК-2М2У БУ 3200/200 ЭУК-2МЯ БУ 3200/200 ЭУК-3МА БУ 3200/200 ДГУ-1М БУ 3200/200 ДГУ-1У БУ 3200/200 ДГУ-1Т БУ 3200/200 ЭУ-1М БУ 3200/200 ЭУ-1У БУ 5000/320 ЭУК-Я БУ 5000/320 ДГУ-1Т БУ 5000/320 ДГУ-1 БУ 5000/320 ЭР-0 БУ 5000/320 ЭР БУ 5000/450 ЭР-Т БУ 6500/400 ЭР БУ 8000/500 ЭР БУ UNOC 320 ДЕ БУ UNOC500ДЕ НБО-1К НБО-Э НБО-Д ЗД86-1	БУ 1600/100 ДГУ БУ 1600/100 ЭУ БУ 2000/125 ЭП БУ 2000/125 ДЭП БУ 2500/160 ДГУ М1 БУ 2900/175 ДЭП-2 БУ 2900/175 ЭПК БУ 2900/175 ЭПК М1

	3Д86-2	
7А	БУ 2500/160 ЭСК-БМ	БУ 1600/100 ДГУ
7Б	БУ 3200/200 ЭУК-2М2	БУ 1600/100 ЭУ
7В	БУ 3200/200 ЭУК-2М2У	БУ 2000/125 ЭП
	БУ 3200/200 ЭУК-2МЯ	БУ 2000/125 ДЭП
	БУ 3200/200 ЭУК-3МА	БУ 2500/160 ДГУ М1
	БУ 3200/200 ДГУ-1М	БУ 2900/175 ДЭП-2
	БУ 3200/200 ДГУ-1У	БУ 2900/175 ЭПК
	БУ 3200/200 ДГУ-1Т	БУ 2900/175 ЭПКМ1
	БУ 3200/200 ЭУ-1М	БУ 2900/200 ЭПК-БМ
	БУ 3200/200 ЭУ-1У	БУ 2900/200 ДЭП-БМ
	БУ 4200/250 ЭК-БМЧ	БУ 3900/225 ЭПК-БМ
	БУ 4500/270 ЭК-БМЧ	БУ 3900/225 ДЭП-БМ
	БУ 5000/320 ЭУК-Я	БУ 4200/250 ЭЧК-БМ
	БУ 5000/320 ДГУ-1Т	БУ 4500/270 ЭЧК-БМ
	БУ 5000/320 ДГУ-1	
	БУ 5000/320 ЭР-0	
	БУ 5000/320 ЭР	
	БУ 5000/450 ЭР-Т	
	БУ 5000/320 ЭК-БМЧ	
	БУ 6000/400 ЭК-БМЧ	
	БУ 6500/400 ЭР	
	БУ 8000/500 ЭР	
	БУ UNOC 320 ДЕ	
	БУ UNOC500ДЕ	
	НБО-1К	
	НБО-Э	
	НБО-Д	
	3Д86-1	
	3Д86-2	
8А	БУ 3200/200 ДГУ-1М	БУ 1600/100 ДГУ
	БУ 3200/200 ДГУ-1У	БУ 2000/125 ДЭП
	БУ 3200/200 ДГУ-1Т	БУ 2500/160 ДГУ М1
	БУ 5000/320 ДГУ-1Т	БУ 2900/175 ДЭП-2
	БУ 5000/320 ДГУ-1	БУ 2900/200 ДЭП-БМ
	БУ UNOC 320 ДЕ	БУ 3900/225 ДЭП-БМ
	БУ UNOC500ДЕ	
	НБО-Д	
	3Д86-1	
	3Д86-2	
8Б	БУ 2500/160 ЭСК-БМ	БУ 1600/100 ЭУ
	БУ 3200/200 ЭУК-2М2	БУ 2000/125 ЭП
	БУ 3200/200 ЭУК-2М2У	БУ 2000/125 ДЭЕ
	БУ 3200/200 ЭУК-2МЯ	БУ 2900/175 ДЭП-2
	БУ 3200/200 ЭУК-3МА	БУ 2900/175 ЭПК

	БУ 3200/200 ЭУ-1М БУ 3200/200 ЭУ-1У БУ 4200/250 ЭК-БМЧ БУ 4500/270 ЭК-БМЧ БУ 5000/320 ЭУК-Я БУ 5000/320 ЭР-0 БУ 5000/320 ЭР БУ 5000/450 ЭР-Т БУ 5000/320 ЭК-БМЧ БУ 6000/400 ЭК-БМЧ БУ 6500/400 ЭР БУ 8000/500 ЭР	БУ 2900/175 ЭПКМ1 БУ 2900/200 ЭПК-БМ БУ 2900/200 ДЭП-БМ БУ 3900/225 ЭПК-БМ БУ 3900/225 ДЭП-БМ БУ 4200/250 ЭЧК -БМ БУ 4500/270 ЭЧК -БМ
8А1	НБО-Д ЗД86-1 ЗД86-2	
8А2	БУ 3200/200 ДГУ-1М БУ 3200/200 ДГУ-1У БУ 3200/200 ДГУ-1Т БУ 5000/320 ДГУ-1Т БУ 5000/320 ДГУ-1	БУ 1600/100 ДГУ БУ 2500/160 ДГУ М1
8А3	БУ UNOC 320 ДЕ БУ UNOC500ДЕ	БУ 2000/125 ДЭП БУ 2900/175 ДЭП-2 БУ 2900/200 ДЭП-БМ БУ 3900/225 ДЭП-БМ
8Б1	БУ 2500/160 ЭСК-БМ БУ 3200/200 ЭУК-2М2 БУ 3200/200 ЭУК-2М2У БУ 3200/200 ЭУК-2МЯ БУ 3200/200 ЭУК-3МА БУ 3200/200 ЭУ-1М БУ 3200/200 ЭУ-1У БУ 4200/250 ЭК-БМЧ БУ 4500/270 ЭК-БМЧ БУ 5000/320 ЭУК-Я БУ 5000/320 ЭК-БМЧ БУ 6000/400 ЭК-БМЧ	БУ 1600/100 ЭУ
8Б2	БУ 5000/320 ЭР-0 БУ 5000/320 ЭР БУ 5000/450 ЭР-Т БУ 6500/400 ЭР БУ 8000/500 ЭР	БУ 2000/125 ЭП БУ 2900/175 ЭПК БУ 2900/175 ЭПКМ1 БУ 2900/200 ЭПК-БМ БУ 3900/225 ЭПК-БМ БУ 4200/250 ЭЧК -БМ БУ 4500/270 ЭЧК -БМ
9А		БУ 1600/100 ДГУ

		БУ 1600/100 ЭУ
9Б		БУ 1600/100 ДГУ БУ 1600/100 ЭУ
9В		БУ 2000/125 ЭП БУ 2000/125 ДЭП
9Г	БУ 2500/160 ЭСК-БМ	БУ 2500/160 ДГУ М1
9Д	БУ 3200/200 ЭУК-2М2 БУ 3200/200 ЭУК-2М2У БУ 3200/200 ЭУК-2МЯ БУ 3200/200 ЭУК-3МА БУ 3200/200 ДГУ-1М БУ 3200/200 ДГУ-1У БУ 3200/200 ДГУ-1Т БУ 3200/200 ЭУ-1М БУ 3200/200 ЭУ-1У	БУ 2900/175 ДЭП-2 БУ 2900/175 ЭПК БУ 2900/175 ЭПКМ1 БУ 2900/200 ЭПК-БМ БУ 2900/200 ДЭП-БМ
9Ж	БУ 4200/250 ЭК-БМЧ БУ 4500/270 ЭК-БМЧ НБО-1К НБО-Э НБО-Д	БУ 3900/225 ЭПК-БМ БУ 3900/225 ДЭП-БМ БУ 4200/250 ЭЧК-БМ БУ 4500/270 ЭЧК-БМ
9З	БУ 5000/320 ЭУК-Я БУ 5000/320 ДГУ-1Т БУ 5000/320 ДГУ-1 БУ 5000/320 ЭР-0 БУ 5000/320 ЭР БУ 5000/450 ЭР-Т БУ 5000/320 ЭК-БМЧ БУ UNOC 320 ДЕ ЗД86-1 ЗД86-2	
9К	БУ 6000/400 ЭК-БМЧ БУ 6500/400 ЭР	
9Л	БУ 8000/500 ЭР БУ UNOC500ДЕ	

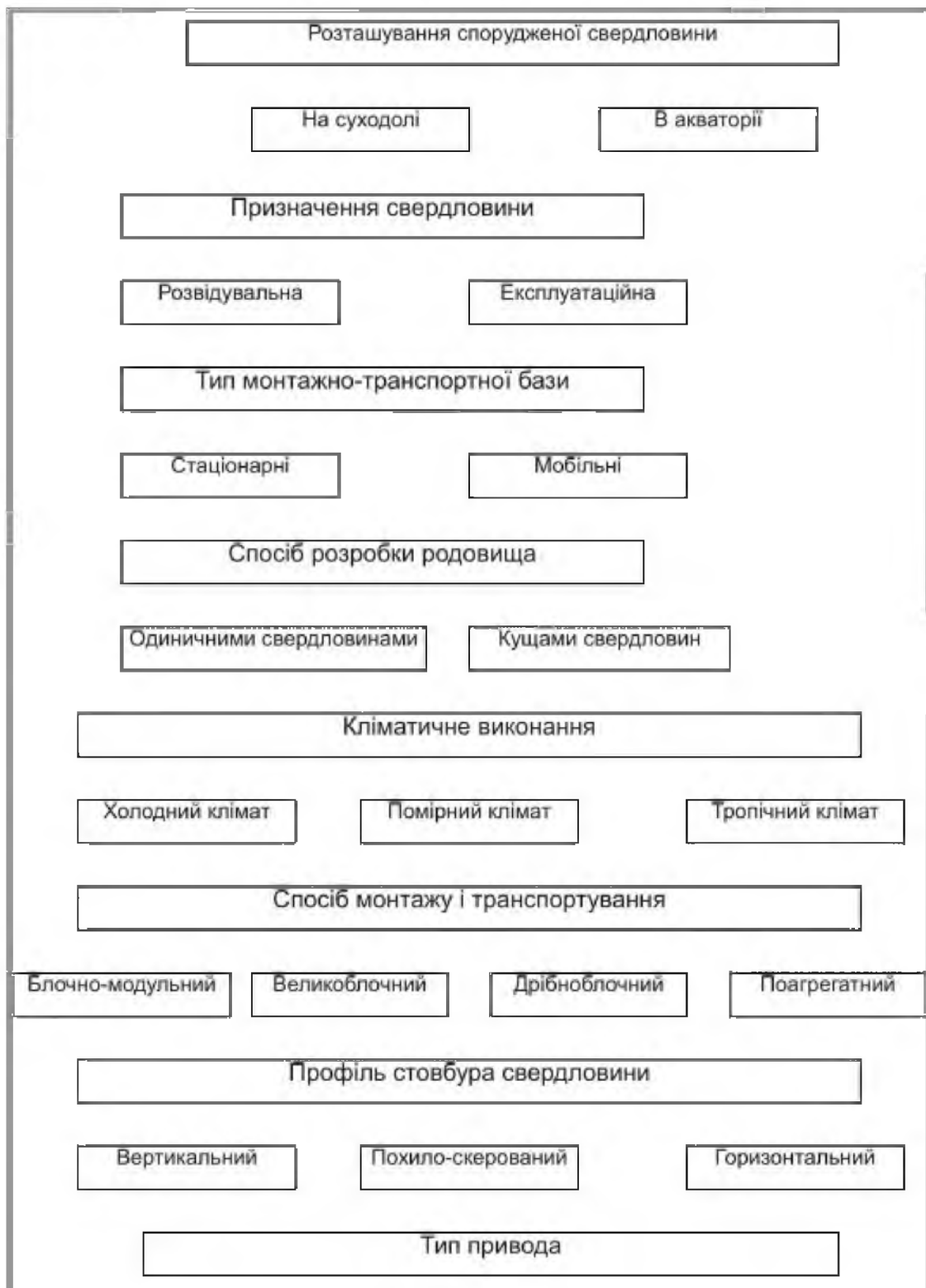
ДОДАТОК Е

Зображення усіх класів бурових установок використаних для реалізації модуля інтелектуальної інформаційної системи підтримки прийняття рішень в нафтогазовій справі



ДОДАТОК Ж

Схема класифікації БУ



ДОДАТОК И

Приклад тесту на базі графічного контенту
(робота з частинками графічного матеріалу)

 Пазл

 Виділення об'єкта

 Тест

 Підбір бурових установок

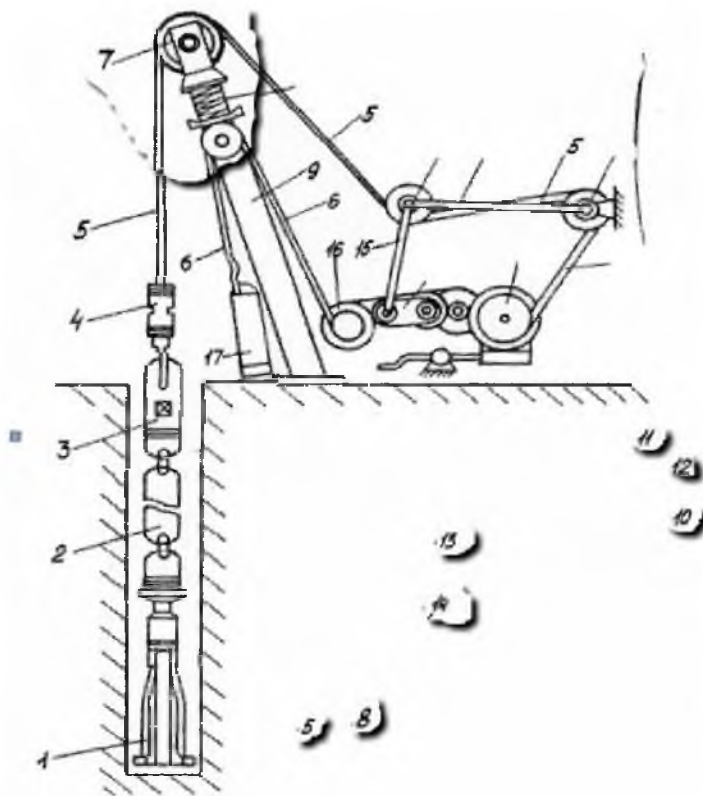


Рисунок 1.2 — Схема ударно-канатного буріння.

З усіх різновидностей ударного буріння в даний час застосовується тільки ударно-канатний спосіб (рис. 3.2).

Буровий снаряд, що складається з долота 1, ударної штанги 2, розсувної штанги-ножиць 3 і канатного замка 4, спускається в свердловину на інструментальному канаті 5, перекинутому через головний ролик 7 та амортизатор 8 шогли 9, обгинає стяжний 10 і направляючий 12 ролики балансірної рами 11.

При загальмованому барабані інструментальної лебідки 13, на якому закріплений кінець каната, шатуно-кривошипний механізм 14 і 15 балансірна рама приводиться в коливальний рух відносно осі направляючого ролика 12.

Відтяжний ролик балансірної рами, опускаючись, натягує канат і піднімає снаряд над вибоєм.

Піднімаючись вгору, ролик 10 звільнює канат, і снаряд під власною вагою падає на вибій, руйнуючи долотом породу.

У міру поглиблення свердловини, канат подовжують, змотуючи його з барабана 13.

Циліндричність свердловини забезпечується поворотом долота в результаті розкручування каната під навантаженням (під час підйому бурового снаряда) і скручування його при знятті навантаження (під час удару долота об породу).

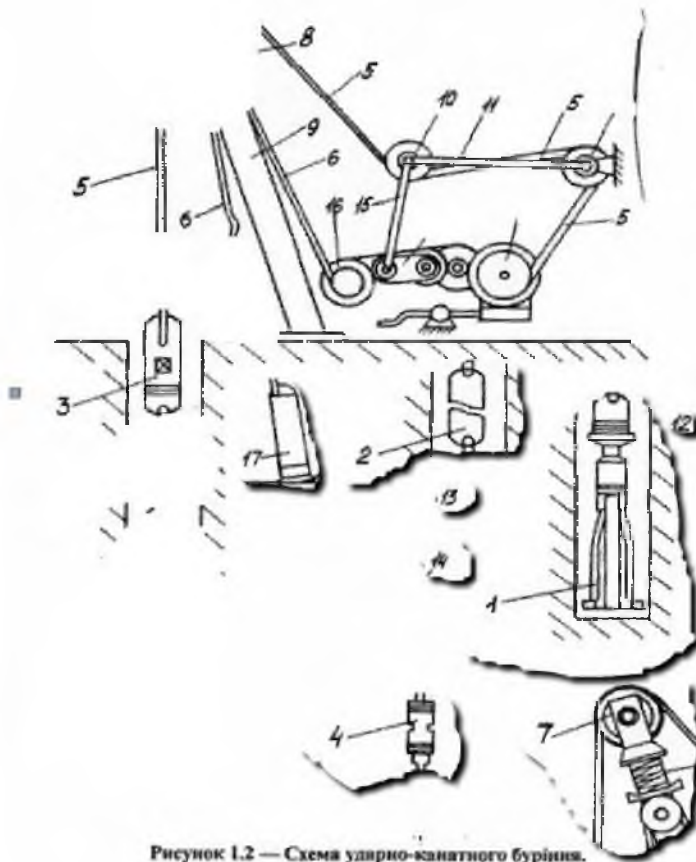


Рисунок 1.2 — Схема ударно-канатного буріння.

З усіх різновидностей ударного буріння в даний час застосовується тільки ударно-канатний спосіб (рис. 3.2).

Буровий снаряд, що складається з долота 1, ударної штанги 2, розсувної штанги-ножиць 3 і канатного замка 4, спускається в свердловину на інструментальному канаті 5, перекинутому через головний ролик 7 та амортизатор 8 щогли 9, обгинає стяжний 10 і направляючий 12 ролики балансірної рами 11.

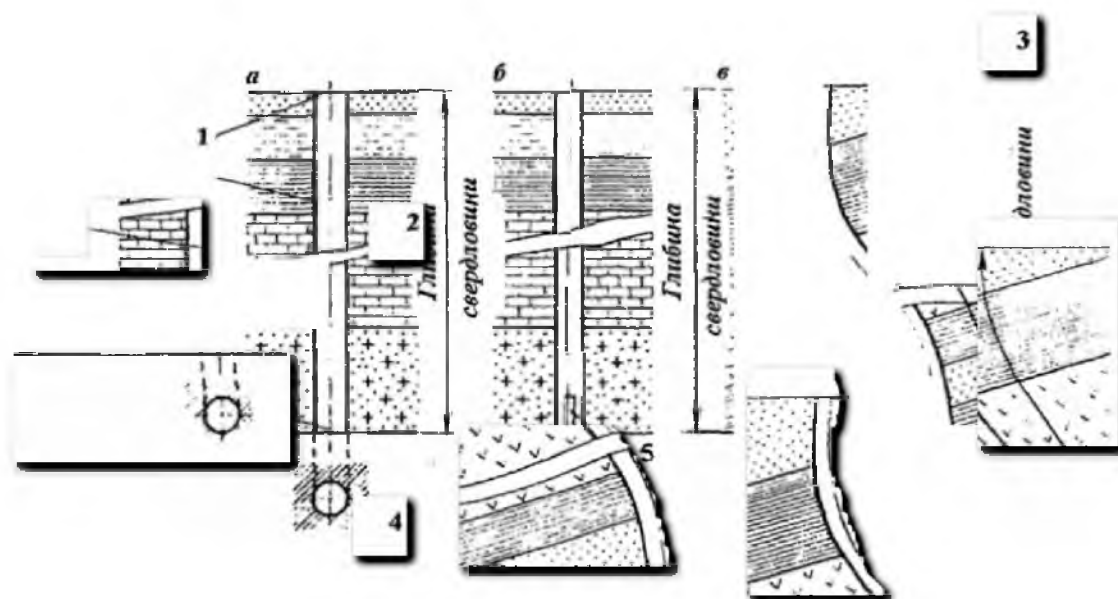
При загальмованому барабані інструментальної лебідки 13, на якому закріплений кінець каната, шатунно-кривошипним механізмом 14 і 15 балансірна рама приводиться в коливальний рух відносно осі направляючого ролика 12.

Відтяжний ролик балансірної рами, опускаючись, натягує канат і піднімає снаряд над вибоєм.

Піднімаючись вгору, ролик 10 звільнює канат, і снаряд під власною вагою падає на вибій, руйнуючи долотом породу.

У міру поглиблення свердловини, канат подовжують, змотуючи його з барабана 13.

Циліндричність свердловини забезпечується поворотом долота в результаті розкручування каната під навантаженням (під час підйому бурового снаряда) і скручування його при знятті навантаження (під час удару долота об породу).



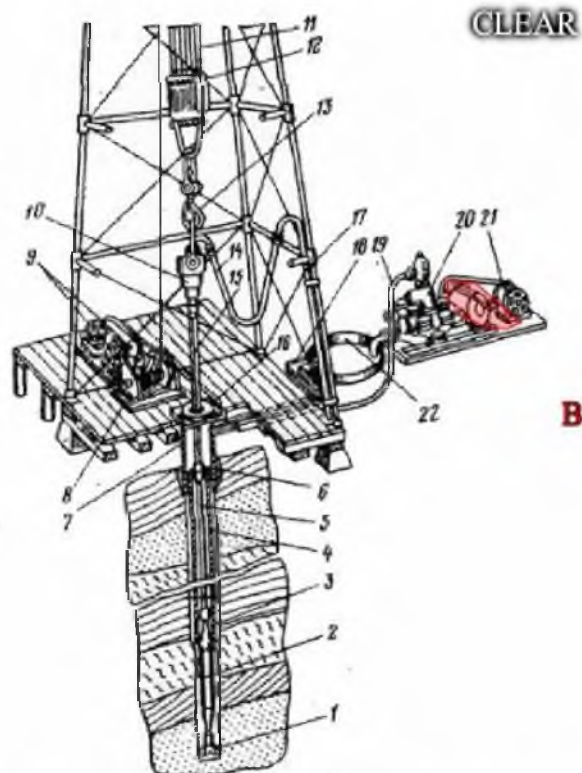
*а, б-вертикальні; в-похила; а,в-буріння без відбору керну;
б-буріння з відбором керну;*

1-уста; 2-стінка (ствол); 3-вісь; 4-вибій; 5-кern.

Рисунок 1.1 — Схема свердловини.

ДОДАТОК К

Приклад тесту на базі графічного контенту
(робота в тестовому режимі)



Виділіть ДВИГУНИ

Рисунок 1.4— Схема установки для буріння нафтових та газових свердловин обертовим способом.

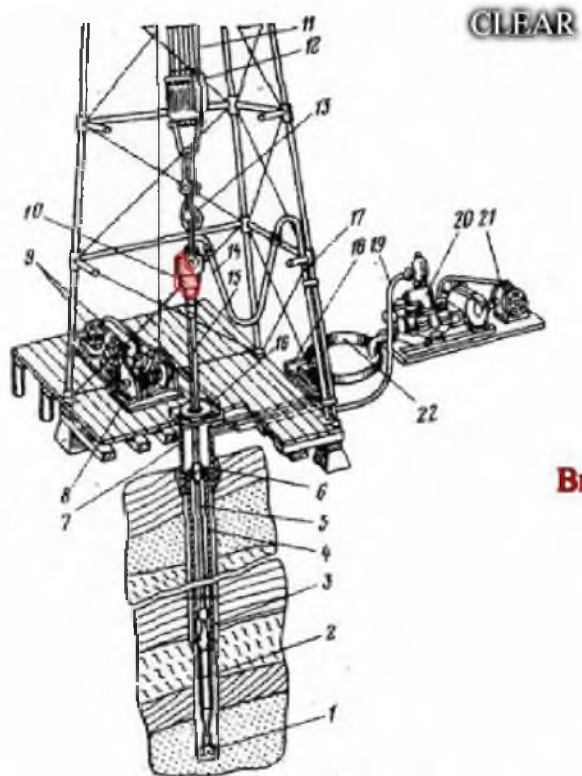


Рисунок 1.4— Схеми установок для буріння нафтових та газових свердловин обертовим способом.

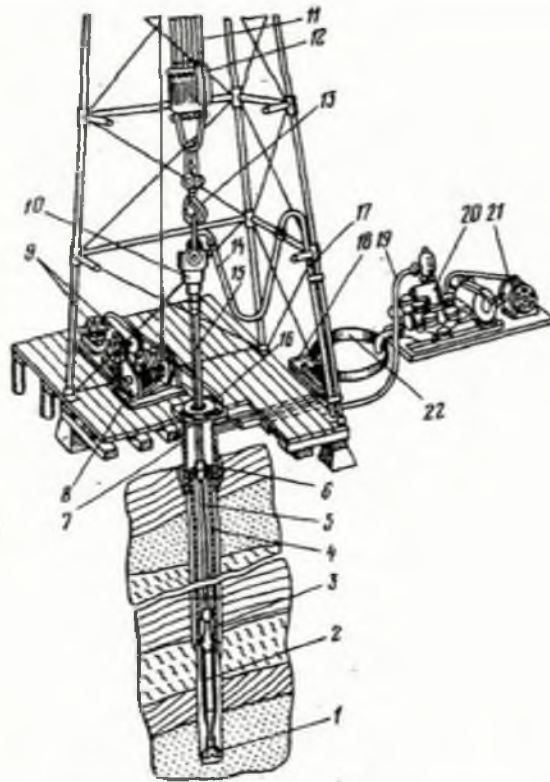


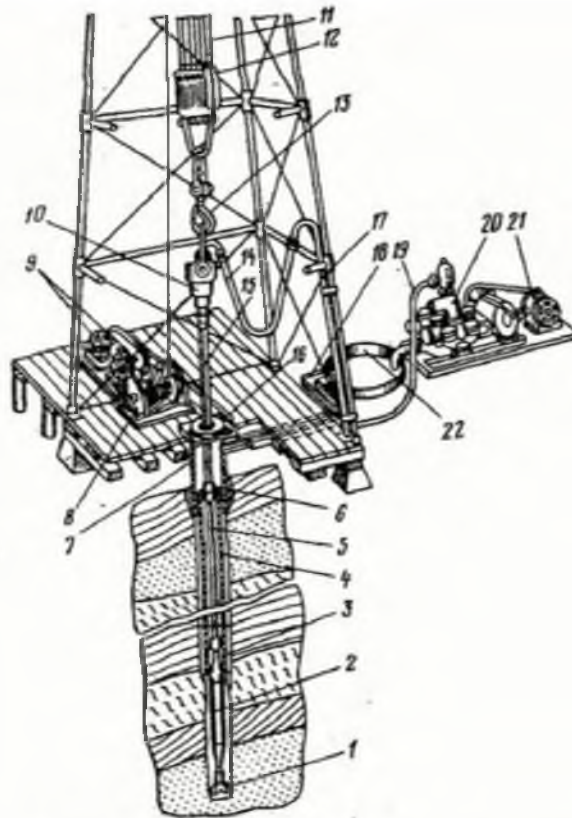
Рисунок 1.4— Схема установки для буріння нафтових та газових свердловин обертовим способом.

гнучкий буровий

- 14
- 11
- 10
- 2



- 0
- 1



бурильні труби це:

- 5
- 14
- 4
- 22



Рисунок 1.4— Схема установки для буріння нафтових та газових свердловин обертовим способом.

✓ 1
✗ 0

ДОДАТОК Л

Приклад тесту на базі графічного контенту
(режим підтримки прийняття рішень)



Виберіть призначення свердловини:

Розвідувальна



Експлуатаційна



Виберіть спосіб монтажу і транспортування:

Блочно-модульний



Дрібноблочний




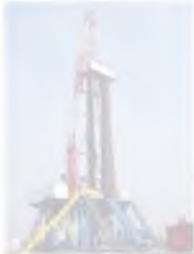




Великоблочний



Поагрегатний



← **Виберіть глибину буріння:** ✕

≤ 1250м	1250м - 1600м	1600м - 2000м	2000м - 2500м	
				
2500м - 3200м	3200м - 4000м	4000м - 5000м	5000м - 6500м	6500м - 8000м
				

Погодони БУ: (БУ 5000/320 СУК-Я)

↔ **Виберіть тип привода:** ✕

Автономний:		Неавтономний:	
Дизельний	Дизель-електричний	Електричний змінного струму	
			
Дизель-гідролічний		Електричний постійного струму	
			

Погодони БУ: (БУ 3200/280 СУ-1М); (БУ 3200)

ДОДАТОК М

Акт передачі у промислове використання методики реалізації процедур підтримки прийняття технологічних рішень в процесі буріння на основі інтелімедійних баз знань та баз даних експертного досвіду

АКТ

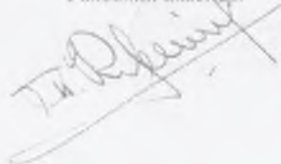
передачі у промислове використання методики реалізації процедур підтримки прийняття технологічних рішень в процесі буріння на основі інтелімедійних баз знань та баз даних експертного досвіду

З метою забезпечення інтелектуальної підтримки прийняття рішень при виборі режимів процесу буріння шляхом встановлення відповідних значень керованих параметрів, Івано-Франківський національний технічний університет нафти і газу, надалі ІФНТУНГ, розробив «Методику реалізації процедур підтримки прийняття технологічних рішень в процесі буріння на основі інтелімедійних баз знань та баз даних експертного досвіду» і передає, а Прикарпатське управління бурових робіт, надалі Прикарпатське УБР, приймає у впровадження для підвищення кваліфікації операторів технологічного процесу при бурінні свердловин.

Методика розроблена кафедрою програмного забезпечення автоматизованих систем ІФНТУНГ (автори: Мельник В. Д., Шекета В. І.).

Від Прикарпатського УБР:

Головний інженер:



Т. М. Кушнір

Від ІФНТУНГ:

Проректор з наукової роботи і міжнародної діяльності, д.т.н.:

О. М. Карпач

Завідувач кафедри програмного забезпечення автоматизованих систем, д.т.н.:

В. М. Юрчишин

Допент кафедри програмного забезпечення автоматизованих систем, к.т.н.:

В. І. Шекета

Асистент кафедри документознавства та інформаційної діяльності

В. Д. Мельник



ДОДАТОК Н

Акт впровадження у навчальний процес наукових результатів дисертаційної роботи асистента кафедри документознавства та інформаційної діяльності Мельника В.Д. «Інтелімедійна інформаційна система підтримки прийняття рішень в нафтогазовій справі»



ЗАТВЕРДЖУЮ:

Проректор з науково-педагогічної роботи Івано-Франківського національного технічного університету нафти і газу,
д-р техн. наук, проф.
[Signature] О. М. Мандрік
«*[Date]*» 2016 р.

АКТ

про впровадження у навчальний процес наукових результатів дисертаційної роботи асистента кафедри документознавства та інформаційної діяльності Мельника В.Д. «Інтелімедійна інформаційна система підтримки прийняття рішень в нафтогазовій справі»

Ми, що нижче підписалися, директор інституту інформаційних технологій, канд. техн. наук, доцент Чигур І.І., завідувач кафедри програмного забезпечення автоматизованих систем д-р. техн. наук, професор Юрчишин В.М., доцент кафедри Шекета В.І. (науковий керівник), склали цей акт у тому, що матеріали дисертації «Інтелімедійна інформаційна система підтримки прийняття рішень в нафтогазовій справі» Мельника В.Д. впроваджені у навчальний процес при вивченні дисципліни «Комп'ютерні системи штучного інтелекту», змістовний модуль ЗМ2, навчальний елемент НЕ2.4 «Комп'ютерні системи засновані на правилах. Інтелектуальне прийняття рішень в нафтогазовій предметній області на основі баз знань» для студентів спеціальності 8.05010201 – Комп'ютерні системи та мережі та дисципліни «Формальні та програмні засоби побудови новітніх інтелектуальних додатків та застосувань», змістовний модуль ЗМ2, навчальний елемент НЕ2.3 «Основи нечіткої логіки. Реалізація прийняття рішень в функціональності новітніх інтелектуальних додатків на основі баз нечітких знань» для студентів спеціальності 8.05010301 – Програмне забезпечення систем.

Директор інституту інформаційних технологій,
канд. техн. наук, доцент

[Signature] І.І. Чигур

Завідувач кафедри програмного
забезпечення автоматизованих систем,
д-р техн. наук, професор

[Signature] В.М. Юрчишин

Доцент кафедри програмного забезпечення
автоматизованих систем, науковий керівник

[Signature] В.І. Шекета

Асистент кафедри документознавства та
інформаційної діяльності

[Signature] В.Д. Мельник