

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

Полібіна Катерина Вячеславівна

УДК 004.89:004.4(043.3)

ДИСЕРТАЦІЯ
СЦЕНАРНО-ПРЕЦЕДЕНТІ МОДЕЛІ ПІДТРИМКИ ІНДИВІДУАЛЬНОЇ
ОСВІТНЬОЇ ТРАЄКТОРІЇ ПІДГОТОВКИ ФАХІВЦІВ З ІНЖЕНЕРІЇ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

122 Комп'ютерні науки

12 Комп'ютерні науки

Подається на здобуття наукового ступеня доктора філософії

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

Науковий керівник:

Козуб Наталія Олександрівна,

кандидат технічних наук, доцент

Хмельницький 2024

АНОТАЦІЯ

Полібіна К.В. Сценарно-прецеденті моделі підтримки індивідуальної освітньої траєкторії підготовки фахівців з інженерії програмного забезпечення. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 Комп'ютерні науки, Херсонський національний технічний університет, Міністерство освіти і науки України, Хмельницький, 2024.

Дисертаційна робота присвячена дослідженню моделей та методів розробки експертної інтелектуальної системи підтримки вибору індивідуальної освітньої траєкторії на основі ефективної організації та використання ретроспективних, поточних та експертних знань.

В роботі досліджено сучасний стан питань створення експертних інтелектуальних систем підтримки вибору індивідуальної освітньої траєкторії та можливих шляхів їх вирішення. Проведений аналіз останніх наукових досліджень дозволив визначити базові принципи побудови, функціональні можливості, пріоритетні напрямки практичного застосування експертних інтелектуальних систем в галузі освіти. Проаналізовано існуючі моделі та методи обробки даних в таких системах. Встановлено, що на поточний момент часу існує нагальна потреба в створенні сучасних експертних інтелектуальних систем підтримки прийняття рішення в слабоформалізованій освітньої галузі, які враховують інтелектуальний аналіз даних та попередній досвід людини, що приймає рішення, і забезпечують прийняття нею рішень в умовах невизначеності; які здатні ефективно вирішувати окрему частку питань, пов'язаних із процесами формування рішень при управлінні навчанням та зниженням впливу людського фактору на цей процес.

Запропоновано застосування в побудові експертній інтелектуальній системі підтримки рішення методів інтелектуальної обробки освітніх даних. З метою підвищення ефективності функціонування системи підтримки вибору індивідуальної освітньої траєкторії запропоновано зменшувати обсяг інформації, яка повинна бути

прийнята до обробки, шляхом експертних ухвалень. Запропонований метод класифікації даних надає можливість використання інтелектуального аналізу освітніх даних без надлишкового «занурення» людини, що приймає рішення, у деталі базових даних або проміжної аналітики.

За результатами проведеного аналізу було визначено концепцію створення експертних інтелектуальних систем підтримки прийняття рішень з використанням сценарно-прецедентного підходу. Цей підхід базується на ідеї того, що система може аналізувати сценарії проблемних ситуацій на основі вже наявних прецедентів, що виникають в процесі управління навчанням. Це дозволяє системі індивідуалізувати навчання, створюючи для кожного користувача свою унікальну траєкторію.

Виходячи з вище вказаного, запропоновано підхід до створення експертної інтелектуальної системи підтримки рішення на основі інтеграції методів інтелектуального аналізу освітніх даних та сценарно-прецедентного підходу, що дозволяє накопичувати, систематизувати, інтегрувати та ефективно використовувати ретроспективні і поточні дані та досвід експертів. Ця інтеграція сприяє поліпшенню процесу формування бази вдалих (підходящих) прецедентів із використанням модифікованого CBR циклу, що використовує експертну інформацію (тестові набори даних) для отримання прецедентів, підвищує релевантність отриманих рішень. Застосування сценарно-прецедентного підходу дозволяє перетворити поточну ситуацію в багатоваріантний сценарій рішення. Отримання широкого спектру варіантів рішень дозволяє виявити найбільш релевантні серед них та проаналізувати можливі наслідки запропонованих альтернативних варіантів. Це дозволяє системі порівняти їх та визначити найефективніший шлях розв'язання проблеми.

Беручи до уваги необхідність зниження впливу людського фактора на процеси ухвалення рішення, було запропоновано використання комбінованої методики формування функції вибору ЛПР, що ґрунтується на нелінійній згортці окремих критеріїв, заданих за допомогою лінгвістичних змінних (метод триплетів).

В якості технічних засобів розробки та реалізації програмного продукту були використані мова програмування C#, середовища програмування MS Visual Studio Enterprise 2017, бібліотеки пакету EPPlus – OfficeOpenXml, бібліотеки пакету Microsoft.Office.Interop.Word, який входить до Microsoft Office Primary Interop Assemblies (PIA) і забезпечує інтерфейс для взаємодії з Microsoft Word за допомогою COM-взаємодії, середовище MS Microsoft Excel.

Запропонована експертна інтелектуальна система підтримки вибору індивідуальної освітньої траєкторії складається з чотирьох підсистем: підсистеми завантаження, перетворення та зберігання даних; підсистеми аналізу та класифікації даних; підсистеми прийняття рішення; підсистеми користувацької візуалізації.

Тестування експертної інтелектуальної системи підтримки вибору індивідуальної освітньої траєкторії «Choice» здійснювалося в Новокаховському приладобудівному фаховому коледжі. Практична апробація створеного прототипу дозволила підвищити якість прийняття рішення індивідуалізованого вибору наукового керівника та теми дипломного проєкта студентів з урахуванням більш точного зіставлення їх академічних та професійних переваг з компетенціями та експертизою потенційних наукових керівників; запропоновано адаптивний вибір місця практики, який дозволяє студентам знаходити оптимальні варіанти розвитку своїх професійних навичок; здійснено інтеграцію даних та зворотного зв'язку, який забезпечує студентів актуальною інформацією, дозволяючи їм ухвалювати рекомендації експертної системи; проаналізовано вимоги ринку праці в галузі ІТ, який стрімко змінюється, що забезпечує студентам можливість вибору таких керівників та місць практики, які максимально відповідають сучасним вимогам ІТ індустрії.

Ключові слова: експертні системи, система підтримки прийняття рішень, індивідуальна освітня траєкторія, інтелектуальний аналіз, сценарно-прецедентний підхід, CBR-цикл.

ABSTRACT

Polibina K.V. Scenario-Precedent Models in Individual Educational Pathway Support for Software Engineering Specialist Training. – Qualifying research paper as a manuscript.

Dissertation for the degree of Doctor of Philosophy on a specialty 122 Computer Science, Kherson National Technical University, Ministry of Education and Science of Ukraine, Khmelnytsky, 2024.

The dissertation focuses on the study of models and methods of expert intellectual system development for support of choice of individual educational pathbased on effective arrangement and use of retrospective, current and expert knowledge.

The paper investigates the problems of building expert intellectual systems to support the choice of individual educational pathway, and possible ways to solve these problems. The analysis of recent research shows the basic principles of construction, functional capabilities, and priority directions of practical application of expert intellectual systems in education. The paper analyses the existing models and methods of data processing in such systems. It has shown that at the current moment there is an urgent need to create modern expert intellectual systems of decision-making support in the weakly formalized field of education. They are to take into account the intellectual analysis of data and previous experience of a decision-making person and ensure his decision in conditions of uncertainty. In addition, they are able to address effectively a separate part of the issues related to the decision-making processes in management of training, and reduce the influence of the human factor.

The research proposes to apply the methods of intellectual processing of educational data in building an expert intellectual decision-making support system. In order to increase the efficiency of the support system in choosing an individual educational path, it is proposed to reduce the amount of information under processing by means of expert decisions. The proposed method of data classification provides an opportunity to use

intelligent analysis of educational data without excessive «immersion» of a decision maker in basic data or intermediate analytics.

Basing on the results of the analysis done, the research defines the concept of building expert intelligent decision support systems using the scenario-precedent approach. This approach is based on the idea that the system can analyze scenarios of problem situations basing on already available precedents arising in the process of training management. This allows the system to individualize learning process by creating a unique pathway for each user.

Based on the above mentioned, the research proposes to build an expert intelligent decision support system based on the integration of methods of educational data mining and scenario-precedent approach. The approach allows accumulating, systematizing, integrating and effectively using retrospective, current data and expert experience. This integration enhances the process of building a precedent base using the modified CBR cycle. The modified CBR cycle, which uses expert information (test datasets) to obtain precedents, improves the relevance of the resulting precedent-based decisions by building a base of successful (suitable) precedents during the execution of the CBR cycle.

The proposed algorithm for decision-making based on the Jaccard-Tanimoto coefficient metric allows implementing an effective method for assessing the similarity of the current precedent with each precedent in the database based on their analogy with more similar cases from the past. Taking into account the need to reduce the influence of the human factor on the decision-making processes, it was proposed to use a combined technique for developing a decision-maker's (DM) choice function, which is based on the nonlinear convolution of individual criteria specified using linguistic variables (triplet method).

For the development and use of the software product, the following were applied as technical means: programming language C#, MS Visual Studio Enterprise 2017 programming environment, EPPlus package libraries – OfficeOpenXml, Microsoft.Office.Interop.Word package libraries, which is part of Microsoft Office

Primary Interop Assemblies (PIA) and provides an interface to interact with Microsoft Word using COM interactions, MS Microsoft Excel environment.

The proposed expert intellectual system to support the choice of individual educational pathway consists of four subsystems: subsystems of data loading, transformation and storage; subsystems of data analysis and classification; subsystem of decision-making; subsystem of user visualization.

The CHOICE expert intellectual system to support the building of individual educational pathway test took place in Nova Kakhovka Instrument-Making Vocational College. Approbation of the CHOICE prototype allowed to:

- improve the quality of decision-making process in individualized choice of supervisor and topic of students' final projects taking into account more accurate comparison of their academic and professional advantages with competences and expertise of potential supervisors;
- offer an adaptive choice of internship that allows students to find the best options for developing their professional skills;
- implement data integration and feedback that provides students with up-to-date information, enabling them to accept the expert system's recommendations; integration of data and feedback is implemented, which provides the students with the possibility to choose the best educational pathway;
- analyse the requirements of the IT job market, which is rapidly changing, providing students with a choice of supervisors and therefore appropriate placements.

Key words: expert systems, decision-making support system, individual educational pathway, intellectual analysis, scenario-precedent approach, CBR-cycle.

Список опублікованих праць за темою дисертації
Публікації в наукових фахових виданнях України:

1. Полібіна К.В. Дослідження моделей, методів та інформаційних технологій оцінки кваліфікації програміста. *Проблеми інформаційних технологій*. 2020. № 27. С. 77-88.

URL: <https://journals.kntu.net.ua/index.php/pit/article/view/706>

DOI: <https://doi.org/10.35546/2313-0687.2020.27.78-88>

2. Полібіна К.В. Ефективність дистанційних освітніх технологій в умовах пандемії. *Інфокомунікаційні та комп'ютерні технології. Вісник Університету «Україна»*. Київ. 2021. № 2 (02), С.280-293.

URL: <https://visn-icct.uu.edu.ua/index.php/icct/article/view/59/46>

DOI: <https://doi.org/10.36994/2788-5518-2021-02-02-21>

3. Polibina K. Intelligent training process management. *Вісник Хмельницького національного університету. Серія: технічні науки*. 2024. №1. С. 426-432.

URL: <https://heraldts.khmnu.edu.ua/index.php/heraldts/article/view/97/79>

DOI: <https://doi.org/10.31891/2307-5732-2024-331-65>

**Публікації в наукових виданнях інших держав та у виданнях України,
що включені до міжнародних наукометричних баз SCOPUS, WoS та ін..**

1. Polibina K. Model of Supporting an Individual Training Pathway for Software Engineers. *Date Added to IEEE Xplore*: 06 December 2021.

Матеріали і тези Міжнародних наукових та науково-практичних конференцій

IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT). Date of Conference: 21-25 Sept. 2021. Lviv, Ukraine.

URL: <https://ieeexplore.ieee.org/document/9628900>

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ	11
ВСТУП	12
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМ ПОБУДОВИ ЕКСПЕРТНИХ СИСТЕМ ПІДТРИМКИ ВИБОРУ ІНДИВІДУАЛЬНИХ ОСВІТНІХ ТРАЄКТОРІЙ	20
1.1 Сучасні тенденції використання експертних систем	20
1.2 Застосування експертних систем в освіті	29
1.3 Експертні системи підтримки індивідуальної освітньої траєкторії	35
1.4 Структура навчального процесу із застосуванням індивідуальної траєкторії навчання	40
1.5 Застосовність експертних систем у навчальному процесі	48
1.6 Постановка задачі дослідження	55
1.7 Висновки до першого розділу.....	56
РОЗДІЛ 2 МЕТОДИ ТА МОДЕЛІ ПОБУДОВИ ЕКСПЕРТНОЇ СИСТЕМИ ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ	58
2.1 Навчальний заклад, як соціально-економічна система	59
2.2 Інтелектуальні системи підтримки прийняття рішень	61
2.3 Експертні системи інтелектуальної підтримки прийняття рішення	65
2.3.1 ETL процеси	67
2.3.2 Інтелектуальний аналіз даних	73
2.3.3 Сценарно-прецедентний підхід	76
2.3.4 Методи зниження впливу факторів суб'єктивності та невизначеності	85
2.3.5 Архітектура ЕС ІППР	89
2.4 Висновки до другого розділу.....	92
РОЗДІЛ 3 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПІДТРИМКИ ВИБОРУ ІНДИВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ	93
3.1 Архітектура ЕС ІППР вибору наукового керівника та теми ДП	93

3.2 Застосування ETL процесів автоматизації завантаження освітніх даних	94
3.3 Інтелектуальний аналіз освітніх даних як складова CBR циклу	97
3.4 Методи та метрики визначення схожості прецедентів	106
3.5 Приклад використання прототипу ЕС ІППР	110
3.6 Висновки до третього розділу.....	118
РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПІДТРИМКИ ВИБОРУ ІНДИВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ	120
4.1 Аналіз процесу обрання теми дипломного проєкта на відділенні Інженерія програмного забезпечення в НКПФК	120
4.2 Вимоги до прототипу ЕС ІППР «Choice»	127
4.2.1 Підсистема автоматизації завантаження освітніх даних	131
4.2.2 Підсистема аналітичного аналізу освітніх даних	134
4.2.3 Підсистема підтримки прийняття рішення ЛПР	138
4.2.4 Підсистему візуалізації прийнятого рішення ЛПР	142
4.3 Вплив впровадження ЕС ІППР «Choice» в навчальний процес	145
4.4 Висновки до четвертого розділу.....	152
ВИСНОВКИ	153
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	155
ДОДАТОК А ЛІСТИНГ КОДУ ПРОГРАМИ	168
ДОДАТОК Б ОТРИМАНІ ВИХІДНІ ДАНІ	190
ДОДАТОК В АКТИ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ	193

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

ВПД – види професійної діяльності;

БД – база даних;

БЗ – база знань;

БП – база прецедентів;

БПР – блок прийняття рішення;

БПС – база параметрів поточної ситуації;

ДП – дипломне проєктування;

ЕС – експертна система;

ЕС ІППР – експертна система інтелектуальної підтримки прийняття рішень;

ІМС – інтелектуальна мультиагентна система;

ІОТ – індивідуальна освітня траєкторія;

ІР – Інтернет речей;

ІС – інформаційна система;

ІСППР – інтелектуальна система підтримки прийняття рішень;

ЛПР – людина, що приймає рішення;

НЗ – навчальний заклад;

НП – навчальний план;

ТВР – технологія віртуальної реальності;

ІІІ – штучний інтелект;

СВР – Case-based reasoning;

ІТ – інформаційні технології;

LMS – системи управління навчанням;

NLP (Natural Language Processing) – обробка природної мови.

ВСТУП

Процес пошуку нових рішень у сфері національної системи освіти загалом та ІТ-освіти зокрема обумовлений прагненням суспільства адаптуватися, по-перше, до нових умов цифровізації всіх сфер людської діяльності, по-друге, відповідати новим реаліям суспільного розвитку та євроінтеграційного поступу України, а по-третє – сприяти формуванню особистості професіонала, що є однією із рушійних сил розвитку суспільства.

Освітній простір є одним із інноваційних майданчиків, на якому відбувається зіткнення двох подібних та водночас взаємозалежних процесів – модернізації та трансформації різних освітніх моделей.

Різносторонній характер модернізації передбачає насамперед інтенсифікацію розвитку освітньої сфери за допомогою інновацій та реформ для її переходу на новий якісний рівень, який орієнтований на перспективу, оскільки, по-перше, здебільшого фахівці з вищою освітою є ініціаторами інновацій, а, по-друге, все помітнішим стає вплив рівня та якості освіти на підвищення матеріального і культурного рівня населення [1].

Трансформація, своєю чергою, спирається, передусім, на історичну тенденцію - мета освіти стала набувати прагматичного характеру: замість формування гармонійно розвиненої особистості ставиться завдання підготовки компетентного, соціально-орієнтованого спеціаліста, пристосованого до запитів ринку праці [2], що потребує не стільки радикальних інновацій і реформ, скільки переорієнтації існуючих освітніх моделей.

З одного боку, в умовах цифрової реальності можна говорити про те, що за останні десять років в Україні певною мірою відбувається модернізація освітнього простору, що забезпечує інформаційну грамотність населення, комп'ютеризацію виробництва, розширення освітніх послуг за допомогою он-лайн навчання, розширення академічного та освітнього простору за рахунок цифрових технологій, і так далі. Однак, з іншого боку, не можна сказати, що в умовах цифрової реальності в

Україні вдалося суттєво трансформувати національну систему освіти. Мова йдеться не так про зміну рамок освітнього простору, якими є Болонський процес як стандартизація системи освіти, мова йде про трансформацію ціннісного фундаменту освітньої моделі.

Шляхи реалізації вище вказаних тенденцій закладено в Стратегії розвитку вищої освіти в Україні на 2022-2032 роки та Операційному плані її реалізації у 2022-2024 роках (Розпорядження КМУ від 23 лютого 2022 р. № 286-р) [3]. У стратегії визначено цілі, напрямки та завдання модернізації національної системи освіти, що будуть сформовані до 2032 року.

Актуальність дослідження. Стрімкий розвиток інформаційних технологій та динамічна зміна ринку праці в сучасних умовах привели до появи нових професій та ускладнення вимог до існуючих, що викликало необхідність гнучкого та швидкого реагування. У зв'язку з цим, впровадження індивідуальних освітніх траєкторій (ІОТ) в освіті, являє собою стратегію, засновану на персоналізованій, гнучкій, керованій системі формування необхідних компетентностей у тих, хто навчається. ІОТ дозволяють оптимізувати навчальний процес, врахувати індивідуальні переваги та особливості, реалізувати можливість неодноразової зміни напряму підготовки без втрати термінів навчання, отримати спеціалізовані знання. Крім цього, в ІОТ закладено значний резерв підвищення якості та ефективності навчального процесу, оскільки вона базується на сучасних освітніх технологіях, зокрема методах штучного інтелекту (ШІ). ІОТ характерна відкритість і мобільність, орієнтація на дослідницьку і проектну діяльність, наявність зворотного зв'язку на кожному етапі навчання, а також накопичувальна система оцінювання досягнень.

Така можливість у системі освіти була завжди, але послідовна зміна освітніх стандартів протягом короткого проміжку часу та самостійне формування освітніми установами переліку дисциплін навчального плану призвели до певних складнощів та ряду протиріч між необхідністю:

- впровадження систем підтримки ІОТ та відсутністю конкретних моделей їх створення;
- супроводу функціонування систем підтримки ІОТ та відсутністю фахівців для цього;
- обробки інформації людиною, яка приймає рішення (ЛПР) при виборі ІОТ та необхідністю зниження впливу людського фактору на процеси прийняття рішення.

Отже, створення системи підтримки вибору індивідуальної освітньої траєкторії є актуальною метою дослідження. Для забезпечення персоналізації та інтелектуалізації підтримки вибору індивідуальної освітньої траєкторії пропонується сумісне використання експертних систем та інтелектуальних систем підтримки рішення (ЕС ІППР).

Аналіз існуючих робіт, які присвячені теорії і практиці створення експертних систем в освіті [6,7,10-12,14,15,20,21,37,50,63,87,96,98], показує, що на даний час питання створення і практичного використання таких систем опрацьовані недостатньо і викликають значний інтерес серед дослідників.

Підхід до створення експертної інтелектуальної системи підтримки рішення на основі інтеграції методів інтелектуального аналізу освітніх даних та сценарно-прецедентного підходу відображає сучасний підхід до розвитку технологій у сфері освіти. Цей підхід базується на здатності системи аналізувати інформацію з різних джерел, включаючи ретроспективні дані та досвід експертів, а також поточні дані, що надходять у реальному часі.

Інтеграція методів інтелектуального аналізу освітніх даних дозволяє системі автоматично обробляти та аналізувати великі обсяги даних, що надає можливість отримувати інсайти та виявляти закономірності. Такий аналіз може допомогти ідентифікувати успішні підходи та тенденції в навчальному процесі, а також виявляти можливі проблеми або слабкі місця.

Сценарно-прецедентний підхід дозволяє системі моделювати різноманітні навчальні ситуації та враховувати історичні прецеденти, що виникли раніше. Це дозволяє системі створювати індивідуалізовані стратегії та рекомендації для користувачів, враховуючи їх поточний контекст та потреби.

Інтеграція цих двох підходів дозволяє системі ефективно використовувати як ретроспективні дані та досвід експертів, так і поточні дані для прийняття релевантних рішень у навчальному процесі. Це робить прийняте рішення більш обґрунтованим і підготовленим для подальшого впровадження в освітню практику.

Вивчений досвід розробки ЕС ІППР в освітній галузі дозволяє зробити висновок, що для ефективного практичного використання таких систем необхідне виконання низки важливих вимог, обумовлених, з одного боку, особливостями функціонування ЕС ІППР, а з іншого – умовами прийняття рішень в слабо структурованій предметній області, якою є освітня галузь.

Зв'язок роботи з науковими програмами, планами, темами. Наукові дослідження виконувались в рамках тематичного плану науково-дослідних робіт відповідно тем: «Інтелектуальні технології гнучкого навчання за індивідуальними траєкторіями», «Розвиток моделей, методів і технологій сценарно-прецедентного управління динамічними об'єктами та системами» кафедри ПЗіТ Херсонського національного технічного університету.

Мета та завдання дослідження. Мета дослідження полягає в аналізі та дослідженні моделей та методів розробки експертної інтелектуальної системи підтримки вибору індивідуальної освітньої траєкторії на основі ефективної організації та використання ретроспективних, поточних та експертних знань.

Для досягнення поставленої мети було сформульовано такі завдання:

1. Комплексно дослідити проблеми побудови експертних інтелектуальних систем підтримки рішень в освіті та можливі шляхи їх вирішення.
2. Розробити математичні моделі та методи підтримки вибору ІОТ на основі сценарно-прецедентного підходу.

3. Розробити інформаційну технологію та процедури підтримки вибору ІОТ на основі сценарно-прецедентного підходу.

4. Створити експертну інтелектуальну систему підтримки вибору ІОТ.

Об'єктом дослідження є процес підтримки вибору ЛПР індивідуальної освітньої траєкторії підготовки фахівців інженерії програмного забезпечення.

Предметом дослідження є методи, моделі та програмні засоби інтелектуалізації та персоналізації підтримки вибору ЛПР індивідуальної траєкторії підготовки фахівців інженерії програмного забезпечення.

Методи дослідження. Поставлені завдання вирішуються з використанням анкетування для збору інформації або думок досліджуваної групи за допомогою структурованих або неструктурованих опитувальників; інтелектуального аналізу даних для виявлення прихованих закономірностей, залежностей та статистичних висновків; дискретної математики, зокрема, теорії множин, для обрахування коефіцієнта Жаккара-Танімото; методи штучного інтелекту та елементи теорії експертних систем для розробки технології вибору рішення; методів правдоподібних міркувань на основі прецедентів для вивчення схожих ситуацій з минулого для прийняття рішень у нових ситуаціях; теорії програмування для вивчення структур, алгоритмів та методів програмування для програмної реалізації системи.

Наукова новизна дослідження полягає в наступному:

вперше:

- розроблено інформаційну технологію підтримки індивідуалізованого вибору ЛПР наукового керівника дипломного проєкта студентів з урахуванням більш точного зіставлення їх академічних та професійних переваг з компетенціями та експертизою потенційних наукових керівників;
- запропоновано адаптивний вибір ЛПР місця практики, який дозволяє студентам знаходити оптимальні варіанти розвитку своїх професійних навичок;
- запропоновано модифікований CBR цикл, що використовує експертну інформацію (тестові набори даних) для отримання прецедентів. Даний метод

підвищує якість вирішення задач на основі міркування за прецедентами за рахунок формування бази вдалих (підходящих) прецедентів у процесі виконання CBR циклу;

- запропоновано використання комбінованої методики формування функції вибору ЛПР, що ґрунтується на методі триплетів, для отримання ранжованого списку релевантних рішень та зниження впливу фактору невизначеності на ЛПР.

отримали подальший розвиток:

- застосування сценарно-прецедентного підходу дозволяє перетворити поточну ситуацію в багатоваріантний сценарій рішення. Отримання широкого спектру варіантів рішень дозволяє виявити найбільш релевантні серед них та проаналізувати можливі наслідки запропонованих альтернативних варіантів. Це дозволяє системі порівняти їх та визначити найефективніший шлях розв’язання проблеми;

- формування індивідуальних траєкторій навчання на основі сценарно-прецедентного компетентнісного підходу виявляється корисним не лише для оптимізації навчального процесу, але й для підтримки прийнятті рішень ЛПР в умовах невизначеності.

Практичне значення одержаних результатів. На основі розроблених моделей, методів та алгоритмів створено прототип ЕС ІППР «Choice», який дозволяє вирішити низку важливих прикладних завдань в галузі застосування інформаційних технологій в освіті – підвищення ефективності, оптимізація процедури прийняття рішень та інформаційної підтримки ЛПР в слабоформалізованих навчальних процесах та зниження впливу людського фактору на процеси вибору рішення.

Практичне застосування створеного прототипу ЕС ІППР «Choice» дозволяє підвищити якість прийняття рішення в закладах освіти. Одержані результати наукових досліджень впроваджені у Новокаховському приладобудівному фаховому коледжі для управління процесом призначення наукових керівників та тем дипломного проектування студентам-випускникам (Акт впровадження від 01.02.2024 р.).

Наукові результати та розробки використовуються в освітньому процесі на відділенні «Економіка та програмування» у ВСП «Новокаховський політехнічний фаховий коледж Національного університету «Одеська політехніка» (Акт про використання від 12.03.2023) при викладанні дисциплін: «Інформаційно-аналітичні системи», «Системи ІППР», «Бази даних», «Об'єктно-орієнтоване програмування». А також на відділенні «Комп'ютерна та програмна інженерія» у ВСП «Херсонський політехнічний фаховий коледж Національного університету «Одеська політехніка» при викладанні навчальних дисциплін «Конструювання програмного забезпечення», «Бази даних», «Людино-машинний інтерфейс», «Основи науково-дослідницької роботи» (Акт використання від 20.03.2023 р.). Акти впровадження наведені в додатку В.

Апробація результатів дисертації доповідались та обговорювались на науковій конференції IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT). Date of Conference: 21-25 Sept. 2021. Lviv, Ukraine.

Особистий внесок здобувача полягає в тому, що всі теоретичні та практичні результати дисертаційного дослідження одержані автором самостійно. Автором проведено аналіз існуючого рівня розвитку експертних систем та інтелектуальних систем підтримки рішення в освіті та технологій, що використовуються при створенні таких систем. Створено математичні моделі та розроблено інформаційну технологію підтримки вибору ЛПР індивідуальної освітньої траєкторії з використанням методів математичного моделювання, штучного інтелекту, сценарно-прецедентного компетентнісного підходу.

Публікації

Основні результати дослідження опубліковані у 4 наукових працях (всі одноосібні), у тому числі 1 стаття у науковому виданні, що входить до наукометричної бази Scopus [123], 3 – у наукових фахових виданнях,

рекомендованих МОН України для публікації наукових досліджень [85,128] та 1 публікація у матеріалах конференцій [123].

Структура та обсяг дисертації

Дисертація складається зі вступу, чотирьох розділів, висновків за кожним розділом, висновку, бібліографічного списку зі 128 найменувань, 7 додатків. Обсяг дисертації складе 195 сторінок, з яких 154 сторінки основного тексту, в тому числі 40 рисунків і 4 таблиць.

РОЗДІЛ 1

АНАЛІЗ ПРОБЛЕМ ПОБУДОВИ ЕКСПЕРТНИХ СИСТЕМ ПІДТРИМКИ ВИБОРУ ІНДИВІДУАЛЬНИХ ОСВІТНІХ ТРАЄКТОРІЙ

В розділі розглядається історична еволюція штучного інтелекту (ШІ) та його практичне застосування, з акцентом на експертних системах, які стали значною сферою інтересу протягом останніх двох десятиліть. Також розглядаються аспекти, які пов'язані з розробкою та застосуванням експертних систем у виборі індивідуальних освітніх траєкторій студента. Розглядається використання експертних систем в освіті як частини ширшої сфери штучного інтелекту, їх можливості для автоматизації освітніх процесів та підтримки прийняття рішень в навчанні, зокрема через персоналізацію навчального процесу. Висвітлюється інтеграція експертних систем в освіту завдяки їхній здатності зберігати, обробляти та генерувати знання.

1.1 Сучасні тенденції використання експертних систем

Сьогодні різні галузі науки і техніки виробляють і розширюють знання в предметних галузях з величезною швидкістю, чим провокують дедалі більший інтерес до програм і систем, які дають змогу не тільки зберігати і переробляти готові дані, а й виробляти нові знання на основі вже наявної інформації. Як наслідок, царина штучного інтелекту стає вельми привабливою для розроблення та впровадження наукових проєктів, у рамках яких особливу перевагу віддають експертним системам, які в останні двадцять років стали актуальним напрямом.

Експертні системи за своєю суттю є цілим програмним обчислювальним комплексом, що містить великий обсяг знань експертів у деякій конкретній галузі та в межах якої система здатна приймати експертні вірні рішення, видаючи чіткі відомості на запит користувача [4].

Експертні системи вирішують конкретні реальні проблеми, з якими у повсякденному житті стикаються в рамках своєї професійної діяльності люди-експерти. Знання в системі організовані як деяка сукупність чітких правил, які дають змогу робити висновки на основі введення вхідних даних або припущень.

Експертні системи, на відміну від інших орієнтованих на практику програмних комплексів штучного інтелекту, мають більш масове застосування, оскільки кожна експертна система дає змогу вирішити необмежену кількість складних для людини завдань із відповідної предметної області. Тому тема розробки експертних систем завжди привертала значну увагу наукової спільноти [5-7].

В історії становлення ІІІ, як наукової галузі, періоди очікувань швидких успіхів регулярно змінювалися періодами падінь, коли отримані результати досліджень починали здаватися не об'єктивними, а інвестиції в дослідження ІІІ різко зменшувалися. Однак поява у 1970-х роках експертних систем та систем, що базуються на знаннях, однозначно пов'язуються з успіхом у даній науковій області, коли методи ІІІ стали використовуватися у реальному світі повсякдення.

З часу впровадження перших так званих класичних систем, що базуються на моделі представлення знань у вигляді міркування на основі правил (rule-based reasoning) [8-13] пройшло понад 40 років, але архітектура подібних систем мало змінилася. Під час впровадження таких систем розробники стикалися з проблемами, такими як складність здобуття знань у експертів, вимога високої кваліфікації користувача, складність обробки великих обсягів інформації та проблеми супроводження. Для вирішення цих проблем проводилися дослідження з метою знаходження кращих методів та інструментів, в результаті чого були розроблені різноманітні оболонки та середовища, поліпшена методологія розробки [14-17], розроблені мови моделювання знань, онтології [18-19], методи та засоби для систем підтримки прийняття рішень [20-22].

Мета вище вказаних досліджень з експертних систем полягала в розробленні програм, які під час розв'язання завдань, важких для експерта-людини, отримувала

би результати, що не поступаються за якістю та ефективністю рішенням, одержуваним експертом. У більшості випадків експертні системи розв'язують завдання, які важко формалізувати, або завдання, що не мають алгоритмічного рішення.

На основі розглянутих робіт і визначень запропонованих дослідниками, можна сформулювати поняття експертної системи.

Експертні системи - це прикладні програми галузі штучного інтелекту, в яких база знань являє собою формалізовані знання висококваліфікованих фахівців у будь-якій вузькій предметній області. Експертні системи призначені для заміни експертів під час вирішення професійно складних завдань через недостатню кількості фахівців, недостатньої оперативності під час ухвалення рішень експертами або через присутність негативних чинників, які можуть впливати на роботу фахівців.

Зазвичай експертні системи розглядають з погляду їх застосування у двох аспектах: для розв'язання конкретних завдань та в якій галузі діяльності вони можуть бути використані.

1. Діагностичні системи: Використовуються для діагностики проблем або станів, наприклад, у медицині або технічному обслуговуванні.

2. Прогностичні системи: Ці системи використовуються для прогнозування подій або результатів на основі наявних даних, наприклад, у метеорології або фінансах.

3. Консультативні системи: Надають поради або рекомендації, які засновані на експертних знаннях, наприклад, в правовій або освітній сфері.

4. Системи підтримки прийняття рішень: Ці системи допомагають в аналізі складних даних та варіантів для підтримки процесу прийняття рішень, особливо в бізнесі та управлінні.

5. Навчальні системи: Використовуються для навчання або тренування користувачів у певних областях знань, використовуючи експертні знання для створення навчального контенту.

6. Системи обробки природної мови: Спеціалізуються на розумінні та генерації природної мови, можуть використовуватися для перекладу, автоматичної відповіді на питання тощо.

7. Системи планування та оптимізації: Використовують експертні знання для планування та оптимізації процесів або ресурсів, наприклад, в логістиці або виробництві.

Кожна з цих категорій може мати багато підкатегорій і спеціалізацію, і вони часто перетинаються одна з одною в реальних застосуваннях.

Все вище вказане класифіковано та представлено нижче (рис 1.1).

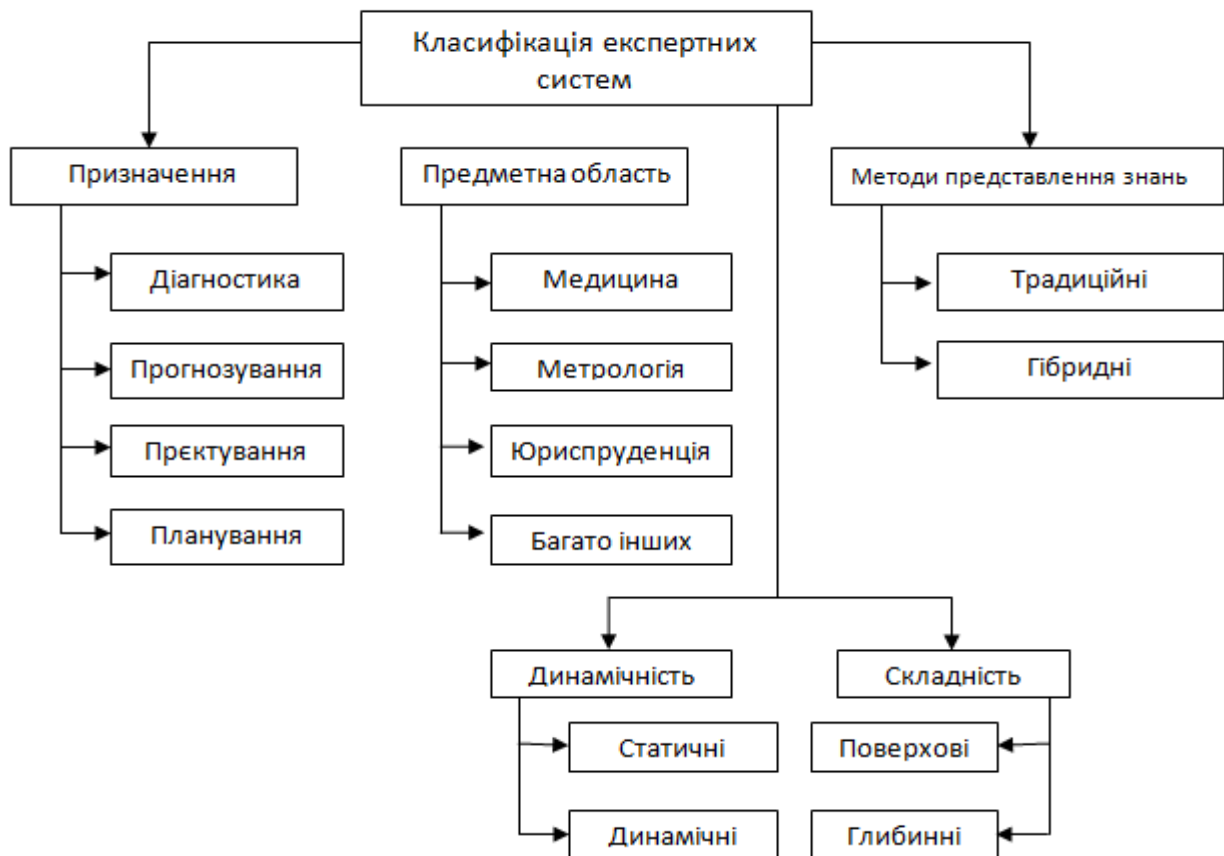


Рисунок 1.1 – Класифікація експертних систем за областями застосування

В експертній системі відомий алгоритм обробки знань, а не алгоритм розв’язання задачі. Тому застосування алгоритму обробки знань може призвести до

отримання такого результату під час розв’язання конкретної задачі, який не був передбачений. Більше того, алгоритм опрацювання знань заздалегідь невідомий і будується під час розв’язання задачі на підставі правил.

Основними відмінностями експертних від інших програмних продуктів є використання не тільки даних, а й знань та спеціального механізму виведення рішень і нових знань на основі наявних. Знання в експертній системі подаються в такій формі, яка може бути легко оброблена на ЕОМ.

Як було вказано вище, в якості знань в автоматизованій системі використовують експертні знання фахівців у цій галузі. Користувач передає на експертизу інформацію й отримує як результат експертне судження про властивість об’єкта (рис.1.2).

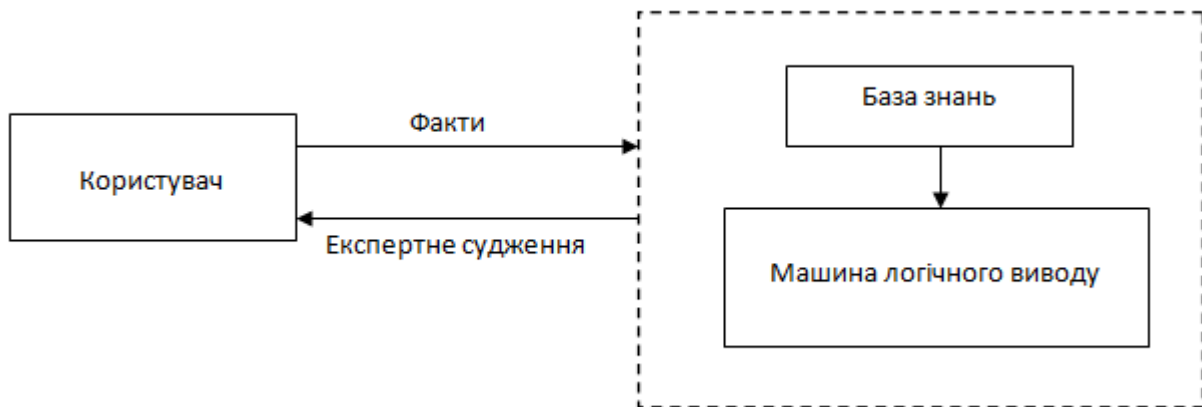


Рисунок 1.2 – Принцип роботи експертної системи

За своєю структурою система поділяється на два основні компоненти: базу знань і машину логічного виведення. База знань містить формалізовані знання фахівців, на підставі яких машина логічного висновку формує судження у вигляді оцінки.

Незалежно від мети розроблення експертної системи, у базі її знань завжди присутні три типи експертних знань, якими вона оперує і на основі яких приймає рішення:

1. Робочі знання, які використовуються для вирішення конкретного завдання або проведення консультації.

2. Структуровані динамічні знання про предметну область, які можуть змінюватися з часом, а точніше оновлюватися.

3. Структуровані статичні знання, незмінні з часом експертні знання про предметну область.

Особливості експертних систем, що відрізняють їх від звичайних програм, полягають у тому, що вони повинні мати такі якості:

1. Компетентність. А саме:

- система повинна за рівнем прийнятих рішень бути наближена до дій і міркувань експерта, тобто в конкретній предметній царині мати той самий рівень професіоналізму, що й експерти-люди;

- здійснювати своєчасні, чіткі та оперативні дії, тобто застосовувати знання ефективно і швидко, уникаючи зайвих обчислень;

- мати адекватну здатність лише поступово знижувати якість роботи в міру наближення до меж діапазону компетентності або допустимої надійності даних.

2. Можливість інтерпретації міркувань у символному вигляді:

- спочатку необхідні знання представляти в символному вигляді;

- здатність переформулювати символні знання.

3. Глибина пізнань:

- працювати в предметній області, незалежно від рівня складності завдань, які перед ними ставляться;

- за необхідності використовувати у своїй роботі для прийняття експертного вірного рішення складні правила.

4. Самосвідомість, тобто досліджувати свої міркування, пояснювати свої дії.

Таким чином, на основі вище розглянутих наукових джерел [4-22], можна говорити про те, що експертні системи об'єднують у собі такі передові методи штучного інтелекту, як:

1. Бази знань: Центральна частина експертної системи, що містить спеціалізовані знання та правила, які використовуються для імітації експертного рішення.

2. Машинне навчання: Використовує алгоритми для аналізу та вивчення з великих наборів даних, дозволяючи системі адаптуватися та покращуватися з часом.

3. Обробка природної мови (NLP): Дозволяє системам розуміти, інтерпретувати та генерувати людську мову, що є ключовим для інтерактивних систем, таких як чат-боти.

4. Вивід у режимі реального часу: Методи, що дозволяють системі швидко обробляти інформацію та надавати рекомендації або рішення у режимі реального часу.

5. Нечітка логіка: Використовується для обробки неоднозначної або неповної інформації, що дозволяє системі робити висновки в умовах невизначеності.

6. Експертні системи, засновані на правилах: Використовують набори правил для виведення висновків або прийняття рішень.

7. Системи, засновані на випадках (case-based reasoning): Рішення приймаються на основі аналізу попередніх подібних ситуацій або «випадків».

8. Онтології та семантичні веб-технології: Для структурування та інтерпретації знань у вигляді, зрозумілому для машин.

Ці методи можуть бути використані окремо або в комбінації для створення потужних експертних систем, які ефективно імітують людську експертну діяльність у різних областях.

Можна стверджувати, що саме експертні системи дали змогу отримати дуже великий комерційний ефект від застосування таких потужних методів.

Сьогодні на ринку продуктів галузі штучного інтелекту існують декілька добре відомих експертних систем, які використовуються в різних галузях.

IBM Watson. Один з найбільш відомих прикладів, Watson використовується в різноманітних областях, включаючи охорону здоров'я, фінанси та обслуговування клієнтів. Він здатний аналізувати великі обсяги даних і надавати рішення, використовуючи обробку природної мови та машинне навчання [23].

Google DeepMind відома своєю системою AlphaGo. DeepMind розробляє передові алгоритми глибокого навчання для розв'язання складних завдань у різних областях, включаючи охорону здоров'я та наукові дослідження [24].

Microsoft Azure AI - це платформа, яка надає різноманітні інструменти та сервіси для створення експертних систем, включаючи машинне навчання та ботів [25].

OpenAI, який є розробником GPT (Generative Pre-trained Transformer) серії моделей, що здатні генерувати текст, переклади, відповіді на запитання та інше, використовуючи сучасні методи обробки природної мови [26].

Palantir Technologies пропонує продукти для аналізу великих даних, які використовуються урядовими організаціями та корпораціями для аналізу, інтеграції та візуалізації даних з різних джерел [27].

Штучний інтелект, інтегрований в Salesforce Cloud, надає передбачувальний аналіз, рекомендації та автоматизацію для підвищення продуктивності у сфері продажів, маркетингу та обслуговування клієнтів [28].

Ці системи демонструють широкий спектр застосувань штучного інтелекту від здоров'я та фінансів до обслуговування клієнтів та аналізу даних, показуючи, наскільки глибоко ШІ інтегрований у сучасний бізнес та повсякденне життя.

Але, крім позитивних особливостей експертних систем, існують і недоліки, про які досить рідко можна отримати інформацію. Насправді, є невеликий перелік висновків і суджень про експертні системи, які було виявлено в результаті спостережень за практичною реалізацією таких систем. Більшість систем такого

виду не цілком придатні для застосування кінцевим користувачем. Тобто, якщо користувач не є впевненим користувачем ПК з необхідними навичками роботи з такими системами, то при взаємодії «користувач-система» можуть виникнути серйозні труднощі.

Запитально-відповідальний режим роботи, який, як правило, дуже часто реалізується в подібних системах, на практиці значно уповільнює роботу системи в цілому та отримання кінцевого рішення.

Актуальним залишається питання того, як привести знання, отримані від експерта, до вигляду, що забезпечує їх ефективну машинну реалізацію.

Також можна сказати, що експертні системи мало ефективні, а іноді й зовсім незастосовні у великих предметних областях. Їх використання обмежується вузькими предметними галузями, в яких експерт може прийняти рішення за час від кількох хвилин до кількох годин.

У тих галузях, де відсутні експерти, розробка і застосування експертної системи виявляється неможливим. Тому має сенс розробляти системи тільки для вирішення когнітивних завдань. Наприклад, їзда на велосипеді не може бути предметною областю для їх розроблення, однак такі системи можна використовувати під час формування, наприклад, футбольних команд.

Реальний фахівець-експерт під час розв'язання професійних завдань зазвичай звертається до своєї професійної інтуїції, знань, заснованих на практичному досвіді або на логіці розсудливості, якщо відсутні формальні методи вирішення.

Таким чином, застосування експертних систем найефективніші під час використання у вузьких предметних областях, або в тому разі, коли робота експерта займає досить тривалу кількість часу, або коли наявність необхідної кількості експертів з якихось причин неможлива.

Виходячи з вище вказаного, можна зробити висновки, що для будь-якої експертної системи характерно:

- обмеження застосування певною сферою експертизи;

- здатність пояснити ланцюжок міркувань зрозумілим способом;
- здатність ухвалювати рішення і видавати коментарі та поради за сумнівних вхідних даних;
- наявність можливості поступового нарощування системи;
- використання набору формалізованих правил;
- здатність самонавчатися;
- отримання поради не у вигляді таблиці з цифр, а природною мовою.

Однак, варто враховувати той факт, що для розробки такої предметної експертної системи необхідна участь особливого роду фахівців, які володіють сукупністю знань і виконують функції «посередників» між експертами в предметній галузі та експертними системами.

1.2 Застосування експертних систем в освіті

Експертні системи в освіті являють собою важливий сегмент штучного інтелекту, орієнтований на автоматизацію освітніх та управлінських процесів. Вони функціонують на основі комп'ютерних алгоритмів, що імітують логіку і міркування людини-експерта, і застосовуються для оптимізації індивідуального навчального шляху студентів, надаючи їм рекомендації щодо курсів, навчальних матеріалів і кар'єрних траєкторій, що ґрунтуються на академічних інтересах студентів, попередньому досвіді та успішності.

В основі таких систем також лежать алгоритми штучного інтелекту, включно з машинним навчанням та опрацюванням природної мови, що дають змогу аналізувати навчальні дані та надавати персоналізовані рекомендації.

Ефективність експертних систем в освіті залежить від якості та обсягу даних про учнів, а також від їхньої здатності інтегруватися з іншими освітніми платформами та інформаційними системами.

Експертні системи в освіті можна класифікувати на основі різних критеріїв, включно з їхньою функціональністю, цілями використання, методами реалізації та типами застосовуваних алгоритмів (табл. 1.1).

Таблиця 1.1

Класифікація ЕС в освіті за різними критеріями

Критерій	Опис
Функціональність	Діагностичні, прогностичні, консультативні, навчальні, планування та оптимізація, обробка природної мови.
Цілі використання	Підтримка прийняття рішень, навчання і розвиток, автоматизація оцінювання, персоналізація навчального процесу.
Методи реалізації	На базі правил, машинне навчання, штучні нейронні мережі, нечітка логіка, експертні системи, засновані на випадках.
Типи алгоритмів	Алгоритми класифікації, регресії, оптимізації, кластеризації, асоціативні правила, генетичні алгоритми.

У зв'язку з розширенням і розвитком дистанційної освіти експертні системи останнім часом зазнають значного зростання та прискореного розвитку, стаючи дедалі популярнішими та широко використовуваними. Це зумовлено їхньою здатністю забезпечувати ефективну підтримку навчального процесу, адаптуючись до індивідуальних потреб учнів, особливо в умовах віддаленого навчання. Як ілюстрацію цього тренду, розгляньмо по одному прикладу експертної системи в кожній із ключових категорій, що підкреслить їхню значущість і різноманітність застосування в сучасному освітньому контексті.

«Knewton» - система адаптивного навчання, яка адаптує навчальний процес до індивідуальних потреб і стилю навчання кожного студента. Адаптивні системи можуть змінювати зміст, темп або складність матеріалу залежно від успішності та відгуку студента [29].

Рекомендаційна система «Coursera» пропонує студентам курси, додаткові матеріали або освітні ресурси на основі їхніх попередніх досягнень, інтересів та академічних цілей. Ця система допомагає учням у навігації по великій кількості доступних освітніх можливостей [30].

Системи підтримки прийняття рішень «Canvas Analytics» призначена для допомоги вчителям та адміністрації в ухваленні освітніх та управлінських рішень. Такі подібні системи можуть надавати аналіз даних про студентів, прогнозувати академічні тенденції та допомагати в плануванні навчальних програм [31].

Інтелектуальні тьюторські системи, наприклад, «Carnegie Learning», симулюють сам-на-сам інтерактивне навчання між викладачем і студентом. Вони забезпечують персоналізований зворотний зв'язок і керівництво, адаптуючись до рівня знань і швидкості навчання студента [32].

Системи управління навчанням (LMS) самі по собі не завжди є експертними системами, але багато хто з них інтегрує елементи експертних систем для більш ефективного управління курсами, відстеження успішності та надання аналітичних даних. Популярна експертна система «Blackboard» інтегрує елементи експертних систем для керування курсами та аналітики успішності [33].

Системи оцінювання та тестування використовуються для автоматизації процесу оцінювання та тестування, пропонуючи адаптивні тести, автоматичну оцінку відповідей і аналіз результатів тестів для визначення рівня знань студентів. Прикладом можна привести експертну систему «Quizlet» [34].

Системи кар'єрного консультування, така, як «PathSource», що допомагає студентам і професіоналам планувати свою кар'єру, аналізуючи їхні інтереси, навички та потреби ринку праці [35].

В контексті використання експертних систем в освіті в Україні, проведений аналіз та пошук [29-35] не надає конкретних прикладів експертних систем, які б активно використовувались у освітніх установах країни. Однак, можна зазначити, що експертні системи в освіті загалом фокусуються на використанні знань для

вирішення специфічних завдань, які зазвичай вимагають експертної оцінки. Такі системи включають компоненти, які:

- зберігають структуровані знання про предметну область;
- оновлюють динамічні знання на основі нової інформації;
- використовують робочі знання для розв'язання конкретних завдань.

Тому, далі мова може йти про існуючі різні платформи та програми, які використовуються в освітній сфері, деякі з них можуть включати в себе елементи експертних систем. Найбільш поширені освітні платформи та програми можуть включати:

Електронні платформи для дистанційного навчання. Це можуть бути місцеві або міжнародні платформи, які надають доступ до навчальних курсів, лекцій та інших освітніх матеріалів. Аналіз ресурсів показав найпопулярніші на сьогодні Coursera, Tutorroom, Khan Academy, EdX, RCampus, Alison, Єдина школа, Мій Клас, Всеукраїнська школа, Google Знання [30].

Системи управління навчанням (LMS), платформи які дозволяють учням та вчителям керувати навчальним процесом, включаючи завдання, оцінки, комунікацію та інші аспекти навчання, такі як Moodle, Adobe Captivate Prime, Google Classroom [33].

Інтерактивні освітні інструменти - програми та додатки, які використовуються для підвищення взаємодії та залучення студентів у навчальний процес, такі як Prometheus, iLearn, Дія.Освіта та інші [31].

Аналітичні та оцінні інструменти – це платформи для оцінювання успішності студентів, аналізу даних про навчання та виявлення тенденцій. В Україні існують різні аналітичні та оцінні інструменти, які використовуються в освітній сфері. Одним із важливих напрямків є розвиток та впровадження компетентнісного підходу в освіті, який включає в себе інновації та оцінку ефективності освітніх програм. Наприклад, в Україні проводяться дослідження та аналізи в контексті освітньої реформи «Нова українська школа», яка спрямована на модернізацію базової та

профільної школи. Ці дослідження включають оцінку якості освіти, ефективності використання бюджетних коштів та розвиток інноваційних освітніх технологій [36].

Науково-практичний журнал «Освітня аналітика України» публікує матеріали, присвячені актуальним питанням освітньої статистики, аналітики, а також питанням, пов'язаним із розвитком людського капіталу та державним управлінням у галузі освіти. Журнал висвітлює теми, такі як управління освітою, фінансово-економічні аспекти освіти, бюджетне фінансування соціально-гуманітарної сфери, розвиток інформаційних та освітніх технологій, а також інші актуальні питання науки та інноваційного розвитку.

Таким чином, можна зробити умовивід, що експертні системи, як правило, не розробляються спеціально для освітніх цілей. Їхнє основне призначення полягає в імітації процесу прийняття рішень людиною-експертом у конкретних областях, навчального процесу.

Однак, окремі елементи експертних систем, такі як алгоритми машинного навчання, обробка природної мови, аналітика даних та адаптивні інтерфейси, можуть бути інтегровані в освітні технології. Таке використання допомагає створювати більш персоналізовані та ефективні навчальні досвіди, але це не означає, що самі експертні системи є навчальними інструментами. Вони лише надають технологічну базу, яка може бути використана в освітніх цілях.

І тому є певні причини. Однією з ключових проблем є нездатність повністю відтворити всі знання експерта. Експертні системи можуть включати лише ті знання, які були явно сформульовані та введені в систему. Це означає, що вони часто втрачають «тихі знання» або інтуїтивні аспекти, які експерт може використовувати у своїй роботі [37].

Експертні системи засновані на логічних правилах та алгоритмах для виведення висновків. Однак, людське мислення часто включає нелінійні, творчі та інтуїтивні елементи, які складно формалізувати і реалізувати в комп'ютерних програмах [37].

В контексті навчання, експертні системи можуть бути корисними для надання специфічної інформації або вирішення конкретних задач, але вони часто не можуть забезпечити глибоке розуміння предмету. Вони можуть надавати відповіді або рішення, але без пояснення базових принципів або концепцій, які лежать в основі цих відповідей [37].

Експертні системи можуть володіти обмеженими або «тонкими» знаннями у певній області, що робить їх здатними лише поверхово передати розуміння навчальних тем студентам [37].

Всі ці фактори обмежують ефективність експертних систем як засобів навчання, особливо у контексті розвитку глибокого розуміння та критичного мислення у студентів. На відміну від експертних систем, успішне навчання часто залежить від здатності вчителя адаптувати підходи, враховуючи індивідуальні особливості студентів та контекст навчання.

Таким чином, можна сказати, що, експертні системи в освіті призначені для того, щоб зробити для студентів більш доступними знання, досвід, навички та інтуїцію кваліфікованих фахівців.

Використання експертних систем в освіті можуть вирішуватися різні завдання:

- коригування процесу навчання (швидкості подачі матеріалу, необхідності додаткових відомостей) з урахуванням індивідуальної підготовленості учня, його індивідуальних особливостей;
- діагностику та прогнозування якості засвоєння предметної інформації та формування змін у послідовності подання навчального матеріалу;
- підтримання професійного рівня учня в даній предметній галузі;
- розробку інструментальних систем для цього.

1.3 Експертні системи підтримки індивідуальної освітньої траєкторії

Основна мета застосування експертних систем в освітній сфері полягає у створенні персоналізованого навчального середовища, яке може адаптуватися до унікальних потреб, рівня знань та індивідуальних вподобань кожного студента. Це досягається шляхом розробки індивідуальних освітніх траєкторій, які враховують здібності та інтереси учнів, а також їхній прогрес протягом навчального процесу.

Індивідуалізована освітня траєкторія передбачає використання методів штучного інтелекту та машинного навчання для аналізу навчального процесу студентів, їхніх відповідей, стилів навчання та успішності. На основі цих даних система може пропонувати варіанти курсів, теми для дослідження, а також регулювати рівень складності матеріалу, що дозволяє студентам навчатися в оптимальному для них темпі.

Такий підхід забезпечує гнучкість у навчанні, дозволяючи студентам вибирати курси, які відповідають їхнім академічним інтересам та кар'єрним цілям, тим самим підвищуючи мотивацію та залученість до навчального процесу. Окрім цього, вчителі та освітні інституції можуть використовувати дані, зібрані експертними системами, для поліпшення якості навчальних матеріалів та методів викладання.

В 1990-х роках голландський вчений Герард Вербеке розробив модель персоналізованого навчання, що базується на використанні інформаційних технологій (ІТ) [38-41].

З подальшим розвитком інформаційних технологій з'являються нові можливості для створення індивідуалізованих освітніх технологій. У 2008 році американські експерти, які включають викладачів, керівників освітніх установ та проєктів, а також науковців, у своєму колективному виданні [42] розглядають механізми та засоби постійного поліпшення якості викладання та навчання через ефективний розвиток та обмін освітніми інноваціями. Багато хто з них відстоює ідею про необхідність не тільки застосування сучасних інформаційних технологій та

інструментів в освіті, а й змін у підходах до самого освітнього процесу - виходу за рамки традиційних інститутів та ієрархій, пошуку форм ефективного використання можливостей, що відкриваються новими технологіями в системі освіти.

За останні 20 років дослідження в галузі підтримки ІОТ тривали та активно розвивалися. Дослідники почали активно вивчати можливості створення адаптивних систем навчання, які можуть індивідуалізувати освіту на основі даних про знання та здібності учнів та їх обробки [43,44].

З розвитком штучного інтелекту стали можливими нові підходи до індивідуалізації навчання, багато дослідників працювали над створенням алгоритмів машинного навчання, які можуть адаптувати освіту до потреб кожного учня [45-48]. В останні роки багато дослідників займаються метааналізом та систематичними оглядами, щоб підсумовувати та аналізувати результати багатьох досліджень у галузі підтримки ІОТ. Це дозволяє сформулювати повніше уявлення у тому, які підходи найефективніші для індивідуалізації навчання [49, 50].

Багато українських дослідників розглядають принципи проектування систем підтримки ІОТ в контексті вирішення конкретних освітніх завдань: вивчення іноземної мови [51,52], вивчення мов програмування [53-57], збільшення конкурентоспроможності студентів на ринках праці [58-60], персоналізації навчання на основі використання штучного інтелекту [61-63], технологій віртуальної реальності (ТВР) [64,65], використання Інтернету речей (ІР) [66,67] та інше.

У контексті вище вказаних статей, розглянемо проблеми підготовки фахівців у галузі програмної інженерії за підтримкою ІОТ.

В 2017 році було проведено глобальне дослідження «Еволюція освіти в галузі програмної інженерії: глобальна перспектива» [68], метою якого було виявлення тенденцій у розвитку освіти в галузі програмної інженерії у різних країнах світу. Стаття містить аналіз даних з 12 країн, включаючи США, Китай, Індію, Україну та ін. Автори розглянули питання, пов'язані з проектуванням навчальних програм,

співпрацею між академічними колами та ІТ-ринком праці, а також необхідністю постійного оновлення навчальних матеріалів.

Автори також обговорюють виклики, з якими стикаються викладачі - швидкі темпи технологічних змін та необхідність постійного оновлення своїх знань та навичок. Нарешті, дослідження наголошує на необхідності подальших досліджень та співпраці між академічними колами та ІТ-індустрією для покращення освіти в галузі програмної інженерії.

У статті [69] автори провели систематичний огляд літератури з метою вивчення ефективності застосування ІОТ та використання ІТ у навчанні для покращення освіти в галузі програмної інженерії. Вони проаналізували 57 досліджень, опублікованих у період з 2000 по 2015 роки, і зробили такі висновки: застосування моделей ІОТ та технологічно збагаченого навчання має позитивний ефект на покращення освіти в галузі програмної інженерії, що у поєднанні з іншими методами навчання, такими як традиційні лекції та практичні заняття, можуть бути особливо ефективними.

Найцікавіша частина досліджень, присвячена практиці реалізації ІОТ при підготовці фахівців з програмної інженерії. Системи підтримки ІОТ використовуються у багатьох вишах по всьому світу для підготовки фахівців у галузі програмної інженерії. У США університет Іллінойсу в Урбана-Шампейн пропонує онлайн-курс з програмної інженерії через свою платформу Coursera, яка називається «Software Engineering Essentials» [70] та включає підтримку ІОТ. Інший приклад - Технологічний інститут Джорджії, також пропонує програму онлайн-магістратури в галузі комп'ютерних наук через свою платформу Udacity [71], яка включає підтримку ІОТ.

У 2020-2021 роках на основі опитування випускників фахівців з програмної інженерії румунських університетів та аналізу їх незадоволеності вибором професії, румунські дослідники [72] зробили висновок про необхідність персоналізації навчання за підтримкою ІОТ. Дослідження італійських та англійських дослідників

[73] підтверджують актуальність цього висновку (однак, деякі з цих досліджень можуть мати обмежену репрезентативність для сучасної України, тому, перед застосуванням цих висновків досліджень до українських студентів, слід провести відповідні дослідження з урахуванням особливостей української молоді, щоб отримати більш точні результати).

В Україні, практично в кожній освітній установі перед фаховій вищій освіти використовують системи підтримки ІОТ при навчанні програмної інженерії. В цілому, впровадження СПІОТ має на меті покращити якість освіти, підготовку фахівців, їх конкурентоспроможність на ринку ІТ-праці, задоволеність студентів та забезпечити розвиток особистості студентів.

В статті [74] розглядається проблема навчання програмуванню, яка пов'язана із необхідністю індивідуалізації освітнього процесу для задоволення потреб різних категорій студентів. Для вирішення цієї проблеми автори пропонують використовувати модель формування ІОТ студента, що базується на компетентнісному підході. Модель передбачає, що кожен студент має свої унікальні потреби та особливості, які можуть бути враховані при моделюванні траєкторії. Для цього необхідно провести діагностику рівня знань та навичок студента з програмування та на основі отриманих результатів розробити індивідуальний план навчання. Цей план має враховувати інтереси студента, його здібності, а також вимоги професійної сфери, де він планує працювати після закінчення вузу.

Ключовими темами статті [75] є проблеми формування ІОТ студента у контексті підготовки програмістів на основі системи дистанційного навчання на досвіді Тернопільського національного педагогічного університету імені Володимира Гнатюка. Автор описує принципи, методики та інструменти, що використовуються у системі індивідуальної освітньої траєкторії та розглядає переваги дистанційного навчання у контексті створення індивідуальних траєкторій та підтримки студентів-програмістів.

В статті [76] автори проводять огляд інструментів, які можуть використовуватись для створення моделі ІОТ, а також описують досвід та результати впровадження даної моделі для студентів, які навчаються за спеціальністю «Програмна інженерія» у Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського». Проте, автори наголошують на необхідності врахування низки викликів, пов'язаних з реалізацією запропонованої ними моделі навчання.

Досвід впровадження моделі формування ІОТ студентів на базі Міжнародного гуманітарного університету наведено у статті [77]. Автори звертають увагу на важливість такої моделі для підвищення якості освіти та конкурентоспроможності випускників на ринку праці. У статті також наголошуються на складнощах та проблемах, пов'язаних із впровадженням ІОТ, таких як відсутність універсальних підходів до її формування, недостатня підготовленість викладачів, брак інформації про професійні потреби ринку праці тощо.

У статті [78] висвітлено досвід застосування ІОТ при навчанні дисципліни об'єктно-орієнтоване програмування (ООП) у Національному університеті «Києво-Могилянська академія». Докладно описані особливості викладання ООП, включаючи підбір необхідних матеріалів, методів та засобів навчання, а також контроль та оцінку результатів навчання студентів. Автори статті також обговорюють деякі проблеми, які можуть виникнути при використанні ІОТ, наприклад, необхідність внесення змін до програм навчання, налаштування системи та забезпечення достатнього рівня технічної підтримки.

Таким чином, на сучасному етапі з великою можливістю освітніх порталів і величезними можливостями Інтернету абітурієнти, що ідуть в освітні установи вже володіють певними знаннями, вміннями і навичками в ІТ-технологіях. Дані знання дуже різноманітні. Це також викликає суперечність у освітньому процесі і посилює появу «слабких», «середніх» та «сильних» студентів.

Тобто стандартний підхід до навчання не завжди здатний задовольнити індивідуальні потреби та інтереси кожного студента. Тому стоїть завдання знайти ефективні шляхи індивідуалізації навчання, що включає реорганізацію навчального процесу, використання інформаційних технологій, впровадження систем підтримки індивідуальної освітньої траєкторії.

1.4 Структура навчального процесу із застосуванням індивідуальної траєкторії навчання

Метою впровадження ІОТ в освітній процес є освоєння обраного студентом професійного виду діяльності, тобто досягнення високого рівня сформованості професійних компетентностей, які належать до обраного виду діяльності.

Проте на сьогоднішній день складання індивідуальних освітніх програм на практиці стикається з певними труднощами. По-перше, прийняття студентом відповідальності за власну освіту, усвідомлення її мети та вибору професійного стандарту до початку навчання не можливе через постійні зміни щодо потреб та індивідуальних можливостей особистості. По-друге, проектування такої програми не може врахувати індивідуальний темп навчання студента. Таким чином, індивідуальна освітня програма студента повинна розглядатися як динамічна сутність, яка може змінюватись та коригуватися в ході освітнього процесу.

Основними документами, що використовуються при побудованні моделі процесу навчання через формування освітньої професійної програми (ОПП) є Закон України «Про вищу освіту», Закон України «Про фахову передвищу освіту», Державні стандарти ЗФПО, Національна рамка кваліфікацій та низька інших наказів та стандартів [79-82].

В зв'язку з тим, що об'єктом дослідження є проектування, розробка та впровадження системи підтримки вибору індивідуальних траєкторій навчання в цілісний освітній процес, для всебічного вивчення цього об'єкта доцільно

використовувати системний підхід, в основі якого лежить вивчення об'єктів як системи. Специфіка системного підходу полягає в тому, що він орієнтує дослідження на виявлення різноманітних типів зв'язків складного об'єкта та зведення їх у єдину теоретичну картину [83]. Для опису зв'язків між вище вказаними об'єктами було побудовано модель процесу навчання (рис. 1.3).

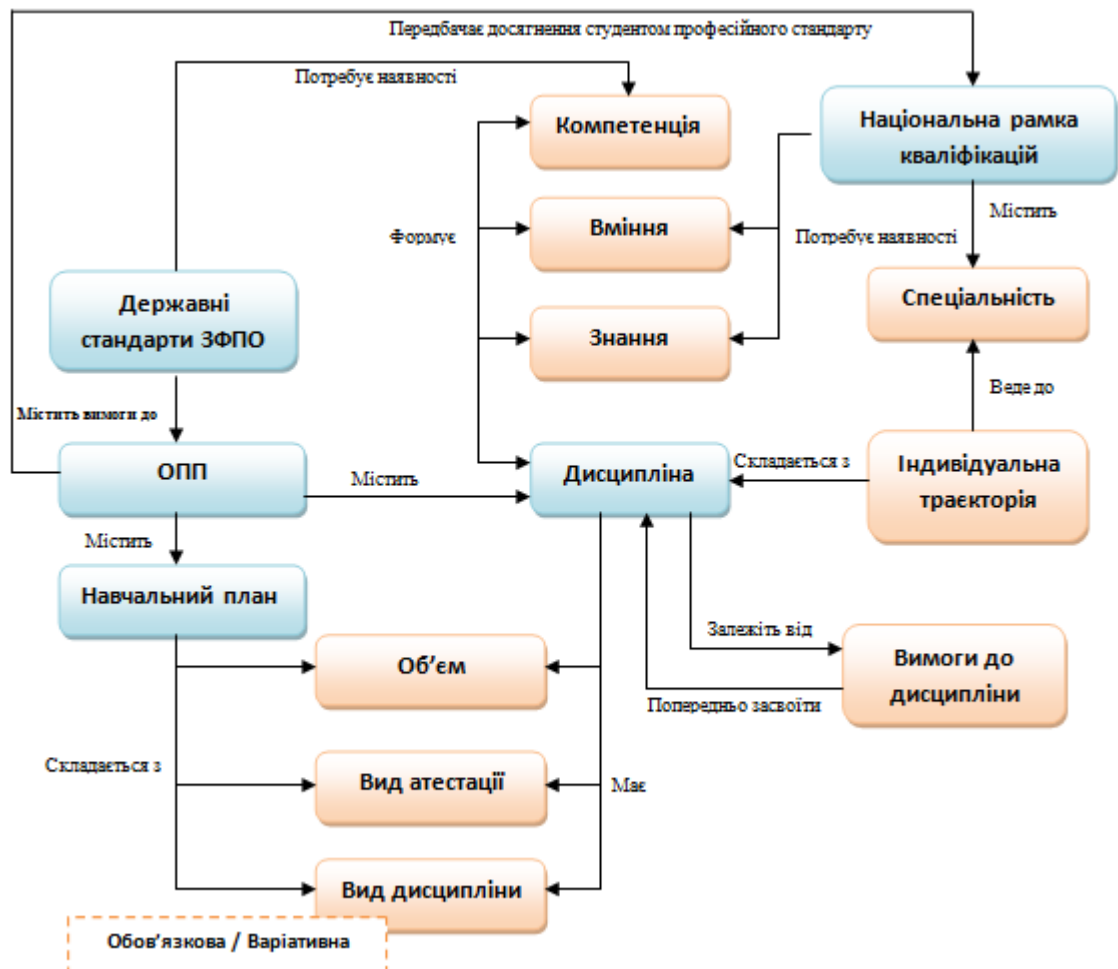


Рисунок 1.3 – Концептуальна модель навчального процесу з застосуванням ІОТ

Ключові елементи, які виділяються на схемі:

- державні стандарти ЗФПО (Загальні фахові стандарти вищої, передвищої фахової освіти), який вимагає наявності ОПП (освітньо-професійної програми);
- ОПП, яка містить вимоги до навчального плану;

- навчальний план, який складається з дисциплін, що в свою чергу мають об'єм, який веде до виду атестації або виду дисципліни;
- дисципліна, яка містить знання та уміння, формується коментарями і веде до професійного стандарту.

Таким чином, на основі схеми навчального процесу можна зробити кілька висновків, що освітня система зосереджена на інтеграції національних стандартів та компетенцій, що вказує на її орієнтацію на досягнення визначених якісних результатів. Присутність індивідуальних траєкторій вказує на гнучкість у підходах до навчання, надаючи студентам можливість особистісного вибору в процесі освіти. Структура враховує різноманітні види дисциплін та атестацій, що підкреслює важливість адаптації та диференціації в навчальному процесі. Схема демонструє взаємозв'язок між навчальним планом, дисциплінами та компетенціями, підкреслюючи їх роль у формуванні фахових кваліфікацій. Освітній процес вбудований у ширший контекст професійної підготовки, враховуючи сучасні потреби ринку праці та вимоги до кваліфікацій.

Одним із ключових моментів зазначеної моделі є співвіднесення професійних та освітніх стандартів, що забезпечує виконання вимог професійного стандарту внаслідок реалізації освітнього стандарту. У зв'язку з цим була побудована модель професійного стандарту (рис.1.4).



Рисунок 1.4 – Модель професійного стандарту

Розглянемо цілісну структуру навчального процесу через його компоненти [84]:

Цільовий компонент:

Мотив навчання – те, що внутрішньо спонукає людину до навчальної діяльності, спрямовуючи цю діяльність на задоволення пізнавальної чи будь-якої іншої потреби.

Змістовний компонент:

Зміст дисципліни визначається за логічною структурою навчального плану (НП). Ця логіка визначена послідовністю вивчення дисциплін, їх взаємозв'язком між собою. Логічна структура навчального плану дозволяє наочно показати порядок вивчення та взаємозв'язок дисциплін та надалі сформувати індивідуальні освітні програми студентів. Крім того, той, хто навчається, познайомившись з нею, зможе зрозуміти, для чого йому необхідно вивчати той чи інший предмет, а викладач - вибудувати навчальний процес з урахуванням взаємозв'язку дисциплін [85].

Ресурсний компонент:

Ресурсний компонент представлений такими підсистемами нижчого рівня:

- матеріально-технічне забезпечення визначатиме, які можливості має викладач для організації навчального процесу;
- базові знання визначаються як знання, без яких неможливе свідоме освоєння дисципліни, що викладається [85]. Базові знання студента визначатимуть, який зміст дисциплін може вибрати викладач. Якщо базові знання недостатні, то викладачеві, перш ніж переходити до вивчення безпосередньо дисципліни, необхідно заповнити їх. Якщо ж базові знання достатні, то студенти можуть освоювати безпосередньо дисципліну;
- особисті якості студентів будуть показувати рівень можливостей студента в освоєнні дисципліни: репродуктивний чи продуктивний, і зрештою – рівень сформованості компітентностей та освоєння професійного виду діяльності;

– кваліфікація викладача визначається вмінням адаптувати зміст навчального курсу до конкретної аудиторії, умінь створювати та застосовувати навчальні технології, методики навчання відповідно до ІОТ кожного студента.

Результативний компонент:

Результатом впровадження індивідуальних траєкторій навчання буде рівень освоєння обраного виду професійної діяльності (ВПД).

Діяльний компонент:

Виходячи з того, що головними суб'єктами педагогічного процесу виділяють викладача та студента, то діяльний компонент визначатиметься як єдність діяльності цих суб'єктів: навчання та викладання [84].

На основі вищесказаного виділимо в структурі процесу навчання із застосуванням ІТО дві підсистеми: навчання та викладання, які в свою чергу охоплюватимуть всі перераховані вище компоненти цілісного освітнього процесу, оскільки будь-яка діяльність включає в себе мету, засіб, результат і сам процес діяльності [84]. Причому цільовий та результативний компоненти перебувають на стику двох виділених підсистем, оскільки вони єдині, як студента, так і для викладача.

Важливою характеристикою структури будь-якого об'єкта є взаємодія його частин, тому розглянемо взаємний вплив виділених компонентів процесу навчання із застосуванням ІОТ (табл. 1.2).

Таблиця 1.2

Взаємовідносини компонентів навчального процесу

Компонент	Зв'язок	Компонент
Мета	Симетричний: Розвинена мотивація до навчання дозволяє досягти мети та навпаки поставлена мета може стимулювати мотивацію	Мотив навчання
	Несиметричний: Мета, як бажання досягти певного рівня володіння ПД визначає логічну структуру НП	Логічна структура НП

Продовження таблиці 1.2

	Несиметричний: Рівень залежатиме від поставленої мети	Рівень засвоєння обраного виду ПД
	Несиметричний: Те, як викладач організує процес навчання, щоб реалізувати мету, залежить від його кваліфікації	Кваліфікація викладача
	Несиметричний: Якщо особисті якості студента відповідають професійним якостям обраного виду ПД, то студенту буде простіше досягти мети	Особисті якості студента
	Несиметричний: Якщо базові знання не дозволяють освоїти комплекс дисциплін, що становлять логічну структуру НП ПД на належному рівні, то досягнення мети буде неможливим	Базові знання студента
	Симетричний: Достатній рівень матеріально-технічного забезпечення показує наскільки легко можна досягти мети. Ціль, у свою чергу, визначатиме вибір відповідного матеріально- технічного забезпечення серед наявного	Матеріально- технічне забезпечення
Мотив навчання	Несиметричний: Чим вищий мотив навчання, тим простіше освоїти комплекс дисциплін, що становлять логічну структуру НП	Логічна структура НП
	Несиметричний: Чим вищий рівень мотивації, тим вище рівень засвоєння обраного виду ПД	Рівень засвоєння обраного виду ПД
	Несиметричний: Рівень кваліфікації визначає, чи зможе викладач підвищити рівень мотивації студента	Кваліфікація викладача
	Несиметричний: Особисті якості студента визначають чи зможе студент підвищити власну мотивацію до навчання	Особисті якості студента
	Несиметричний: Висока мотивація дозволяє подолати недоліки в базових знаннях	Базові знання студента
	Несиметричний: Дозволить підвищити рівень мотивації у студента	Матеріально-технічне забезпечення

Продовження таблиці 1.2

Логічна структура НП	Несиметричний: Рівень освоєння дисциплін показує рівень освоєння обраного ВПД	Рівень засвоєння обраного виду ПД
	Несиметричний: Високий рівень кваліфікації викладача дозволить студентам впоратися з будь-якою дисципліною	Кваліфікація викладача
	Несиметричний: Зміст дисциплін навчального плану визначає які особисті якості потрібні студенту для досягнення мети	Особисті якості студента
	Несиметричний: Зміст дисциплін навчального плану визначає які базові знання потрібні студенту для досягнення мети	Базові знання студента
	Несиметричний: Чим вищий рівень зв'язності дисциплін навчального плану, тим краще має бути матеріально-технічне забезпечення навчального процесу	Матеріально-технічне забезпечення
Рівень засвоєння обраного виду ПД	Несиметричний: Рівень освоєння обраного виду ПД як результат залежатиме від усіх елементів індивідуальної траєкторії навчання	Кваліфікація викладача
		Особисті якості студента
		Базові знання студента
		Матеріально-технічне забезпечення
Кваліфікація викладача	Несиметричний: Викладач з високою кваліфікацією зможе досягти результату, навіть якщо особистісні якості студента не відповідатимуть обраному виду професійної діяльності	Особисті якості студента
	Несиметричний: Викладач з високою кваліфікацією зможе досягти результату, навіть якщо базові знання студента не будуть відповідати обраному виду професійної діяльності	Базові знання студента

Продовження таблиці 1.2

	Несиметричний: Кваліфікація викладача визначає, чи зможе він організувати ефективний процес навчання за доступного матеріально-технічного забезпечення	Матеріально-технічне забезпечення
Особисті якості студента	Несиметричний: Особисті якості студента визначають наскільки швидко студент впорається з підвищенням рівня базових знань	Базові знання студента
	Симетричний: Певні якості студента дозволять досягти мети навіть за низького рівня матеріально-технічного забезпечення. Високий рівень матеріально-технічного забезпечення дозволить досягти мети навіть при відсутності певних якостей у студента	Матеріально-технічне забезпечення
Базові знання студента	Несиметричний: Високий рівень матеріально-технічного забезпечення дозволить досягти мети навіть при відсутності певних базових знань у студента	Матеріально-технічне забезпечення

Представимо виділені зв'язки у вигляді графа. Підставою існування такого графа є множина виділених компонентів та відношення, яке задане на цій множині: «Елемент X впливає на елемент Y» (рис. 1. 5).

Цільовий та результативний компоненти є спільними для обох підсистем, навчання та викладання, і описують кінцеві результати, до яких прагнуть як студенти, так і викладачі. В контексті експертних систем, ці компоненти можуть бути визначені через набір очікуваних результатів або компетенцій, які студент повинен розвинути, і рівень майстерності, якого викладач прагне досягнути в своїй педагогічній практиці. Цільовий і результативний компоненти об'єднують зусилля студентів і викладачів, створюючи спільний простір для досягнення освітніх цілей, який може бути ефективно підтриманий за допомогою експертних систем.

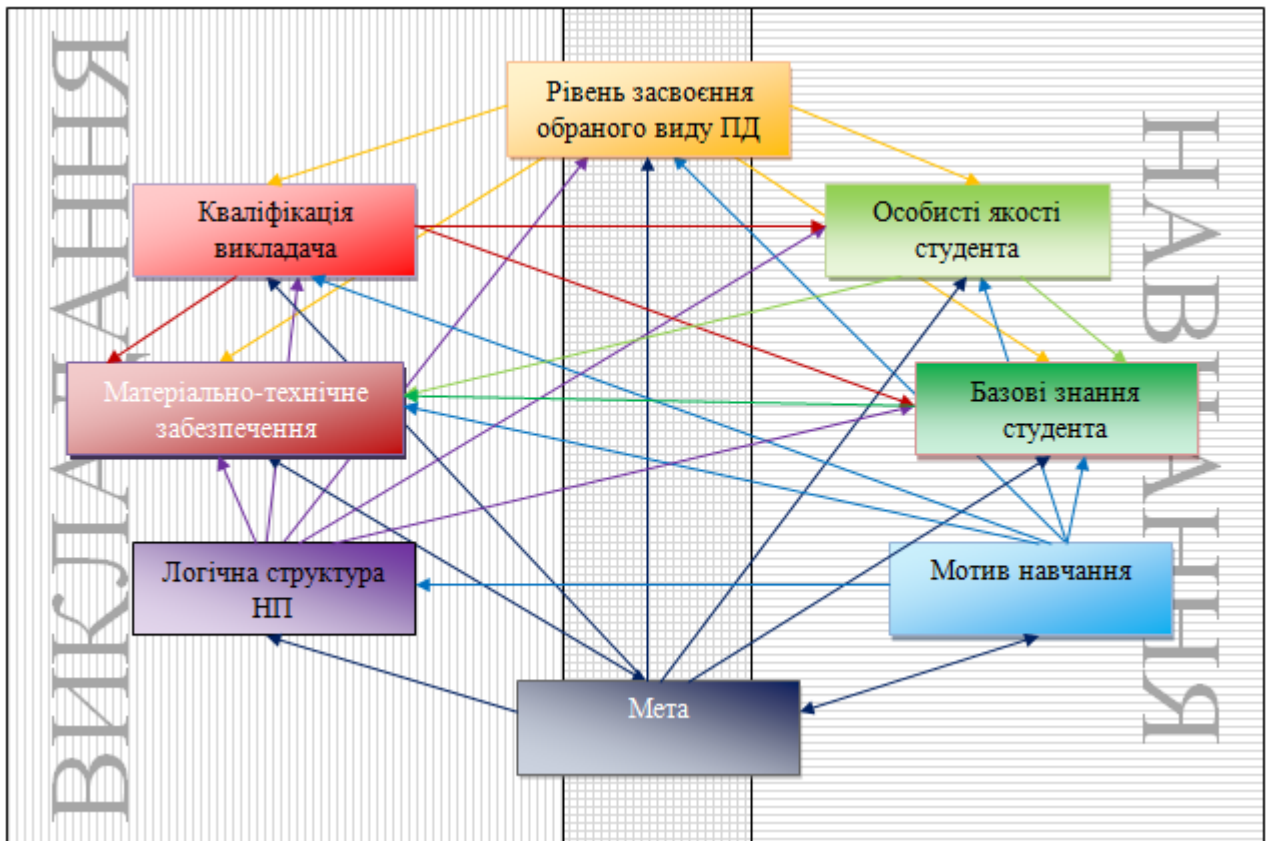


Рисунок 1.5 – Структура процесу навчання із застосуванням ІОТ

Таким чином, індивідуальна освітня траєкторія є центральним елементом сучасної освітньої системи, орієнтованої на інтеграцію ІТ та врахування індивідуальних особливостей студентів. Експертні системи можуть підвищити якість навчання, забезпечуючи персоналізацію освітнього процесу та адаптацію до особистих потреб та цілей студентів.

1.5 Застосовність експертних систем у навчальному процесі

Як було вказано вище, комп'ютерні навчальні системи, в тій чи в іншій формі містять окремі компоненти експертних систем. У більшості випадків ці технології застосовуються як пошук прийомів, які б давали максимально близьку кореляцію з

оцінкою, запропонованою викладачем, і відповідно пов'язані з блоком обробки та визначенням оцінки за результатами навчання.

Однією з головних проблем є проєктування навчального середовища, здатного повноцінно функціонувати за мінімальної участі викладача і при цьому забезпечувати належний контроль навчального процесу [6,7].

Структура експертної системи містить у собі чотири основні елементи: блок прийняття рішень, базу знань, базу даних та інтерфейс взаємодії із зовнішнім середовищем.

Блок прийняття рішень (БПР) є ядром, що надає інтелектуальності всій системі. Принципи функціонування БПР забезпечують застосовність системи в цілому у вирішенні поставленого завдання. Усі інші елементи виконують допоміжну роль.

У базі знань містяться функції, що керують реакцією БПР на події, які виникають у процесі роботи та являють собою алгоритмічну модель ухвалення рішень експертами в даній галузі знань. Функції, що керують реакцією, алгоритмізуються у вигляді нечітких правил, які відносно просто редагуються відповідно до прикладної специфіки. Логістика інформаційних потоків БПР багато в чому залежить від функцій реакцій на події.

База даних є сховищем інформації будь-якого формату, що стосується даної теми на даний момент.

Інтерфейс користувача - це спосіб взаємодії між користувачем і системою. У потужних системах користувачеві надається можливість ставити запитання і отримувати відповіді звичайною мовою, однак розробка такої системи - доволі складне завдання, простіше створити систему, в якій запитання і відповіді обираються із заданого списку.

База знань містить відомі факти, виражені у вигляді об'єктів і умов. Крім описових уявлень про дійсність, вона включає вирази невизначеності - обмеження на достовірність факту [86]. У цьому відношенні вона відрізняється від традиційної

бази даних внаслідок свого символного, а не числового або буквеного змісту. Під час опрацювання інформації бази даних використовуються заздалегідь визначені логічні правила. Відповідно, база знань, що представляє вищий рівень абстракції, має справу з класами об'єктів, а не з самими об'єктами (рис.1. 6).



Рисунок 1.6 – Діаграма основних компонентів експертної системи та їхня взаємодія

Вхідні дані передаються до бази знань. Від бази знань потік інформації йде до механізму виведення. Механізм виведення пов'язаний з інтерфейсом користувача, де користувачі можуть взаємодіяти з системою.

Модуль навчання знаходиться в центрі схеми і забезпечує зв'язок між інтерфейсом користувача та адаптивним навчанням, даючи змогу системі адаптувати навчальний матеріал і завдання до потреб студента.

Оцінка та зворотний зв'язок є компонентом, що забезпечує оцінку роботи студента і взаємодіє з вихідними даними, що містять результати та аналітику для подальшого використання в системі.

Лінія від блоку оцінка та зворотний зв'язок до вихідних даних ілюструє, що зворотний зв'язок впливає на вихідні дані, які можна використати для подальшого аналізу та вдосконалення навчального процесу.

Ця схема дає чітке уявлення про те, як експертні системи можуть використовуватися для підтримки персоналізованого навчання, адаптуючи освітній процес до індивідуальних потреб студентів.

Центральною складовою експертної системи є механізм, що здійснює пошук у базі знань за правилами раціональної логіки для отримання рішень [87].

Ця компонента активізується при отриманні запиту користувача і виконує такі завдання:

- порівнює інформацію, що міститься в запиті користувача, з інформацією бази знань;
- здійснює пошук певної мети або причинних зв'язків;
- оцінює відносну визначеність чинників, ґрунтуючись на відповідних коефіцієнтах довіри, пов'язаних із кожним чинником.

Наступний компонент експертної системи - рівень довіри. У базу знань надходять факти. Зв'язок між фактами представлений евристичними правилами - виразами декларативного знання про відносини між об'єктами [37]. Кожне таке правило має складову «якщо» (передумову) і компонент «то» (висновок), які визначають прямі та зворотні причинно-наслідкові зв'язки. Дійсні твердження тільки ймовірні, тобто ступінь їхньої визначеності не завжди абсолютні. Такі твердження відносної впевненості часто ґрунтуються на статистичних, імовірнісних або просто суб'єктивних передумовах [37]. Загальноприйнята шкала і шкала, що використовується в системі навчального призначення, варіюються від 0 до 100 - рівень довіри найвищого ступеня.

Розділимо побудову експертної системи для освітніх цілей на низку підзадач. Розглянемо діаграму (рис. 1.7), яка відображає структуру побудови експертної системи для освітніх цілей, розбиваючи процес на дев'ять основних підзадач. Кожна підзадача представлена послідовно, відображаючи логіку розробки системи. Ця діаграма відображає комплексний підхід до створення експертних систем навчального процесу, який забезпечує ретельне врахування всіх аспектів роботи

системи - від збору та обробки знань до адаптації до індивідуальних потреб користувачів.

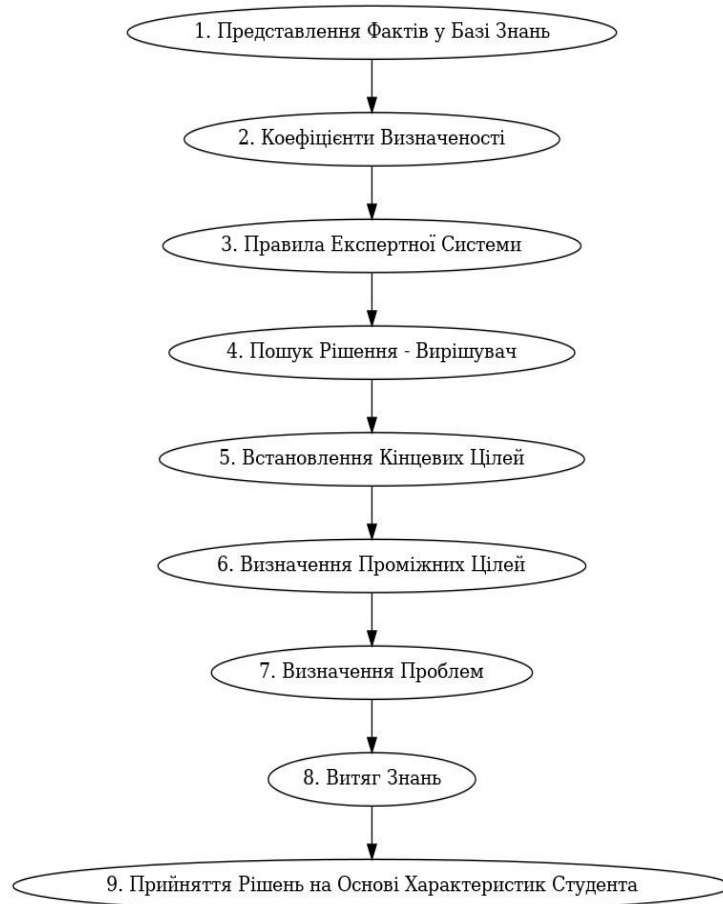


Рисунок 1.7 – Структура побудови експертної системи для освітніх цілей

Подання фактів у базі знань. База знань складається з фактів і правил. Факти описують те, що відомо про предметну область на даний момент. Правила встановлюють ситуаційні, концептуальні, причинні або прецедентні взаємозв'язки між цими фактами [88]. Представимо факти, які визначають об'єкти, описуючи їхні атрибути та надаючи їм еквіваленти або значення. Під словом «об'єкт» мається на увазі як фізичний предмет (наприклад, «оцінка»), так і загальні уявлення («добре» або «відмінно») [88]. З об'єктами пов'язуються атрибути, за якими і активізуються об'єкти в системі (наприклад, «незадовільна оцінка»). Для впорядкування виразів

фактів об'єднуємо їх у пари «об'єкт – значення», з'єднавши ім'я об'єкта з ім'ям атрибута. Для представлення об'єктів у базі знань використовують ланцюгові списки. Кожна одиниця цього списку називається вузлом і містить поля, у які заноситься інформація про об'єкт. Одне з полів слугує покажчиком, що повідомляє системі, де шукати наступний вузол списку. Крім того, кожен вузол у списку об'єктів має покажчик на список значень, пов'язаних з ім'ям цього об'єкта.

Коефіцієнти визначеності у контексті освітніх експертних систем використовуються для оцінки ймовірності того, що певна інформація або дані відповідають реальності. Ці коефіцієнти можуть застосовуватися для моделювання невизначеності у знаннях студентів, виборі навчальних матеріалів, оцінюванні здібностей і прогресу у навчанні, дозволяють викладачам зрозуміти, наскільки вони можуть довіряти прогнозам системи.

Правила експертної системи. Експертна система обробляє символічне представлення реальності за допомогою евристичних правил і методу зворотного ланцюжка. У цьому методі консультація (порада) починається з певної конкретної мети або кінцевого результату. Цей підхід протилежний методу прямого ланцюжка, де міркування починається з визначення проблеми. Правило складається з двох частин: передумови та висновку. Як передумова, так і висновок є фактами бази знань, вираженими парами «об'єкт – значення». Правила мають такий формат: Якщо <передумова>, ТО <висновок>. Наприклад, передумова: «Якщо студент вирішив усі завдання», то висновок: «Він отримає 5». Або передумова: «Якщо студент отримав 5», то висновок: «Він вирішив усі завдання». Сенс правила у тому, що якщо правильна передумова, то правильний і висновок. Ці прості відносини «якщо-то» представляють вузли рішення, за якими машина виведення просувається до поставленої мети. Правило може містити в собі булевський оператор «І» для утворення складніших виразів [67].

Пошук рішення - вирішувач. База знань представляє опис предметної області експертної системи. Вирішувач є інтерпретатором правил, який використовує факти

цієї бази знань для розв'язання поставлених проблем. Він здійснює це шляхом формулювання пробних гіпотез і перевірки їх на відповідність зазначеній меті. Оператор задає мету консультації у вигляді імені об'єкта. Вирішувач використовує набір правил, намагаючись отримати значення зазначеного об'єкта - мети. Система продовжує пошук до тих пір, доки одне з передбачуваних рішень не виявиться правильним.

Постановка кінцевих цілей. Насамперед потрібно визначити віддалену, кінцеву мету - який результат очікуємо отримати, коли експертна система розв'яже проблему. Мета має виражати дію або подію, що відображає вплив експертної системи на загальний перебіг подій. На цьому рівні опису ступінь невизначеності може бути досить великим.

Визначення проміжних цілей. Для кожної кінцевої мети система може мати низку проміжних цілей - дій або приватні проблеми. Бувають цілі, досягнення яких призводить до досягнень кінцевих цілей. На цьому рівні цілі теж виражаються як дії або події, але їхня невизначеність навмисно скорочується. Коли визначено проміжні цілі, проблема розпадається на під задачі. Кожна мета представляє передбачуваний результат, пропонований програмою під час розв'язання приватної проблеми. Крім установлення бажаних результатів розв'язання проблеми, проміжна мета виконує й іншу важливу функцію внаслідок своєї специфічної проблеми, вона може потребувати для свого розв'язання окремих або ізольованих систем знань.

Визначення проблем. Після того, як експерти (викладачі) зі знань визначили цілі експертної системи, стають досить очевидними типи проблем, які належить розв'язувати, і спосіб підходу системи до їх розв'язання.

Витяг знань. Після визначення цілей і завдань системи група розробників стикається з проблемою ефективного вилучення експертних знань. Найбільш очевидним методом є просте опитування. Інтерв'ювання експертів або отримання знань із довідників - це прямий шлях, однак він забере надто багато часу. Використання аналогій або моделей може значно скоротити цю роботу. Ось один із

варіантів: зібрати файл експертних рішень, пов'язаних із проблемами в предметній області, потім проаналізувати головні правила, на яких ґрунтуються ці експертні рішення. Або, отримати джерело інформації через пряме спостереження або експериментування. Класичний спосіб вилучення експертних знань полягає в тому, щоб пройти через усі стадії процесу експертизи, спостерігаючи, як у тому чи іншому випадку чинить експерт (викладач).

Проблема вибору подальших дій вирішується експертною системою на основі характеристик студента за кількома ключовими показниками: результати з поточного матеріалу, засвоєння попереднього, поточний морально-психологічний стан студента. Це може бути як продовження викладання нового матеріалу, так і питання з попередніх пройдених тем або закінчення навчання.

Запропонований основоположний принцип побудови подібних систем може бути застосовані практично за будь-якої моделі навчання. Закладена у системі можливість редагування та нарощування бази знань сприятиме своєчасному коригуванню правил і фактів, що керують процесом навчання. Усе описане вище свідчить про потужність і гнучкість алгоритмів і функцій, закладених в експертній системі, що дають змогу проектувати на її базі сучасні навчальні комплекси нового покоління.

1.6 Постановка задачі дослідження

Наукова гіпотеза дослідження полягає в наступному. Експертні системи в освіті відіграють ключову роль у формуванні та підтримці персоналізованого навчального процесу. Вони сприяють ефективному вирішенню задач, пов'язаних з індивідуалізацією навчання та розробкою освітніх траєкторій, що відповідають специфічним потребам кожного студента. Коректність вибору індивідуальної освітньої траєкторії в свою чергу, може бути досягнуто шляхом застосування експертних систем в інтеграції з інтелектуальними системами прийняття рішення

(ІСППР). Таким чином, досягнення головної мети дисертаційного дослідження – підтримка вибору індивідуальної освітньої траєкторії, може бути реалізовано шляхом розробки нових моделей і методів побудови ЕС ІППР. Галуззю застосувань ЕС ІППР в такому разі є ситуації підтримки вибору оптимальної індивідуальної освітньої траєкторії людиною, що приймає рішення (ЛПР) в умовах невизначеності.

Проведений аналіз наукових досліджень дозволив визначити концептуальні засади створення ЕС ІППР для підтримки індивідуального вибору освітньої траєкторії та її основні функції:

1. Система повинна мати базу даних, що включає і накопичує інформацію за трьома напрямками: освітні данні, експертна інформація - формалізовані знання експертів у предметній галузі та базу прецедентів - інформація щодо розвитку подій в освітньому середовищі, що вже мала місце в минулому.
2. Система повинна базуватися на принципах моделювання індивідуальних освітніх траєкторій, спрямованих на досягнення конкретних освітніх та професійних цілей. Необхідна можливість прогнозування розвитку освітньої ситуації та візуального подання такого прогнозу.
3. Математичні моделі, які використовуються в системі, повинні відповідати вимогам адекватності, ефективності та технічної здійсненності.
4. Форми відображення інформації в системі повинні забезпечувати її швидке сприйняття та високий рівень усвідомлення користувачем ЕС ІППР.

1.7 Висновки до першого розділу

Під час проведення дослідження було проаналізовано виклики та перспективи створення та впровадження в освітній процес експертних систем. З урахуванням проведеного аналізу, можна стверджувати, що експертні системи в освіті відіграють ключову роль у формуванні та підтримці персоналізованого навчального процесу. Вони сприяють ефективному вирішенню задач, пов'язаних з індивідуалізацією

навчання та розробкою освітніх траєкторій, що відповідають специфічним потребам кожного студента.

Зроблено висновок про значущість інтеграції інтелектуалізації та персоналізації в експертні системи для підвищення їхньої ефективності та адаптації до вимог сучасного освітнього процесу. Виявлено, що впровадження методів інтелектуалізації та персоналізації може значно покращити точність і релевантність освітніх рекомендацій.

Також, дослідження показало, що застосування експертних систем дозволяє здійснювати діагностичні та аналітичні функції, які сприяють виявленню потенційно неефективних або невідповідних навчальних шляхів. Такі системи надають студентам рекомендації щодо оптимального вибору рішень та дій для досягнення їх особистих навчальних та професійних цілей.

Водночас ці системи стикаються з певними обмеженнями, такими як проблеми доступу для кінцевих користувачів та труднощі, які пов'язані з впливом факторів невизначеності та суб'єктивності на прийняття освітніх рішень. Крім того, експертні системи можуть бути обмежені у великих предметних областях та часто не враховують інтуїтивні аспекти експертних знань.

Встановлено, що на поточний момент часу існує нагальна потреба в створенні сучасних експертних інтелектуальних систем підтримки прийняття рішення в слабоформалізованій освітньої галузі, які враховують інтелектуальний аналіз даних та попередній досвід людини, що приймає рішення, і забезпечують прийняття нею рішень в умовах невизначеності.

У контексті розвитку штучного інтелекту, експертні системи зарекомендували себе як визначний напрямок, здатний вирішувати складні задачі в управлінні навчанням та прийнятті складних рішень.

РОЗДІЛ 2

МЕТОДИ ТА МОДЕЛІ ПОБУДОВИ ЕКСПЕРТНОЇ СИСТЕМИ ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ РІШЕННЯ

В даному розділі розглядається побудова експертної системи інтелектуальної підтримки рішень при виборі індивідуальної освітньої траєкторії студента людиною, яка приймає рішення (ЛПР). Пропонується спільне використання експертної системи (ЕС) та інтелектуальної системи підтримки прийняття рішень (ІСППР). ЕС ІППР дають змогу розробити структуровану схему, що відображає весь перебіг процесу прийняття рішень у невизначених і розпливчастих умовах, якою є освітня галузь. Такі системи здебільшого моделюють та інтерпретують дії користувача з організації своїх знань про об'єкт і роблять із них висновки та ефективно супроводжують весь процес прийняття рішення. У певному сенсі ЕС ІППР допомагає ЕОМ застосувати досвід користувача при прийнятті ефективних рішень.

У зв'язку з тим, що метою дослідження є підвищення ефективності та оптимізація процедури прийняття рішень взагалі та інформаційної підтримки ЛПР в слабоформалізованих навчальних процесах зокрема, для досягнення поставленої мети пропонується використання сценарно-прецедентного підходу побудови ЕС ІППР, як інтеграції ситуаційного (сценарного) і прецедентного підходів [89]. Очевидна перевага пропонованого підходу визначається широким поширенням як прецедентного, так і сценарного підходів, а отже, методів їх формалізації (математичний апарат) та інструментально-програмної реалізації (мови програмування, формати представлення даних тощо). Для підтвердження доцільності та працездатності пропонованого сценарно-прецедентного підходу до побудови ЕС ІППР в роботі розглянуто типовий приклад слабоформалізованої ситуації прийняття рішення - вибір наукового керівника і теми дипломного проекту студентів-випускників.

2.1 Навчальний заклад, як соціально-економічна система

Навчальний заклад (НЗ) відрізняється своїм нематеріальним характером основних процесів. Особливі труднощі кількісного вимірювання діяльності зосереджені в галузі організації навчального процесу, який характеризується категоріями «знання», «вміння», «навички».

Та більше, НЗ - це соціально-економічна система, що динамічно розвивається [90-93]. Специфіка цих систем багато в чому визначається їхньою слабо структурованою природою. Основний елемент невизначеності в них - людський фактор, що значно знижує прогностичні можливості процесу прийняття рішень [89,94].

Взагалі кажучи, соціально-економічні процеси у сфері освіти можуть бути однозначно відокремлені в окремий кластер процесів, оскільки їхня специфіка визначається суб'єктивністю та неоднорідністю самої освітньої системи. У загальному вигляді освітня система уявляється багаторівневою системою з численними прямими і зворотними зв'язками, що володіє окремими властивостями, які, роблячи позитивний внесок у структурування і функціонування системи загалом, утруднюють за необхідності швидке застосування коригувальних і керівних впливів. Обсяги і темпи надходження інформації до людини, яка приймає рішення, можуть перевершувати професійні можливості ЛПР щодо її осмислення. Для прийняття ефективних управлінських рішень в освіті необхідно враховувати всі чинники, що впливають та правильно оцінювати поточну ситуацію [95].

Підготовка прийняття рішення ЛПР вимагає аналізу та прогнозування можливого стану об'єкта управління, оскільки від точності прогнозування залежить успішність впровадження цих рішень. Чим точніше прогноз, тим більш ефективно можна спрогнозувати наслідки впровадження стратегій та планів [96].

Точність прогнозування, у свою чергу, залежить від кількох факторів, серед яких актуальність, достовірність та повнота зібраної та збереженої інформації за

кожним параметром оцінки грають ключову роль. Недостатня актуальність чи повнота інформації можуть призвести до неточностей у прогнозуванні та втрати можливостей для вчасного реагування на зміни.

Усі перелічені фактори ускладнюють процес вибору правильного варіанта рішення. Причому тільки автоматизація процесу обробки та вивчення даних не приведе до бажаного результату. Динамічний характер зміни обсягів інформації вимагає від систем управління освітою адаптивності та гнучкості. Стандартні методи обробки та аналізу даних можуть бути недостатніми для ефективного управління такими об'ємами інформації. Тому впровадження інтелектуальних систем підтримки прийняття рішень (ІСППР) є найперспективнішим шляхом.

ІСППР зосередили у собі технології з високим рівнем автоматизації процесів підготовки та обробки інформації, а також вироблення варіантів рішень. Для забезпечення освітнього процесу використовуються інтелектуальні інформаційні системи [98-101], експертні навчальні системи, інтелектуальні мультиагентні системи [97], інтелектуальні системи безперервного навчання [100-104], інтелектуальні адаптивні навчальні системи [105-108] та інші види програмного забезпечення. Для аналізу і вироблення пропозицій у ІСППР використовують різні моделі та підходи: інформаційний пошук [109], інтелектуальний аналіз даних [110], пошук знань у базах даних [114], міркування на основі прецедентів [101-102], імітаційне моделювання [100], еволюційні обчислення [112] і генетичні алгоритми, нейронні мережі [113], ситуаційний аналіз, когнітивне моделювання тощо.

Таким чином, ІСППР можуть враховувати комплексність і динаміку інформаційного середовища, використовуючи алгоритми штучного інтелекту та машинного навчання для аналізу великих обсягів даних, виявлення взаємозв'язків та закономірностей, а також для прогнозування можливих наслідків впровадження різних стратегій та рішень. Вони дозволяють здійснювати інформовані та обґрунтовані рішення в умовах нестабільності та невизначеності, що є ключовими

аспектами управління освітою. Отже, ІСППР стають важливим інструментом для підтримки рішень в сучасній освітній сфері.

2.2 Інтелектуальні системи підтримки прийняття рішень

Під ІСППР розуміють організаційно-технічну систему, що складається з інтелектуального комплексу засобів підтримки прийняття рішень, що взаємопов'язаний і взаємодіє з користувачами та мережами ЕОМ і виконує розв'язання заданих завдань [50].

Укрупнена структура ІСППР (рис.2.1) аналогічна іншим інтелектуальним системам і включає чотири основні підсистеми: адаптивна підсистема, інтерактивна підсистема, підсистема обробки даних, підсистема управління та зовнішнє середовище.



Рисунок 2.1 – Структура ІСППР

Перша підсистема складається з кількох рівнів: мікро-, макро- і мета рівнів [44]. На цих рівнях ефективно використовувати різні моделі еволюції [83,86,112].

На кожному рівні будується інтегрована цільова функція, визначаються свої граничні умови й обмеження. Далі будується узагальнена цільова функція для всієї ІСППР. Цільові функції на мікро-, макро- і метарівні будуть окремими цільовими функціями:

$$f_1 = \alpha_1 k_1 + \alpha_2 k_2 + \dots + \alpha_n k_n, (\alpha_1 + \alpha_2 + \dots + \alpha_n = 1), \quad (2.1)$$

де f_1 – цільова функція на мікрорівні, $k_1, k_2, \dots, k_n$ – окремі цільові функції, $\alpha_1, \alpha_2, \dots, \alpha_n$ – коефіцієнти, що визначають ступінь належності (важливості) кожного критерію на цьому рівні, n – кількість приватних критеріїв. Відповідно, на макро- і метарівні запишемо:

$$f_2 = \beta_1 m_1 + \beta_2 m_2 + \dots + \beta_n m_n, (\beta_1 + \beta_2 + \dots + \beta_n = 1), \quad (2.2)$$

$$f_3 = \gamma_1 r_1 + \gamma_2 r_2 + \dots + \gamma_n r_n, (\gamma_1 + \gamma_2 + \dots + \gamma_n = 1), \quad (2.3)$$

де $f_2(f_3)$ – цільова функція на макрорівні (метарівні), $m_1, m_2, \dots, m_n(r_1, r_2, \dots, r_n)$ – окремі цільові функції, $\beta_1, \beta_2, \dots, \beta_n(\gamma_1, \gamma_2, \dots, \gamma_n)$ – коефіцієнти, що визначають ступінь належності (важливості) кожного критерію на цьому рівні, n – кількість окремих критеріїв.

Для спрощення взято загальне число окремих критеріїв (n) на всіх рівнях. Тоді узагальнена цільова функція ІСППР запишеться:

$$f = \delta_1 f_1 + \delta_2 f_2 + \delta_3 f_3, (\delta_1 + \delta_2 + \delta_3 = 1), \quad (2.4)$$

де $\delta_1, \delta_2, \delta_3$ – коефіцієнти, що визначають ступінь належності (важливості) кожного критерію мікро-, макро- і метарівня в інтегрованому критерії (f) для всієї ІСППР.

На кожному з цих рівнів реалізуються задані комплекси алгоритмів під час взаємодії із зовнішнім середовищем. Інтелектуальна система повинна весь час взаємодіяти із зовнішнім середовищем, тому це середовище є регулятором роботи ІСППР. Як зовнішнє середовище виступатиме ЛПР, експертна підсистема тощо. Взаємодія може здійснюватися різними способами:

- на основі чіткої і нечіткої функціональної відповідності (багато до одного);
- на основі чіткої та нечіткої ін’єктивної відповідності (один до багатьох);
- на основі бієктивної відповідності (взаємнооднозначної).

Друга підсистема аналізує вхідні описи мовою користувача на основі наявних знань і формує внутрішнє неповне та розпливчасте уявлення завдання. Тут важливим є завдання чіткої множини вихідних даних:

$$X = \{x_1, x_2, \dots, x_n\} \quad (2.5)$$

Та визначення на ній нечіткої множини:

$$\tilde{A} = \{ \langle \mu_1, x_1 \rangle, \langle \mu_2, x_2 \rangle, \dots, \langle \mu_l, x_l \rangle \}, \quad (2.6)$$

при цьому $\mu_i \in M(X)$ функція належності елемента x_i , а величина $M(X)$ змінюється на інтервалі $[0,1]$.

Третя підсистема перетворює неповний і розпливчастий опис завдання на повний і чіткий і знову передає його інтерактивній підсистемі. Далі процес відбувається ітераційно до отримання задовільного рішення.

Четверта підсистема керує процесом розв’язання, взаємодіючи з адаптивною, інтерактивною підсистемою та підсистемою обробки даних.

Запропонуємо одну з можливих концептуальних схем ІСППР (рис. 2.2). Тут діалоговий процесор є структурною одиницею інтелектуального інтерфейсу введення та виведення. ЛПР у взаємодії з діалоговим процесором на основі адаптації, блочно-ієрархічного підходу і принципів декомпозиції може здійснювати

розбивку необхідної задачі прийняття рішень на множину функціональних підзадач меншої розмірності. Інакше кажучи, після розбиття задачі прийняття рішення на частини і розв’язання їх під час одержання загального рішення мають бути не втрачені основні властивості, що притаманні загальній задачі.

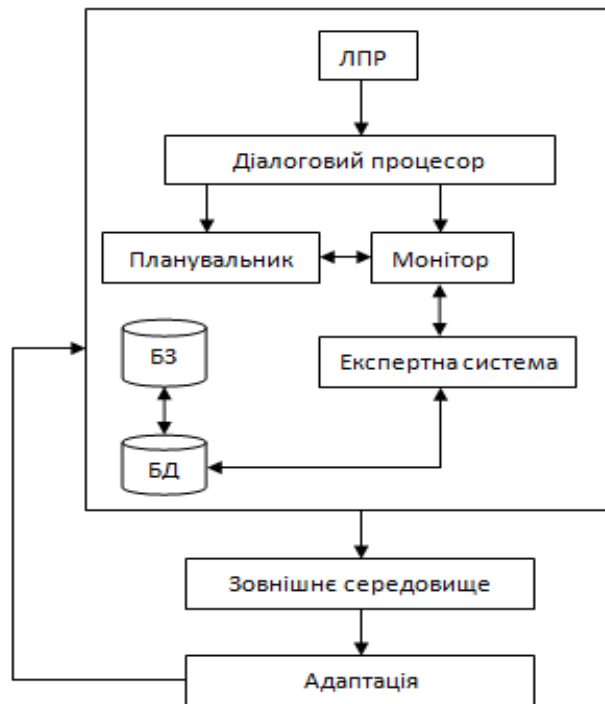


Рисунок 2.2 – Концептуальна схема ІСППР

Основна мета планувальника - пошук інформації. Спільно з експертною системою він здійснює аналіз перетворених даних, отриманих через діалоговий процесор. Залежно від значень внутрішніх, зовнішніх і керуючих параметрів конструктивних обмежень, заданих режимів роботи тощо обирається кілька альтернативних шляхів вирішення. Спільно з базою знань (БЗ), базою даних (БД), сховищем даних і блоком адаптації планувальник здійснює реалізацію процедур пошуку для знаходження множини рішень і для вибору деякої підмножини ефективних рішень серед наявних.

В ІСППР БЗ складатиметься з трьох основних блоків: база загальних знань, база системних знань, база прикладних знань. У першому блоці зберігатимуться

загальні знання, необхідні для розв'язання всіх задач прийняття рішень. У другому блоці зберігатимуться знання про всі внутрішні зв'язки самої системи. У третьому блоці зберігатимуться всі прикладні знання, наприклад, правила й обмеження на процес прийняття рішень, комплекси алгоритмів тощо. На відміну від стандартних баз даних (БД), які також присутні в ІСППР і взаємодіють із БЗ, останні дають змогу опрацьовувати знання і в результаті цього отримувати нові знання [32].

Як було зазначено вище, у даному дослідженні розглядається ідея спільного використання експертної системи (ЕС) та інтелектуальної системи підтримки прийняття рішень (ІСППР). ЕС ІППР дозволяє створити структуровану схему, яка відображає весь процес прийняття рішень у сфері освіти, яка характеризується невизначеністю та розпливчастістю. ЕС ІППР в основному моделюють та інтерпретують дії користувача, використовуючи свої знання про предмет, і на їх основі роблять висновки та ефективно супроводжують весь процес прийняття рішення. У певному сенсі, ЕС ІППР сприяє тому, що комп'ютер може використовувати експертний досвід користувача для прийняття ефективних рішень.

2.3 Експертні системи інтелектуальної підтримки прийняття рішення

На сьогодні важливим завданням під час ухвалення рішень є розробка і побудова нечітких ЕС. Це програмно-алгоритмічні комплекси, що виконують різноманітні функції ухвалення рішень. Вони можуть: консультувати користувача; аналізувати проміжні та кінцеві результати ПР; навчати користувача й ЕОМ; навчатися в процесі розв'язання конкретного завдання; давати поради користувачеві та ЕОМ; класифікувати проблему, що розглядається; здійснювати пошук на заданих математичних моделях; ухвалювати конкретне рішення на будь-якому етапі оптимізації; робити достовірні висновки з неповних і розпливчастих даних і знань; взаємодіяти з іншими ЕС; передавати й набувати нових знань.

Таким чином, експертна системи інтелектуальної підтримки прийняття рішень - це складний процес, який включає кілька етапів:

1. Перший етап - це чітко визначити, яку конкретну проблему або завдання потрібно вирішити за допомогою ЕС ІППР.

2. Другий етап - збір даних та експертного знання, які необхідні для розв'язання проблеми (це може включати статистичні дані, експертні оцінки, правила прийняття рішень тощо).

3. Третій етап - отримані дані піддаються аналізу та обробці з використанням методів статистики, машинного навчання та інші техніки для виявлення взаємозв'язків та закономірностей у даних.

4. Четвертий етап - визначається архітектура експертної системи, включаючи структуру бази знань, алгоритми прийняття рішень та інтерфейс користувача.

5. П'ятий етап - створюються програмні модулі, які реалізують функціональність експертної системи на основі розробленої архітектури.

Після реалізації системи проводяться тестування для перевірки її працездатності та відповідності вимогам. Після цього можуть проводитися корекції та налагодження для поліпшення роботи системи. Після цього система може бути використана для прийняття рішень у реальних умовах, а також може потребувати підтримки та обслуговування.

Розв'язання цього завдання доцільно здійснювати за допомогою математичних моделей і методів оцінювання можливих сценаріїв (варіантів рішень), що сприймаються як рекомендації для подальшого оцінювання ЛПР, та інтелектуального аналізу. Наявність фази інтелектуального аналізу є абсолютно необхідною з огляду на те, що рішення, які ухвалені ЛПР для досягнення мети, можна розділити на дві великі групи:

- несподівані, принципово нові та новаторські рішення, які не можна виявити в результаті автоматизованого опрацювання даних на комп'ютері;

– рішення, засновані на типових сценаріях, які отримані на аналогіях або шляхом комбінації окремих рішень, генерація яких доступна комп'ютеру.

Тому цей тип ЕС ІППР представляє собою інтегрований інформаційний процес, що складається з технічних засобів та програмного забезпечення, які взаємодіють з людиною. Він здатен на основі знань та інформації, а також з мотивацією, синтезувати мету, приймати рішення та знаходити ефективні способи досягнення цілей.

Існує чотири основні задачі, що треба вирішити під час розроблення ЕС ІППР:

- перша задача – це автоматизація трудомісткого процесу обробки та завантаження зібраної інформації;
- друга задача пов'язана з виявлення нових, часом прихованих, взаємозв'язків у даних та їх інтерпретації в інтересах особи, яка приймає рішення;
- третя задача пов'язана з комплексністю та багатокроковістю схвалюваних рішень, що зумовлюють труднощі прогнозування можливих наслідків;
- четверта задача пов'язана з впливом факторів суб'єктивності та невизначеності на ЛПР.

2.3.1 ETL процеси

У базах знань експертних систем міститься фактична та емпірична інформація, правила і процедури, необхідні для моделювання знань і досвіду експертів у конкретній галузі. Ці дані можуть бути подані у вигляді баз даних, файлів знань або інших структурованих форматів, які дають змогу системі здійснювати пошук, витяг і використання інформації для розв'язання завдань.

На рис. 2.3 наведено інформаційне сховище, яке являє собою базу узагальненої інформації, що формується з безлічі зовнішніх і внутрішніх джерел, на основі яких виконуються статистичні групування та інтелектуальний аналіз даних.

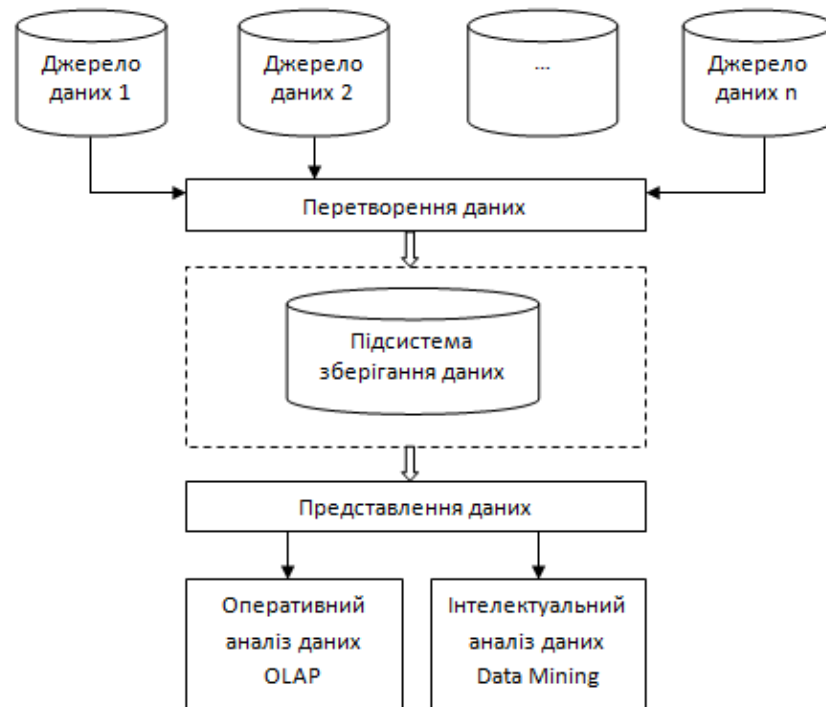


Рисунок 2.3 – Архітектура інформаційного сховища ЕС ІППР

Інформація до БД може надходити з різноманітних джерел, тому дані необхідно попередньо обробити. В процесі попередньої обробки даних для подальшого аналізу також необхідно враховувати додаткові проблеми, що виникають під час роботи з даними, і брати до уваги такі процедури, як очищення і попередня обробка даних, фільтрація даних, сортування даних, перекодування даних (у тому числі, створення категоріальних змінних із кількісних), пропуски в даних і методи їх вирішення та виявлення аномальних даних.

Таким чином, вирішення першого завдання полягає у застосуванні методів ETL (Extract, Transform, Load) [116-118] для перетворення даних у формат, придатний для експертної системи.

Програмне забезпечення цього класу призначене для вилучення, приведення до загального формату, перетворення та завантаження даних у сховище даних.

Розглянемо традиційну схему ETL (рис.2.4):

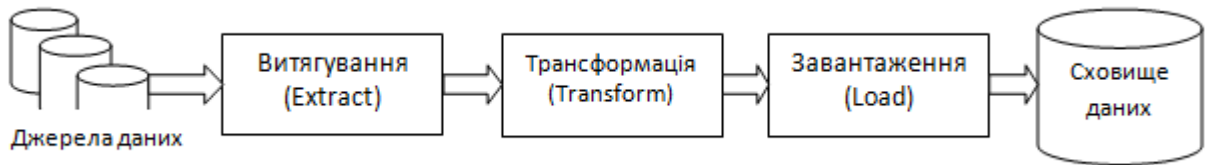


Рисунок 2.4 – Традиційна модель ETL процесів

Витягування (Extract). Етап витягування є першим кроком у процесі ETL. Під час цього кроку дані збираються з одного або кількох джерел. Ці джерела можуть бути різноманітними, включно з реляційними базами даних, файлами (наприклад, CSV, Excel), API веб-сервісів, потоковими даними тощо. Цей процес включає зчитування даних і їхнє перетворення у формат, придатний для подальшої обробки.

Трансформація (Transform). На етапі трансформації дані очищаються, збагачуються, нормалізуються та перетворюються для задоволення бізнес-вимог і правил. Цей процес може включати:

- очищення даних від помилок, дублікатів або некоректних значень;
- нормалізація для забезпечення консистентності форматів;
- збагачення шляхом додавання додаткової інформації з інших джерел;
- агрегація даних для підсумовування;
- фільтрація для видалення непотрібної інформації.

Завантаження (Load). Останній етап, завантаження, включає перенесення оброблених даних у цільову систему зберігання. Це може бути склад даних, де дані зберігаються в структурованому форматі для подальшого аналізу, звітності та підтримки прийняття рішень. Важливо, що дані повинні бути доступними для кінцевих користувачів у вигляді, який найкраще підходить для їхніх потреб, що може включати регулярне оновлення даних для забезпечення їх актуальності.

Але, виходячи з завдань даного дослідження та подальшої інтеграції інформаційно-аналітичної системи в експертну систему, запропонуємо інше

застосування процесу ETL, щоб переміщувати інформацію з однієї програми до іншої на постійній основі.

Експертні системи не є сховищами даних у класичному сенсі. Вони являють собою програмні системи, які використовують знання експертів для вирішення певних завдань. Однак, експертні системи можуть містити в собі бази знань, які зберігають інформацію, використовувану для прийняття рішень.

У базах знань експертних систем міститься фактична та емпірична інформація, правила і процедури, необхідні для моделювання знань і досвіду експертів у конкретній галузі. Ці дані можуть бути подані у вигляді баз даних, файлів знань або інших структурованих форматів, які дають змогу системі здійснювати пошук, витяг і використання інформації для розв'язання завдань.

Таким чином, ETL може бути використаний для перетворення даних у формат, придатний для експертної системи за наступними етапами (рис.2.5):

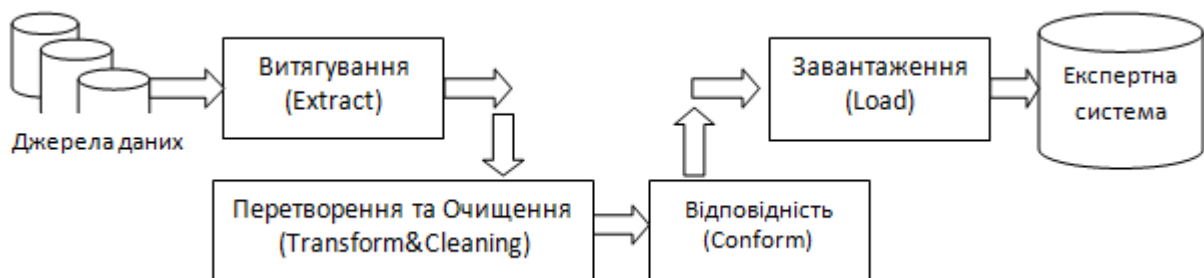


Рисунок 2.5 – Модифікована модель ETL процесів

Витягування (Extract). Необроблені дані, які витягнуті з вихідних систем, частіше за всього структуровані. Дані від структурованих вихідних систем (таких як бази даних або Excel-файли) часто пишуться в плоскі файли або реляційні таблиці на цьому кроці. Це дозволяє вихідному витягу бути простим і швидким і дозволяє легко перезапускати витяг, якщо стався збій. Зібрані дані можуть тоді бути зчитані багаторазово, якщо це необхідно. Також первісно зібрані дані можуть також бути збережені принаймні для одного циклу вилучення, щоб відмінності між послідовними вилученнями могли бути обчислені.

Перетворення (Transform). На етапі перетворення даних до витягнутих даних застосовується низка правил або функцій, щоб підготувати їх до завантаження в кінцеву ціль. Деякі дані не потребують будь-якого перетворення взагалі. Але в багатьох випадках може знадобитися один або кілька типів перетворень для задоволення потреб сховища даних:

- вибір тільки певних стовпців для завантаження;
- переведення закодованих значень (наприклад, якщо вихідна система кодує оцінки студентів як «2», «3», «4», «5», а кінцева система «незадовільно», «задовільно», «добре», «відмінно»);
- кодування значень вільної форми (наприклад, перетворення «жінка» на «Ж»);
- отримання нового розрахункового значення;
- сортування або впорядкування даних на основі списку стовпців для підвищення продуктивності пошуку;
- об'єднання даних з декількох джерел (наприклад, пошук, злиття) даних;
- транспонування або поворот (перетворення стовпців на рядки або навпаки);
- розщеплення стовпця на кілька стовпців (наприклад, перетворення списку слів, розділених комами, у вигляді рядка в одному стовпці, в окремі значення в різних колонках);
- відокремлення стовпців, що повторюються;
- перегляд і перевірка відповідних даних з таблиць;
- перевірка даних. Якщо якісь дані не пройшли перевірку, відбувається повна відмова від даних, часткова відмова, або відмова не відбувається взагалі. Таким чином, жоден, деякі або всі дані передаються до наступного кроку залежно від конструкції та правил обробки винятків та інші.

Очищення (Cleaning). У більшості випадків рівень якості даних для вихідних систем, відрізняється від якості, необхідної сховищу даних. Якісна обробка даних може містити багато кроків:

- перевірка на допустимі значення;
- гарантія несуперечливості значень.

Відповідність (Conforming). Відповідність структури даних потрібна щоразу, коли два або більше джерел даних об'єднуються в сховище даних. Окремі джерела даних не можуть бути запитані разом, якщо хоча б деякі текстові мітки в цих джерелах не ідентичні і якщо подібні числові показники не раціоналізовані математично так, щоб відмінності та відносини між цими показниками були доцільними. Відповідність структури даних вимагає загальних угод створення документообігу.

Завантаження (Loading). Тепер потрібно зробити дані готовими до запитів, завантажуючи до іншої системи.

Існують два підходи до написання ETL процедур:

- їх можна написати вручну;
- можна скористатися спеціалізованими засобами ETL.

Кожен із підходів має низку переваг і недоліків, і вибір того чи іншого методу написання процедур ETL визначається вимогами до підсистеми завантаження даних у кожному конкретному випадку. Детально переваги та недоліки кожного з підходів до написання процедур ETL описано в роботі [118].

Для реалізації ETL процесів в даному дослідженні було розроблено процедури, що зберігаються, мовою C# у середовищі Visual Studio Windows Forms. Написання вручну дає можливість використання широко поширених парадигм програмування, наприклад, об'єктно-орієнтованого програмування, можливість використання багатьох наявних методик і програмних засобів, що дають змогу автоматизувати процес завантаження даних та можливість побудови найбільш гнучкого рішення.

Далі здійснюється інтелектуальний аналіз перетворених даних за допомогою різних методів DataMining, таких як класифікація даних та пошук асоціативних правил тощо. Застосування перерахованих методів дасть змогу виявити приховані закономірності, не представлені явно у вихідних даних [119].

2.3.2 Інтелектуальний аналіз даних

Взагалі інтелектуальний аналіз даних (data mining) - це не нове явище, це давня практика. Однак, у зв'язку зі зростанням застосування інформаційних технологій в освіті, виник інтерес до нових методів та підходів до автоматизованого виявлення нових, часом прихованих, взаємозв'язків у даних та їх інтерпретації в інтересах особи, яка приймає рішення. Взагалі термін «data mining» має на увазі виявлення нових, нетривіальних і практично корисних знань, необхідних для прийняття рішень у різних галузях діяльності.

Educational data mining (EDM) - це напрям у межах інтелектуального аналізу даних. Метою EDM є опрацювання та аналіз даних, отриманих у межах освітнього процесу. Одне з основних завдань EDM - знаходження прихованих закономірностей у даних. Таким чином, суть EDM - це методи, інструменти та дослідження, які пов'язані з автоматичним витяганням даних, що стосуються навчальної діяльності [120].

Інтелектуальний аналіз освітніх даних порівняно новий напрям досліджень. У 2008 році група вчених уперше провела конференцію з EDM у канадському Монреалі. Відтоді конференція стала щорічною і далі проводилися у США, Іспанії, Греції. З 2010 року видається журнал «Educational data mining» [120], основний зміст якого присвячено питанням використання методів математичної статистики та інтелектуального аналізу даних в освіті.

Дані, які накопичені про різні аспекти роботи освітнього закладу, можуть бути використані для прийняття рішень з управління освітнім процесом. Дослідження [6,7] показують, що великий вплив на процес прийняття рішень у сфері освіти чинять експертні оцінки відповідальних осіб (зокрема, завідувачів кафедр та/або відділень), а також досвід та інтуїція викладачів тощо. Таким чином, ЛПР несе велику відповідальність за його наслідки. Щоб зробити процес прийняття рішення більш об'єктивним, необхідно користуватися сучасними спеціалізованим науковим

апаратом, реалізованим у системах підтримки прийняття рішень та експертних системах зокрема.

Існує безліч завдань, у яких методи статистики, машинного навчання і вилучення знань корисні для всіх учасників освітнього процесу. Більшість досліджень у цій галузі передбачає використання у навчальному процесі так званих інтелектуальних навчальних систем (Intelligent Tutoring Systems, ITS) [98-100]. Однак, незважаючи на розроблення і застосування ITS науковими дослідниками [102], такі системи поки що не набули такого широкого поширення в освітніх установах України, на відміну від інформаційних систем управління та систем дистанційного навчання (Learning Management System, LMS). У зв'язку з цим найбільший інтерес становить аналіз освітніх даних, отриманих із цих джерел, а не з інтелектуальних навчальних систем.

З огляду на сказане вище, дамо визначення, що інтелектуальний аналіз освітніх даних - це сукупність методів виявлення в даних раніше невідомих, практично корисних та інтерпретованих знань про освітній процес та його учасників з метою підтримки прийняття рішень.

Основні методи інтелектуального аналізу даних базуються на статистиці, машинному навчанні та теорії баз даних. До цих методів належать: пошук аномалій, пошук асоціацій, класифікація, кластеризація, регресійний аналіз, факторний аналіз, нейронні мережі, пошук асоціативних правил, вивід по прецедентах тощо.

У межах перелічених методів реалізовано велику кількість підходів, що базуються на різних алгоритмах. Зокрема, роботу [103] присвячено прогнозуванню успішності з використанням методів класифікації. У статті [104] прогнозування успішності здійснюється на основі логів з LMS Moodle. У [121] розв'язується окреме завдання передбачення незадовільної оцінки з навчальної дисципліни за допомогою генетичних алгоритмів.

В рамках даного дослідження для інтелектуального аналізу даних пропонується використання методу бінарної класифікації.

В загальному вигляді завдання класифікації можна представити таким чином:

Нехай є безліч об'єктів, які розділені на класи за деякими ознаками. У навчальній вибірці задано кінцеву множину об'єктів та їхніх ознак. Для кожного з об'єктів навчальної вибірки відомо, до яких класів вони належать. Належність же решти об'єктів до класів невідома. Потрібно побудувати алгоритм, здатний класифікувати довільний об'єкт із вихідної множини, тобто вказати найменування (або номер) класу, до якого об'єкт віднесено в результаті застосування алгоритму класифікації.

Математично це можна записати таким чином. Нехай X - це множина описів об'єкта, Y - це множина класів. Існує невідома цільова функція $f: X \rightarrow Y$, значення якої відомі тільки на навчальній вибірці:

$$X_m = \{(x_1 y_1, x_2 y_2, \dots, x_m y_m)\}, \quad (2.7)$$

де $x_1, x_2, \dots, x_m$ - вектори ознак об'єктів, а $y_1, y_2, \dots, y_n$ - найменування класів, до яких належать відповідні об'єкти.

Потрібно побудувати алгоритм $a: X \rightarrow Y$, здатний класифікувати довільний об'єкт $x \in X$. Бінарна класифікація передбачає наявність лише двох класів, тобто $Y = \{-1; +1\}$.

Таким чином, нехай X описується набором параметрів $X = \{x_1, x_2, \dots, x_m\}$, де кожен параметр x_i кореспондує з певним типом даних, які можна знайти в базі даних. Для кожного параметра x_i існує функція визначення відповідності f_i , яка бере значення з бази даних і перевіряє, чи воно відповідає даному параметру X .

Формування параметрів: для кожного параметра x_i множини параметрів X виконується наступний процес:

- з БД отримується набір даних $D_i = \{d_{i1}, d_{i2}, \dots, d_{im}\}$, який можливо використовувати для параметра x_i ;

– застосовується функція f_i для кожного елемента d_{ij} із набору D_i , де $j=1,2, \dots, m$. Функція $f_i(d_{ij})$ повертає значення, яке вказує на ступінь відповідності між d_{ij} та x_i .

Створення тестової вибірки:

- в результаті вищеописаного процесу для кожного параметру x_i утворюється набір значень відповідності, які визначаються як $F_i = \{f_i(d_{i1}), f_i(d_{i2}), \dots, f_i(d_{im})\}$;
- тестова вибірка для X формується, як сукупність всіх наборів F_i для кожного параметру $F = \{F_1, F_2, \dots, F_n\}$.

Таким чином, тестова вибірка F представляє собою зібрані та оброблені дані з бази даних, які відповідають множині описів об'єкта X і можуть бути використані для подальшого аналізу.

Запропонований метод надає можливість використання інтелектуального аналізу даних без надлишкового «занурення» ЛПР у деталі базових даних або проміжної аналітики. Математичні методи дають змогу виявляти тенденції, що існують у даних. Зазвичай такі закономірності не можна виявити під час простого перегляду даних, оскільки зв'язки занадто складні, а обсяг вибірки надмірно великий.

Вище зазначена концепція побудови ІСППР для освітньої галузі поки що є недостатньою для прийняття рішень у слабоструктурованих предметних областях. Недолік полягає в статистичному методі, а саме в тому, що така система дає приблизну відповідь, тому що вона заснована на статистичних даних. Саме тому видається перспективною побудова повноцінної експертної системи інтелектуальної підтримки прийняття рішення, що допомагає накопичувати, систематизувати, інтегрувати та ефективно використовувати досвід експертів предметної області для прийняття ЛПР релевантного рішення.

2.3.3 Сценарно-прецедентний підхід

Метод міркування на основі прецедентів (Case-Based Reasoning (CBR)) дає змогу розв'язувати нові проблеми, витягуючи минулі успішні прецеденти та

адаптуючи їх до нових умов. В основі методу прецедентів лежить фундаментальна ідея, згідно з якою саме так часто чинить особа, яка ухвалює рішення, знаходячи у своїй пам'яті інформацію, що стосується розв'язуваної задачі, і адаптуючи минулі умови до нових реалій. Вважається, що базові принципи CBR-підходу було викладено в роботах Р. С. Шенка [8, 9], який уперше запропонував узагальнити знання про минулі ситуації у формі так званих скриптів, контейнерів знань, які можна використовувати як у процесі навчання, так і в процесі ухвалення рішень. Ця модель, звана динамічною пам'яттю Шенка, стала у наступні роки фундаментом для створення великої кількості систем, заснованих на прецедентному підході.

Суть прецедентного підходу полягає в застосуванні накопиченого досвіду розв'язання проблеми в процесі вироблення розв'язання нових задач.

Основним поняттям є прецедент - це структуроване представлення накопиченого досвіду у вигляді даних і знань, що забезпечує його подальшу автоматизоване опрацювання за допомогою спеціалізованих програмних систем [101]. Як правило, прецедент складається з опису проблемної ситуації та сукупності дій, що вживаються для усунення цієї проблемної ситуації (її вирішення).

Основні характеристики прецедентів [101]:

- прецедент являє собою особливе знання, прив'язане до контексту, що дає змогу використовувати знання на прикладному рівні;
- прецеденти можуть набувати різної форми (вигляду): охоплюючи різні за тривалістю проміжки часу; пов'язуючи рішення з описами проблем; результати із ситуаціями тощо;
- прецедент фіксує тільки той досвід, який може навчити (бути корисним), прецеденти, що фіксуються, потенційно можуть допомогти фахівцеві (ЛПР) досягти мети, полегшити її формулювання в майбутньому або попередити його про можливу невдачу або непередбачену проблему.

Прецедент дає змогу структурувати одиниці досвіду, при цьому вибір структури залежить від завдань, під час розв'язання яких необхідно забезпечити

повторне використання досвіду. Загальна структура прецеденту включає два основні компоненти:

- ідентифікаційна (характеризує) частина - описує досвід у такий спосіб, який дає змогу оцінити можливість його повторного використання в певній ситуації;
- навчальна частина - описує урок (навчальне знання, рішення) як частину одиниці досвіду, наприклад, розв’язання проблеми або його частина, доказ (висновок) рішення, альтернативні або невдалі рішення.

Наведена структура прецеденту дає змогу уявити процес розв’язання нового завдання з використанням накопиченого досвіду, що зберігається в базі (бібліотеці) прецедентів (рис. 2.6).

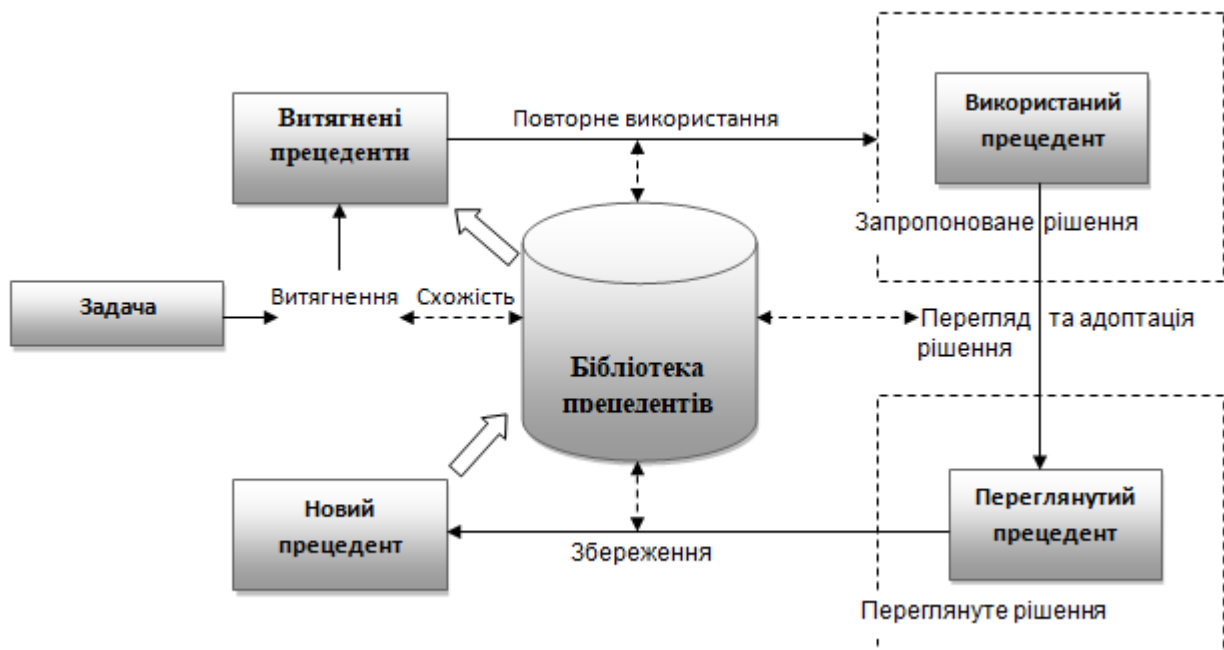


Рисунок 2.6 – Цикл CBR

Розглянемо докладніше кожен з етапів CBR-циклу:

Відновлення (витяг). Розв’язання проблеми починається з подання її користувачем у вигляді нового прецеденту. На стадії відновлення (вилучення) з бази прецедентів вибирають одні або кілька прецедентів, які вважають корисними

для розв’язання нової проблеми. Прецеденти відбирають, ґрунтуючись на подібності (близькості) нової проблеми до проблем, опис яких містять прецеденти з бази прецедентів. Передбачається, що близькі проблеми мають подібні рішення. Точна формалізація подібності дуже сильно впливає на якість роботи CBR системи. Зазначимо, що оцінка подібності є знанням, яке сильно залежить від предметної області. У разі використання складних структур для представлення прецедентів обчислення оцінки подібності може виявитися дуже складною, зокрема і в обчислювальному відношенні. Крім того, при зростанні бази прецедентів зазвичай спостерігається зменшення ефективності вилучення, тому що доводиться аналізувати все більшу кількість прецедентів. Для більш ефективного вилучення пропонується застосовувати індексування прецедентів.

Перелічимо основні типи метрик, які можуть бути використані в задачах пошуку близьких прецедентів [89]: евклідова метрика, міра схожості Хеммінга, імовірнісна міра подібності, міра подібності Роджерса-Танімото, Манхеттенська метрика, відстань Чебишева, метрики Махалобіса, Журавльова, Брея-Кертиса, Чекановського, Жаккара та інші.

Повторне використання. Рішення або інша інформація, використовувана для розв’язання проблеми, що містяться у витягнутих (відновлених) прецедентах повторно використовується для розв’язання нової проблеми. Можливість повторного використання в низці випадків досягається шляхом адаптації (перетворення) відновлених рішень. На даний момент існує кілька методів адаптації для CBR систем [101]. Основна відмінність між різними методами адаптації полягає в застосуванні або трансформаційного, або породжувального підходу.

Трансформаційна адаптація покладається на набір правил адаптації або операторів, які описують те, яким чином відмінності в описі проблеми впливають на зміни в рішенні. Відмінності між новим і витягнутим прецедентом аналізуються, далі відбирається набір застосовних перетворень, і виконуються запропоновані зміни

рішення. Основи цього підходу було сформульовано в роботах із трансформаційної аналогії [101].

Методи породжувальної адаптації для свого ефективного застосування вимагають наявності вирішувача (машини виведення), здатного на підставі узагальнених знань (без використання прецедентів) розв'язати проблему. При цьому використовуються не рішення, але послідовність кроків (порядок) розв'язання проблеми, описані в прецедентах. Ця послідовність використовується для управління (спрямування) вирішувачем.

Стадія повторного використання може бути в обчислювальному відношенні дуже витратною, особливо в разі використання складних структур для представлення прецедентів, у яких розв'язання прецеденту потребує детального опису. Витрати на адаптацію залежать від ступеня подібності (близькості) між відновленим (витягнутим) і новим прецедентами.

Переробка (перегляд, перевірка). Цю стадію дуже слабо відображено в літературі, у більшості випадків це пов'язано з тим, що завдання перегляду (перевірки) рішення зазвичай не є завданням CBR системи, тому що за своєю природою дуже складна і важко формалізована. Рішення, визначене CBR системою, перевіряється і по можливості виправляється людиною - експертом предметної області.

Збереження. На стадії збереження відбувається навчання CBR системи, суть якого полягає в додаванні переглянутого прецеденту до бази прецедентів. Таким чином, досвід розв'язання нової проблеми стає доступним для повторного використання в майбутньому. Однак цей підхід має очевидні недоліки, найбільш важливим з яких є безперервне зростання бази прецедентів, що надалі призводить до зниження ефективності вилучення. Щоб уникнути виникнення цієї проблеми було запропоновано стратегії вибіркового додавання прецедентів у базу прецедентів і вибіркового «забування» вже збережених прецедентів [101].

Деякі етапи в CBR-циклі можуть бути відсутніми і виконуватися експертом або людиною, яка приймає рішення (ЛПР). Це пов'язано з необхідністю при формуванні БП використовувати тільки достовірну інформацію або інформацію, підтверджену експертом. Таким чином, можна мінімізувати кількість прецедентів у БП CBR-системи і підвищити ступінь їх достовірності.

В загальному випадку до прецеденту може включати такі компоненти:

- опис проблеми або проблемної ситуації;
- розв'язання задачі щодо проблемної ситуації та рекомендації ЛПР;
- результат застосування рішення.

Опис результату може включати список виконаних дій, додаткові коментарі та посилання на інші прецеденти. Прецедент може мати як позитивний, так і негативний результат застосування рішення, а також у деяких випадках може наводитися обґрунтування вибору цього рішення та можливі альтернативи.

У більшості випадків для подання прецедентів достатньо простого параметричного представлення у вигляді набору параметрів із конкретними значеннями та рішення [102]:

$$\text{Case}=(x_1, x_2, \dots, x_n, R), \quad (2.8)$$

де $x_1, x_2, \dots, x_n$ - параметри ситуації, що описує даний прецедент ($x_1 \in X_1, x_2 \in X_2, \dots, x_n \in X_n$), n - кількість параметрів R – рішення, $X_1, \dots, X_n$ – області допустимих значень відповідних параметрів.

У деяких випадках такого подання буває недостатньо, оскільки є обмеження, пов'язані з виразними можливостями параметричної моделі подання прецедентів. При параметричному поданні важко забезпечити врахування залежності між параметрами прецеденту (наприклад, причинно-наслідкові).

У CBR циклі може використовуватися не тільки БП, а й узагальнені знання про предметну область для підтримки процесу міркування на основі прецедентів. Ця підтримка може бути слабкою або сильною, а може й бути відсутньою взагалі.

Як правило, у CBR циклі на різних етапах потрібне залучення людини, що приймає рішення (ЛПР). Якщо процес вилучення прецедентів може виконуватися автоматично, то для процесу адаптації та повторного використання прецедентів може знадобитися участь ЛПР.

Формалізоване подання вищесказаного може бути подано таким чином: якщо задано опис деякого об'єкта $s \approx s_j$ та існує близький опис $l_j \approx (s_j, r_j, h_j)$, де l_j - структура прецедента, s_j - опис об'єкта або проблемної ситуації, r_j - опис рішення для заданого об'єкта або проблемної ситуації, h_j - опис результату його застосування та можливих обмежень, то можна стверджувати, що r_j , є схожим (наближеним) рішенням заданого опису об'єкта.

При цьому передбачається послідовне вирішення певного набору завдань із виділенням етапів їх вирішення. Для розроблення функціональної моделі, зробимо структурну декомпозицію ЕС ІППР на різних фазах функціонування. CBR-цикл, що відповідає рис. 2.7, формально представимо як сукупність послідовно виконуваних фаз F_i :

$$CBR = (F_1, F_2, F_3, F_4, F_5, F_6), \quad (2.9)$$

де F_1 – фаза вибору прецедентів, подібних до об'єкта, що ініціюється, на основі оцінки подібності;

F_2 – фаза прийняття рішення в проблемній ситуації шляхом використання витягнутого прецеденту;

F_3 – фаза верифікації та адаптації витягнутого прецеденту;

F_4 – фаза збереження новоприйнятого рішення у сховищі прецедентів;

F_5 – фаза перегляду сховища прецедентів на основі оцінювання якості прецеденту за допомогою синтаксичних або семантичних вимірювань;

F_6 – фаза відновлення (реконструкції) сховища прецедентів.

Для формування структурної та функціональної моделей ЕС ІППР виділимо «входи» та «виходи» для кожної фази.

Фаза 1. «Входом» для цієї фази є об'єкт, що ініціює P зі сховища прецедентів D :

$$(P, D) \rightarrow F_1 \quad (2.10)$$

Особливістю першої фази є використання функції подібності у вигляді ступеня подібності. Результат роботи цієї фази - множина подібних прецедентів S_p , яка своєю чергою є вихідною для другої фази, що формально запишемо у вигляді відображення:

$$F_1(SIM) \xrightarrow{S_p} F_2 \quad (2.11)$$

Фаза 2. Проміжним результатом цієї фази є ранжована множина прецедентів S_r та опорний прецедент P_0 для подальшої адаптації формованого рішення, при цьому використовується функція оцінки релевантності REL .

Кінцевим результатом є сформоване рішення R_s , яке є вихідним для третьої фази. Уявімо зазначені перетворення у вигляді відображення:

$$F_2 \xrightarrow{S_r, P_0} A_2(REL) \xrightarrow{R_s} F_3 \quad (2.12)$$

де A_2 – проміжна частина другої фази.

Фаза 3. Проводиться тестування сформованого рішення на відповідність реальному контексту проблемної ситуації. Зазначимо, що завдання цієї фази вирішують функції корисності з використанням показника важливості, який

характеризує відносний внесок опорного рішення R_s у сформоване рішення R_{result} , яке є вихідним для четвертої фази:

$$F_3(U) \xrightarrow{R_s} F_4 \quad (2.13)$$

Фаза 4. Проводиться внесення в структуру ЕС ІППР змін, пов'язаних із процесом формування нового рішення для проблемної ситуації. З ініціувального об'єкта P та сформованого рішення R_{result} синтезується новий прецедент P_s , він є вихідним для п'ятої фази:

$$F_4(P) \xrightarrow{P_s} F_5 \quad (2.14)$$

Фаза 5. Проводиться оцінювання поточного стану сховища прецедентів для підтримання необхідного рівня якості прецедентної системи, на вході - новий прецедент P_s . Міра якості Q визначається як комплексна функція двох показників:

$$Q = f(C, P), \quad (2.15)$$

де C – оцінка компетентності прецедентної системи, P - оцінка ефективності прецедентної системи. Результатом є оцінка якості СПР S_q :

$$F_5(Q) \xrightarrow{S_q} F_6 \quad (2.16)$$

Фаза 6. Виконується управління сховищем прецедентів. Виконується узагальнення і класифікація, потім перераховуються оцінки якості системи:

$$F_6 \xrightarrow{M} D || (C), (P), (C) \rightarrow (Q), \quad (2.17)$$

де D – сховище прецедентів.

Якщо оцінка якості системи знижується, повертаємось до виконання п'ятої фази, якщо ні - система зупиняється до виявлення наступної проблемної ситуації. На основі наведених вище відображень входів і результатів на кожній фазі будується структурна модель функціонування системи (рис.2.7).

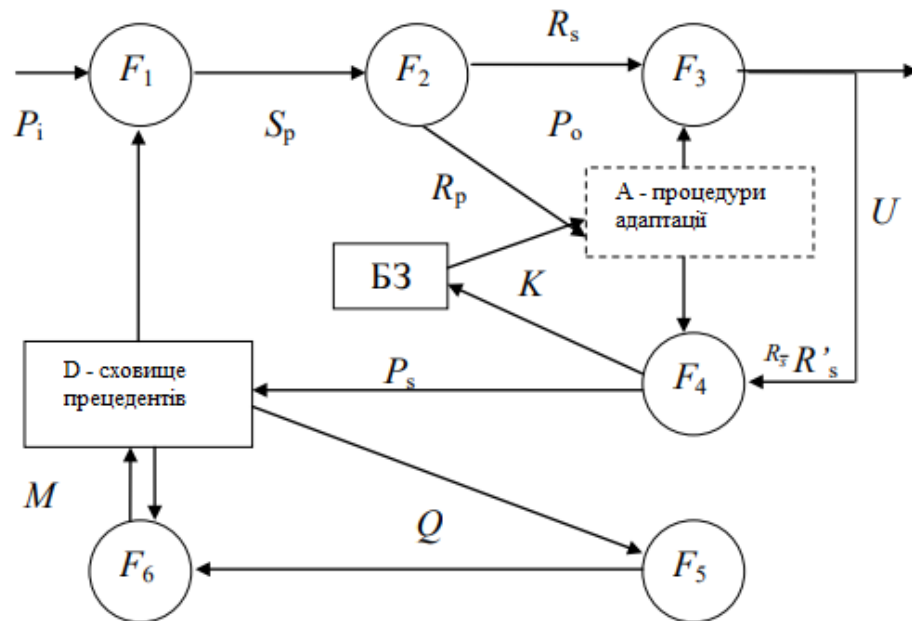


Рисунок 2.7 – Схема розширеного CBR циклу

Основною перевагою CBR-систем є їхня простота і легкість реалізації, що робить прецедентні системи гарним засобом для представлення знань і підтримки ухвалення рішень. Однак, зазначимо головні недоліки прецедентного підходу – це складність урахування динамічних чинників та складність врахування обмежень щодо ухвалення рішень, що задаються множиною цільових факторів, а не станів. Традиційно зазначені недоліки долають або послаблюють, розробляючи гібридні системи, коли об'єднують механізм ухвалення рішень на основі прецедентів із механізмом, що ґрунтується на правилах або обмеженнях, що, своєю чергою, ускладнює ЕС ІППР[102].

Таким чином, для використання прецедентної ЕС ІППР у предметних галузях в умовах неповноти та невизначеності інформації актуальним є розроблення гібридної

ЕС ІППР на основі ситуаційного або сценарного підходу та прецедентної системи. Зазначений клас таких систем називається сценарно-прецедентними [101]. Така комбінація підходів може забезпечити вирішення проблем невизначеності та виникнення чинників суб'єктивності ЛПР.

2.3.4 Методи зниження впливу факторів суб'єктивності та невизначеності на ЛПР

Відповідно до [122], усі невизначеності, що можуть мати місце в ЕС ІППР, можна поділити на три класи: невизначеності, пов'язані з неповнотою знань про проблему, щодо якої ухвалюють рішення, невизначеність, пов'язана з неможливістю точного врахування реакції довкілля на дії ЛПР, невизначеність, зумовлена неточним розумінням своїх дій самою ЛПР. Ступінь невизначеності залежить від: складності ситуації, доступності альтернатив, неясності результатів, отриманих після реалізації рішення, відсутності чітких уявлень про взаємозв'язок можливих рішень, чіткості переваг ЛПР можливим результатам, отриманим після прийняття того чи іншого рішення, низки інших важко обчислювальних чинників.

Запровадити будь-яку міру для оцінювання впливу кожного з перелічених чинників доволі складно. Застосування точних математичних моделей і аналітичних залежностей у цьому випадку часто дає досить грубі, а іноді навіть якісно невірні, результати.

Існує декілька проблем, що становлять складність під час розроблення ЕС ІППР. Перша проблема пов'язана з багатокритеріальністю функції вибору ЛПР і складністю ранжування окремих критеріїв. Існує також проблема сприйняття запропонованого системою рішення ЛПР.

Для подолання вказаних проблем використаємо комбіновану методику формування функції вибору ЛПР, яка заснована на нелінійній згортці окремих критеріїв, які задані за допомогою лінгвістичних змінних. Під час критеріального аналізу ситуації в просторі критеріїв розглянемо два підпростори S та D , які є

підмножинами m - вимірною Евклідового простору, де m є числом критеріїв: $S \in R^m$, $D \in R^m$. S – це підпростір, у якому ЛПР бажано мати значення критеріїв, що характеризують об’єкт після виконання рішення (сценарію дій), D – це підпростір, який визначає за оцінками ЛПР поточний стан об’єкта, щодо якого ухвалюють рішення.

Значимість i -го критерію (його вага) g_i буде деякою функцією від значень i -го критерію в підпросторах S та D , які позначимо як G_i^S та G_i^D :

$$G_i = c_i F_i(G_i^S, G_i^D) \quad (2.18)$$

Функцію F_i можна подати у вигляді абсолютної величини різниці або частки значень G_i^S та G_i^D , c_i – коефіцієнт, що задається на основі експертних оцінок. Отримані значення ваг використовують під час формування узагальненої багатокритеріальної функції уподобань ЛПР, що задається виразом:

$$L_R = G_{1,k} l_{k,R} \oplus G_{2,j} l_{j,R} \oplus \dots \oplus G_{m,n} l_{n,R} \quad (2.19)$$

де L_R це значення функції вибору ЛПР для варіанта рішення R , G_{ij} оцінка ступеня важливості i -го критерію для j -ої лінгвістичної змінної l , яку можна визначити через вираз:

$$l_j = \left(\frac{X_j^R - X_{j,k}^{\min}}{X_{j,k}^{\max} - X_{j,k}^{\min}} \right) + \Theta_{j,k}, \quad (2.20)$$

де X_j^R значення j -го параметра при варіанті рішення R , $X_{j,k}^{\min}$ та $X_{j,k}^{\max}$ це нижні і верхні значення j -го параметра для даної лінгвістичної змінної, $\Theta_{j,k}$ це числове значення k -ї лінгвістичної змінної для j -го параметра.

Запропонований метод дозволяє структурувати вибір рішення, використовуючи лінгвістичні змінні та експертні оцінки. Він дає змогу враховувати не тільки якісні аспекти, а й кількісні характеристики, як-от важливість кожного

критерію та поточний стан проблеми. Це допомагає зробити більш поінформований та обґрунтований вибір.

Проблема сприйняття запропонованого системою рішення ЛПР є такою, яка значною мірою визначає ефективність роботи системи загалом. Для її розв'язання необхідно насамперед урахувати такі характерні особливості діяльності ЛПР:

- поведінка ЛПР цілеспрямована і його рішення орієнтоване на досягнення певної мети;
- ЛПР несе відповідальність за наслідки ухваленого рішення, його своєчасність і якість;
- ЛПР обирає тільки обґрунтовані рішення, які він може змістовно аргументувати;
- ЛПР приймає рішення поетапно включно з коригуванням цілей, завдань і способів впливу на об'єкт;
- ЛПР у процесі розв'язання задачі має працювати з природною, звичною для нього, і широко вживаною в цій галузі змістовною інформацією [122].

Для отримання багатоваріантного сценарію рішення та подолання факторів невизначеності при прийнятті рішення, використання сценарно-прецедентної моделі виявляється важливим і корисним. Ця модель поєднує у собі сценарний підхід, який дозволяє врахувати можливі сценарії подій та їхніх наслідків при прийнятті рішень, з прецедентною системою, що базується на аналізі аналогічних ситуацій з минулого та використанні їхніх рішень.

Такий підхід дозволяє зменшити ступінь суб'єктивності у процесі прийняття рішень шляхом залучення більш об'єктивних критеріїв та знань, які відображаються через аналіз історичних прецедентів та розгляд різних можливих сценаріїв.

Зменшення впливу помилок, допущених ЛПР у даних ним суб'єктивних оцінках, досягається за допомогою використання запропонованого алгоритму вибору рішень (рис.2.8):

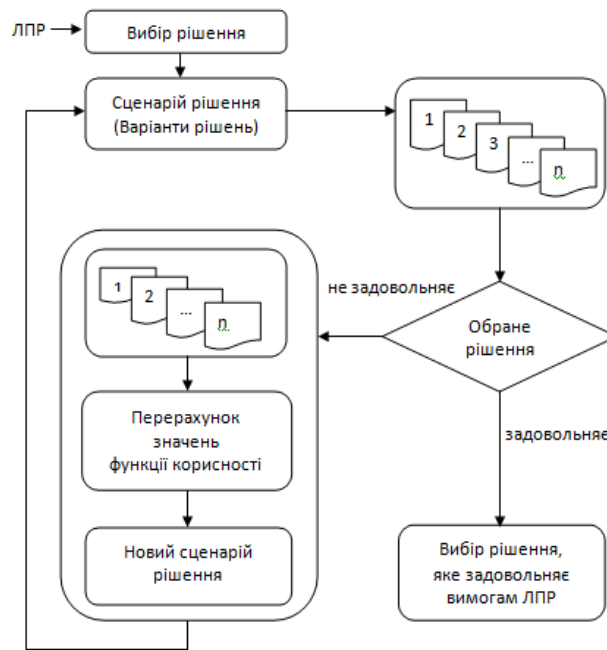


Рисунок 2.8 – Алгоритм вибору рішення ЛПР

1. Система пропонує сценарій рішення, яки надає декілька варіантів рішень.
2. Якщо вибрано один з варіантів рішення, процедуру вибору вважають завершеною, інакше здійснюється перехід до п. 3;
3. Тут здійснюється перерахунок значень функції схожості та отримується новий сценарій рішення та відбувається вибір нового варіанту, що відповідає перевагам ЛПР.
4. Здійснюється збереження отриманого рішення в сховище системи для подальшої обробки.

Практична реалізація запропонованого методу дозволить підвищити ефективність і якість вибору ЛПР релевантного рішення за рахунок підвищення оперативності забезпечення необхідною для прийняття рішень інформацією та зниження впливу суб'єктивних чинників при використанні експертних оцінок.

2.3.5 Архітектура ЕС ІППР

На підставі проведених досліджень запропонуємо архітектуру ЕС ІППР. Вона включає чотири підсистеми:

- підсистему завантаження, перетворення та зберігання даних;
- підсистему аналізу та обробки даних;
- підсистему прийняття рішення;
- підсистему користувацької візуалізації.

Запишемо ланцюжок формальних перетворень у ЕС ІППР:
визначення та формування цілей => ETL процеси => Educational Data Mining => CBR цикл => отримання ЛПР об'єктивного рішення.

Виходячи з вищепованого дослідження, ЕС ІППР має реалізовувати такі функціональні вимоги:

1. Визначення та формування цілей: Початковий етап включає в себе визначення мети прийняття рішення та формування конкретних цілей, які потрібно досягти.

2. ETL процеси: На цьому етапі проводяться процеси збору, трансформації та завантаження даних з різних джерел в систему ЕС ІППР. Ці дані можуть включати інформацію про прецеденти, статистичні дані, експертні знання та іншу важливу інформацію.

3. Educational Data Mining (EDM): Після збору даних використовується процес EDM для аналізу і виявлення корисних знань та патернів в них. Цей етап може включати в себе застосування різних алгоритмів машинного навчання для виявлення взаємозв'язків та закономірностей у даних.

4. CBR цикл: Підхід Case-Based Reasoning (CBR) використовується для прийняття рішень на основі аналізу аналогічних прецедентів. Система порівнює поточну проблему з раніше вирішеними ситуаціями та використовує аналогічні рішення для вирішення поточної задачі.

5. Отримання ЛПР об'єктивного рішення: На заключному етапі система формує об'єктивне рішення на основі аналізу даних, знань та прецедентів, що були зібрані та оброблені. Це рішення може бути підкріплене аналітичними доказами та рекомендаціями для кращого розуміння та прийняття рішення ЛПР.

ЕС ІППР в основному моделюють та інтерпретують дії користувача з організації своїх знань про об'єкт і роблять із них висновки. У певному сенсі ЕС ІППР допомагає комп'ютерній системі застосувати досвід користувача під час ухвалення ефективних рішень.

У зв'язку з тим, що метою дослідження є підвищення ефективності та оптимізація процедури прийняття рішень взагалі та інформаційної підтримки ЛПР в слабоформалізованих навчальних процесах зокрема, для досягнення поставленої мети використання сценарно-прецедентного підходу побудови ЕС ІППР вважається цілком доцільним та актуальним. В цьому разі, загальна схема процесу прийняття рішень, яка реалізована в запропонованій ЕС ІППР буде виглядати наступним чином (рис. 2.9).

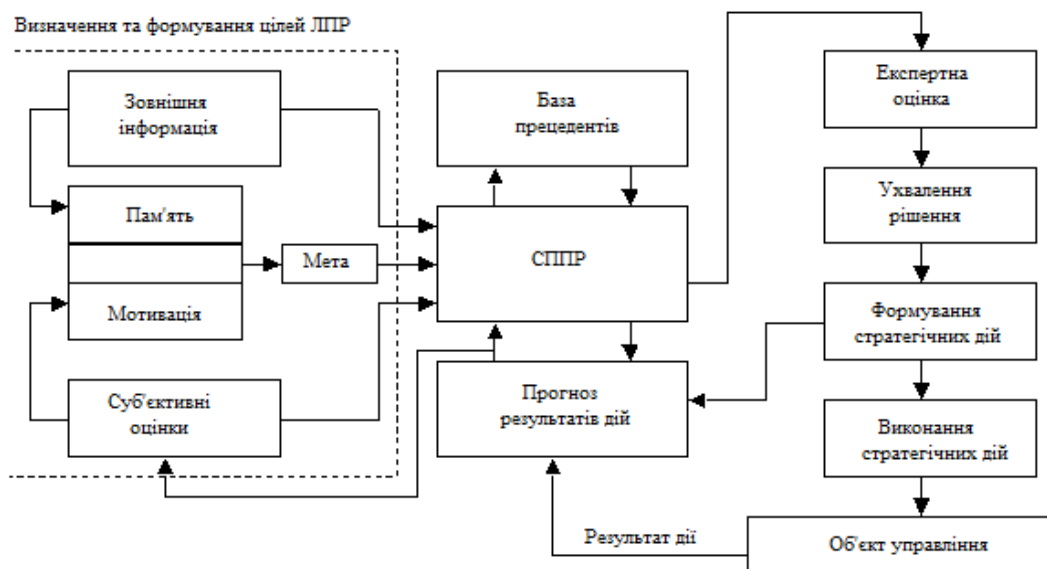


Рисунок 2.9 – Загальна схема процесу прийняття рішень в ЕС ІППР

2.4 Висновки до другого розділу

Зазначене спільне використання експертної системи та інтелектуальної системи підтримки прийняття рішень (ЕС ІППР) створює можливість розробки структурованої схеми підтримки рішення ЛПР, що наочно відображає весь процес прийняття рішень в умовах невизначеності та розпливчатості. Експертний опис об'єктів, які входять у предметну область, сприяв усуненню суперечностей, багатозначності та надмірності інформації. Використання сценарно-прецедентного підходу до побудови систем підтримки такого класу надає можливість динамічної трансформації освітніх даних з неявної форми (у вигляді прецедентів) у явну форму (у вигляді правил прийняття рішень).

Запропоновано підхід, де інтелектуальний аналіз даних використовується як етап CBR циклу для створення бази прецедентів. Це дозволило системі збирати та аналізувати велику кількість даних, витягувати корисну інформацію та створювати базу знань на основі попередніх ситуацій і вирішень. Така база прецедентів стала основою для подальшого прийняття рішень в системі, що покращило її ефективність та точність.

РОЗДІЛ 3

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПІДТРИМКИ ВИБОРУ ІНДИВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ

Для підтвердження доцільності та працездатності пропонованого сценарно-прецедентного підходу до побудови ЕС ІППР вибору навчальної освітньої траєкторії в роботі розглянуто типовий приклад слабоформалізованої ситуації прийняття рішення - вибір наукового керівника і теми дипломного проекту студентів-випускників в контексті підтримки їх індивідуальної освітньої траєкторії.

3.1 Архітектура ЕС ІППР вибору наукового керівника та теми дипломного проекту

Таким чином, спираючись на вище вказану архітектуру (підрозділ 2.3.5) та загальну схему прийняття рішення в ЕС ІППР (рис. 2.8) побудуємо архітектуру ЕС ІППР вибору наукового керівника та теми дипломного проекту :

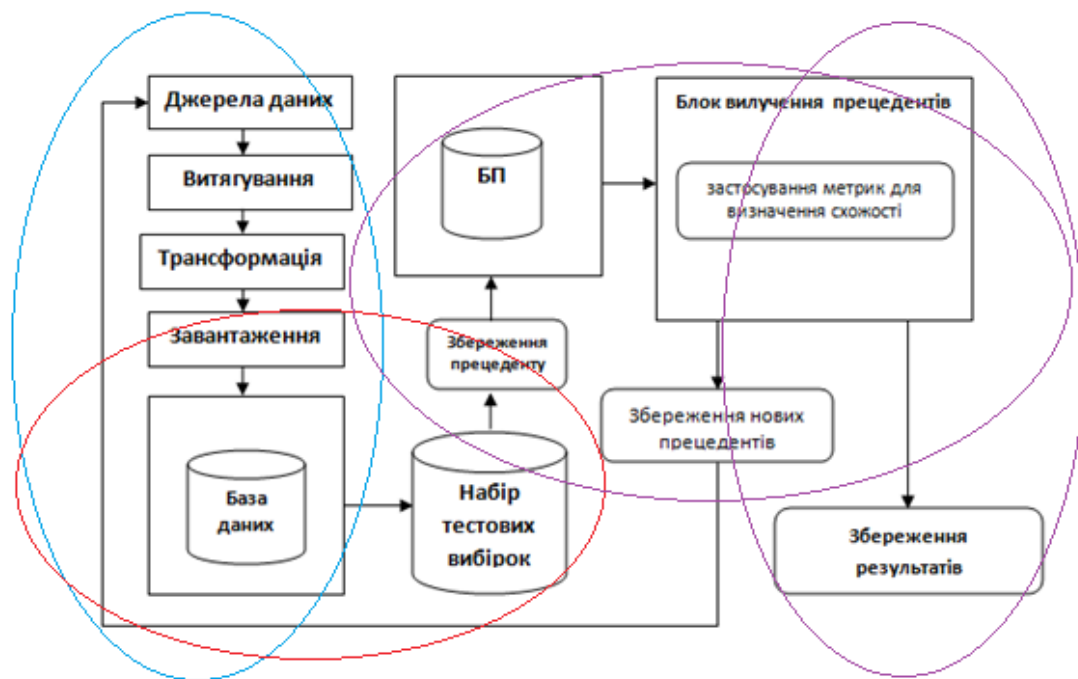


Рисунок 3.1 – Структурна модель процесу функціонування ЕС ІППР

Структурна модель процесу функціонування ЕС ІППР (рис.3.1) являє собою структуру з чотирьох підсистем, що перетинаються:

- підсистема автоматизації завантаження освітніх даних;
- підсистема аналітичного аналізу освітніх даних;
- підсистема підтримки рішення ЛПР на основі розширеного CBR циклу;
- підсистему візуалізації прийнятого рішення ЛПР.

Виходячи з вищеописаного дослідження, ЕС ІППР має реалізовувати такі функціональні вимоги:

- використання різнотипних джерел даних (навчальні плани, відомості успішності, бази даних, інформаційні бази облікових систем та ін.) для початкового завантаження і подальшого поповнення інформаційного сховища даних за допомогою віще вказаних ETL процесів обробки освітніх даних;
- розв’язання завдань аналітичної обробки даних за допомогою інтелектуального аналізу освітніх даних;
- використання попереднього досвіду вирішення завдань управління за допомогою використання прецедентного підходу;
- зниження факторів впливу невизначеності отриманої інформації за допомогою сценарно-прецедентного підходу;
- візуалізацію отриманого рішення для подальшого опрацювання та застосування.

3.2 Застосування ETL процесів для автоматизації завантаження освітніх даних

Навіть якщо робота освітнього закладу керується єдиною інформаційною системою, що зберігає свою інформацію в реляційній базі даних (такі бази називаються оперативними), у більшості випадків подібні системи не годяться для

надання аналітичної інформації, оскільки оперативні системи і сховища даних працюють за різними принципами. Оперативні системи містять поточну інформацію, наприклад, список дисциплін за вибором, що вивчає студент. Сховище даних містить історичну інформацію, тобто в наведеному прикладі список всіх дисциплін за вибором. Стан оперативної системи весь час змінюється, у ній відбувається велика кількість невеликих транзакцій, наприклад, заноситься інформація про те, що студент обрав іншу дисципліну для вивчення. Інформація у сховищі залишається незмінною і лише поповнюється новими даними за певним розкладом. Оперативні системи лежать в основі роботи навчального закладу, тоді як сховища даних допомагають відповісти на запитання, як функціонує навчальний заклад і використовуються під час розроблення стратегій, спрямованих на підвищення ефективності управління освітою.

Оскільки перед оперативними системами і сховищами даних ставлять різні завдання, архітектури їх також різняться. Під час побудови сховища зазвичай використовують багатовимірну модель даних [113]. За такого підходу інформація розбивається на два класи: факти і вимірювання. Факти - це числові характеристики, що позначають деяку подію, які завжди оточені текстовим контекстом - вимірами.

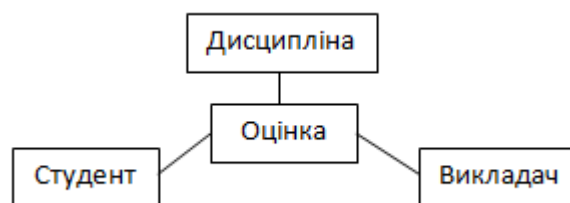


Рисунок 3.2 – Багатовимірна модель даних

В такому разі етапи збору, аналізу та обробки інформації можна представити наступним чином:

Витягування (Extract). Збір даних таких як, оцінки студентів, відвідуваність, результати опитувань, даних з електронних навчальних платформ, адміністративні

дані тощо. Можлива інтеграція даних з різних шкіл або університетів, для дослідницьких проектів або для створення регіональних або національних освітніх звітів.

Трансформація (Transform). Очищення даних складається з видалення помилок, таких як неправильні оцінки або відсутні дані. Нормалізація даних забезпечує консистентності даних, наприклад, уніфікація формату дат або оцінок. Агрегація полягає у створенні підсумкових звітів за класами, курсами або викладачами. Підготовка даних для специфічного аналізу, наприклад, для вивчення зв'язку між відвідуваністю та успішністю.

Завантаження (Load). Створення єдиного репозиторію даних - завантаження оброблених даних у централізовану базу даних для зберігання та аналізу та/або завантаження в інші системи для подальшого використання.

Перше завдання, яке необхідно було вирішити, - це автоматизація трудомісткого процесу обробки та завантаження зібраної інформації. Основну складність становив той факт, що більшість вищих навчальних закладів мають вкрай низький рівень автоматизації діяльності, практично відсутні централізовані бази даних, дані в наявних базах важко формалізуються. Також труднощі виникали через те, що більшість операцій у вищих навчальних закладах обробляється вручну, оскільки процес потребує застосування творчості.

В даному дослідженні це завдання було та реалізовано за допомогою MS Excel та спеціальної надбудови від розробників системи управління реляційними базами даних Microsoft SQL Server, що дає змогу виявляти неявні закономірності в наборах даних. Для вирішення завдання необхідно завантажити надбудову Microsoft SQL Server Master Data Services для Microsoft Excel. Надбудова SQLServerDMXLAAddIn поширюється вільно і доступна на сайті компанії Microsoft. Встановлення надбудови SQLServerDMXLAAddIn ніяких складнощів не викликає і не потребує опису.

Таким чином, стоїть завдання розробити алгоритм ETL-процесу, що перетворює Excel-звіти в данні, що будуть завантажуватися далі в ЕС ІППР (рис.3.3).



Рисунок 3.3 – Схема обробки та завантаження даних

Для реалізації ETL процесів в даному дослідженні було розроблено процедури, що зберігаються, мовою C# у середовищі Visual Studio Windows Forms (Додаток А). ЛПР викликає процедуру завантаження даних для кожного заповненого даними шаблону (рис.3.3). За рахунок використання Excel шаблонів зменшується трудомісткість наповнення експертної системи даними з боку ЛПР.

Після реалізації ETL процесів база знань являє собою велику інформаційну базу анкетних даних викладачів та студентів, також інформацію з педагогічного навантаження та відомості успішності студентів. Данні представлено у вигляді таблиць Excel. Кожен рядок у таблиці відповідає одному об'єкту класифікації. Будь-який об'єкт характеризується якісними ознаками: прізвище викладача, запропонована тема дипломної роботи, дисципліни для вивчення, галузь наукових інтересів. Вибірки такого типу є вхідними даними для створення прецедентів.

Далі відбувається інтелектуальний аналіз освітніх даних для створення бази прецедентів з подальшим застосуванням в генерації рішення ЛПР.

3.3 Інтелектуальний аналіз освітніх даних як складова CBR циклу

Ефективність роботи з аналізу освітніх даних багато в чому залежить від представлення об'єктів, що аналізуються, у вигляді системи показників.

Як відмічалось вище для вищих навчальних закладів характерно те, що більша частина інформації про студентів міститься в численних анкетах, які вони заповнюють протягом усього навчання, файлах документообігу навчального закладу, файлах ЄДБО, навчальних планів, оціночних відомостей та інших. Розрізненість інформації не дає змоги ефективно вистроювати індивідуальну освітню траєкторію.

З урахуванням різноманітності форм подання даних, використовуваних алгоритмів і сфер застосування, EDM проводиться за допомогою електронних таблиці та різного роду надбудови над ними.

Традиційні БД складаються із сукупності записів-рядків, що містять різні поля (зокрема, відомості про студентів та викладачів). Саме ці поля і являють собою аналізовані ознаки, які ЛПР використовують для виявлення залежностей, оцінки і формування відповідного рішення застосування даних. Під час EDM досліджується безліч об'єктів або варіантів і встановлюється або не встановлюється залежність між результуючим показником і факторами, що впливають на нього. Такий підхід може бути використаний і для вирішення нетривіальних завдань. Запропоноване рішення дасть змогу здійснювати всебічний аналіз студентів завдяки тому, що проводиться консолідація інформації в Excel - файли і може дозволити здійснювати комплексний аналіз інформації з різних джерел.

Характеристиками студентів є місце проживання, стать, рік народження, відомості про попередню освіту, форма навчання, спеціальність, рік навчання, курс, група, дисципліни, що вивчаються в рамках основного плану навчання, дисципліни за вибором, оцінки попередніх курсів навчання, участь у наукових дослідженнях, сфера інтересів тощо. Саме за цими полями здійснюється пошук закономірностей і робиться спроба визначити чинники, які явно або неявно сприяють вибору

індивідуальної траєкторії навчання. Уся ця інформація необхідна для формування сценаріїв майбутніх ухвалень рішень.

В рамках даного дослідження запропоновано розглядати інтелектуальний аналіз освітніх даних як складову CBR циклу.

Цикл CBR це ітеративний процес аналізу та розв'язання проблем, що базується на знаннях, отриманих з аналізу подібних ситуацій з минулого (прецедентів):

1. Формування бази прецедентів: Система використовує очищені та структуровані дані для формування бази прецедентів. Ця база містить попередні випадки, які використовуються для порівняння та аналізу у нових ситуаціях.

2. Використання методів та метрик: За допомогою методів та метрик порівняння нових ситуацій зі збереженими прецедентами отримуються рекомендації для прийняття найкращих рішень або рекомендацій, що закінчує цикл.

На етапі формування тандему проєкту, деякі характеристики можуть уточнюватися в міру деталізації, інші характеристики можуть визначатися з різним, уточненим значенням у результаті використання, CBR-цикл має отримати розвиток – до відповідних фаз додано фази оновлення бази прецедентів та ускладнення структури прецеденту.

3. Оновлення бази прецедентів: Після прийняття рішень або рекомендацій, нові дані та ситуації використовуються для оновлення та покращення бази прецедентів.

4. Ускладнення структури прецедентів: На новому рівні цикл трансформації знань може спричинити ускладнення структури прецедентів, можливе введення ієрархічної або складнішої логічної форми.

Концепція CBR циклу заснована на припущеннях про статичність опису прецеденту і незалежність прецедентів і відповідно проблемних ситуацій один від одного.

Таким чином, розглянемо перший етап запропонованого CBR циклу.

Проблема подання прецеденту - насамперед проблема вибору інформації, яку треба включати в опис прецедентів, знаходження відповідної структури для опису його змісту, а також визначення, яким чином має бути організована база знань прецедентів для ефективного пошуку та багаторазового використання.

Поставлене в дослідженні завдання можна звести до завдання класифікації. Завдання класифікації даних формулюється таким чином. Для опису об'єкта будемо використовувати ознаки $X_1, X_2, \dots, X_n$ параметричного опису прецедента. Кожен об'єкт (прецедент) $C \in PR$ та характеризується набором конкретних значень цих ознак (атрибутів або параметрів) $c = \{x_1, x_2, \dots, x_n\}$, де x_i - значення i -ї ознаки.

Розглянемо окремі види ознак. Вхідний вектор X містить компоненти x_i , звані далі ознаками, або атрибутами об'єкта. Значення, які можуть набувати ознаки об'єкта, належать до трьох основних типів: кількісні або числові, категоріальні та шкальовані [123]. Те, яких значень набувають ознаки, може мати великий вплив на процес розв'язання задачі.

Розглянемо завдання навчальної вибірки (початковий набір прецедентів) з категоріальним типом ознак (текстовий набір даних).

Особливість категоріальних ознак полягає в тому, що це елементи деякої неупорядкованої множини, і не можна говорити, що якесь значення більше або менше за інше. Можна тільки порівнювати їх.

Для вирішення цього завдання розглядаються такі підзадачі як:

- відбір кандидатів в наукові керівники;
- оцінка рівня компетентності кандидатів;
- особисті характеристики.

Відбір кандидатів пропонується проводити на основі аналізу їх досвіду у минулих навчальних роках близького за характером (змістом) до запланованих тем дипломних проєктів. При цьому на відміну від деяких відомих методик до критеріїв оцінки кандидатів необхідно включати критерії, що характеризують не тільки тематику, а й особисті характеристики та їх загальні компетенції. Тоді під час

планування наукового керівництва проводиться пошук схожої тематики, і визначаються близькі за змістом роботи, формується список за ступенем близькості робіт, що виконувалися раніше. Таким чином, формується склад викладачів, які можуть розглядатися як кандидати для включення до складу наукового керівництва дипломними проєктами.

Множина ознак прецедента була означена як x_1 – прізвище викладача, x_2 – пропонуємо тема дипломного проєкта, x_3 – рекомендовані дисципліни для вивчення, x_4 – галузь наукових інтересів викладача, x_5 – характеристика викладача, чи викладає рекомендовану дисципліну, x_6 – чи має досвід роботи зі студентами, які не встигають (мають низький загальний рівень знань).

За аналогічним алгоритмом створюється набір даних студента, після навчання, що є описом поточної ситуації: x_1 – прізвище студента, x_2 – бажана тема дипломного проєкта, x_3 – загальний рівень знань студента (виражений категоріальними ознаками: «незадовільний», «задовільний», «добрий», «відмінний»), x_4 – попередній досвід наукових досліджень, x_5 – галузь наукових інтересів для подальшого вдосконалення професійної компетенції.

Параметр « x_3 », який описує рівень знань студента, в Excel-файлі бази знань представлено у вигляді числових оцінок з предметів. Для ефективного використання цього параметра в прототипі програми та з метою спрощення аналізу, параметр « x_3 » було перетворено з числового признаку на категоріальний за допомогою умовного відображення в Excel. Це було зроблено з використанням функції =ЕСЛИ(), що дозволяє класифікувати числові оцінки у категорії відповідно до їхнього значення.

Формула, використана в Excel, має такий вигляд:

=ЕСЛИ(V2<3;"Незадовільний"; ЕСЛИ(И(V2>=3; V2<3,8);"Задовільний"; ЕСЛИ(И(V2>=3,8; V2<4,5);"Добрий"; ЕСЛИ(V2>=4,5;"Відмінний"; ""))))

Функція =ЕСЛИ() в цій формулі використовується для визначення категорії рівня знань на основі числової оцінки:

«Незадовільний» для оцінок менше 3;

«Задовільний» для оцінок від 3 до 3,8 (включно);

«Добрий» для оцінок від 3,8 до 4,5 (включно);

«Відмінний» для оцінок 4,5 та вище.

Такий підхід до перетворення даних сприяє більш зручному та наочному представленню інформації в прототипі програми, дозволяючи легко класифікувати та аналізувати рівень знань студентів.

Інтелектуальний аналіз освітніх даних, як складова CBR циклу, виглядає наступним чином:

1. Створення тестових вибірок.

Нехай прецедент C описується набором параметрів $C=\{x_1, x_2, \dots, x_n\}$, де кожен параметр x_i кореспондує з певним типом даних, які можна знайти в базі даних. Для кожного параметра x_i існує функція визначення відповідності f_i , яка бере значення з бази даних і перевіряє, чи воно відповідає даному параметру прецедента.

Формування параметрів прецедента: для кожного параметра x_i прецедента C виконується наступний процес:

- з БД отримується набір даних $D_i=\{d_{i1}, d_{i2}, \dots, d_{im}\}$, який можливо використовувати для параметра x_i ;
- застосовується функція f_i для кожного елемента d_{ij} із набору D_i , де $j=1, 2, \dots, m$. Функція $f_i(d_{ij})$ повертає значення, яке вказує на ступінь відповідності між d_{ij} та x_i .

Створення тестової вибірки:

- в результаті вищеописаного процесу для кожного параметру x_i утворюється набір значень відповідності, які визначаються як $F_i=\{f_i(d_{i1}), f_i(d_{i2}), \dots, f_i(d_{im})\}$;
- тестова вибірка для прецедента C формується, як сукупність всіх наборів F_i для кожного параметру $F=\{F_1, F_2, \dots, F_n\}$.

Таким чином, тестова вибірка F представляє собою зібрані та оброблені дані з бази даних, які відповідають параметрам прецедента C і можуть бути використані для подальшого аналізу.

Нехай поточна ситуація Q описується набором параметрів $Q=\{q_1, q_2, \dots, q_n\}$, де кожен параметр q_i кореспондує з певним типом даних, які можна знайти в базі даних. Для кожного параметра q_i існує функція визначення відповідності g_i , яка бере значення з бази даних і перевіряє, чи воно відповідає даному параметру поточної ситуації.

Формування параметрів поточної ситуації: для кожного параметра q_i поточної ситуації Q виконується на ступний процес:

- з БД отримується набір даних $D_i=\{d_{i1}, d_{i2}, \dots, d_{im}\}$, який можливо використовувати для параметра q_i ;
- застосовується функція g_i для кожного елемента d_{ij} із набору D_i , де $j=1, 2, \dots, m$. Функція $g_i(d_{ij})$ повертає значення, яке вказує на ступінь відповідності між d_{ij} та q_i .

Створення тестової вибірки:

- в результаті вищеописаного процесу для кожного параметру q_i утворюється набір значень відповідності, які визначаються як $G_i=\{g_i(d_{i1}), g_i(d_{i2}), \dots, g_i(d_{im})\}$;
- тестова вибірка для параметрів поточної ситуації Q формується, як сукупність всіх наборів G_i для кожного параметру $G=\{G_1, G_2, \dots, G_n\}$.

Таким чином, тестова вибірка G представляє собою зібрані та оброблені дані з бази даних, які відповідають параметрам поточної ситуації Q і можуть бути використані для подальшого аналізу.

2. Створення бази прецедентів та бази опису параметрів поточної ситуації.

Нехай маємо тестову вибірку $T=\{t_1, t_2, \dots, t_n\}$, де кожен t_i представляє собою набір даних, що описує потенційний прецедент:

- формування бази прецедентів: проведемо ініціалізацію порожнього списку або колекції для зберігання унікальних прецедентів $C=\{\}$;
- для кожного елемента t_i в тестовій вибірці T перевіряємо, чи елемент t_i вже існує в C за допомогою функції перевірки $f_{exist}(t_i, C)$, яка повертає істину (*true*), якщо t_i вже присутній у C . Якщо $f_{exist}(t_i, C)$ повертає хибу (*false*), додаємо t_i до C ;

– для збереження бази прецедентів в ресурсний файл використовуємо відповідні інструменти C# для серіалізації колекції C у ресурсний файл.

Вище вказаний процес може бути представлений як ітераційний алгоритм (рис.3.4), де для кожного елемента тестової вибірки виконується перевірка на унікальність i , у разі необхідності, відбувається його додавання до бази прецедентів.

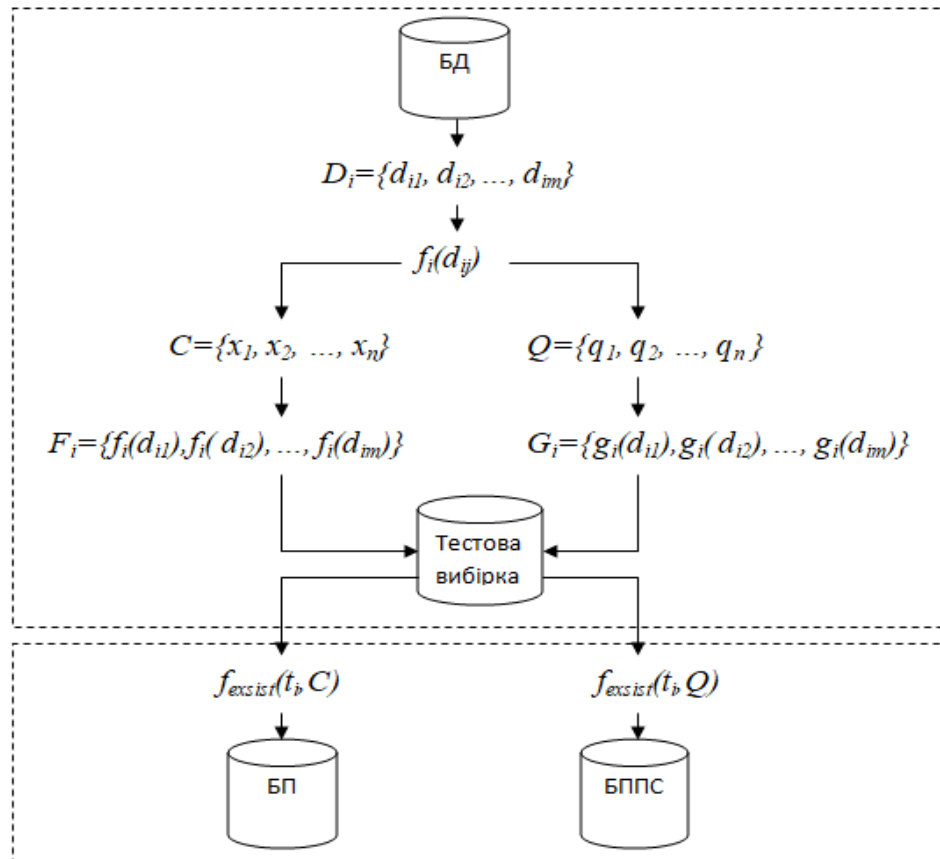


Рисунок 3.4 – Алгоритм виконання першого етапу CBR циклу

Для створення опису поточної ситуації Q застосовується ітераційний алгоритм, аналогічний тому, що використовується для створення бази прецедентів. Процес включає перевірку кожного елемента тестової вибірки на унікальність i , за необхідності, його інтеграцію в колекцію Q , яка представляє поточну ситуацію: проводимо ініціалізацію порожньої колекції Q , яка буде використовуватися для зберігання унікальних даних поточної ситуації, для кожного елемента в тестовій вибірці проводиться перевірка на унікальність в межах колекції Q та, якщо елемент

визнано унікальним (тобто він відсутній в Q), він додається до колекції, після завершення ітераційного процесу заповнення Q , колекція зберігається в БППС. Це дозволяє використовувати зібрані дані в майбутньому для аналізу, порівняння або як вхідні дані для інших процесів.

Таким чином, структура модулю ЕС ІППР, який призначено для зберігання та пошуку інформації про данні минулого навчального року можна представити в загальному вигляді (рис. 3.5).

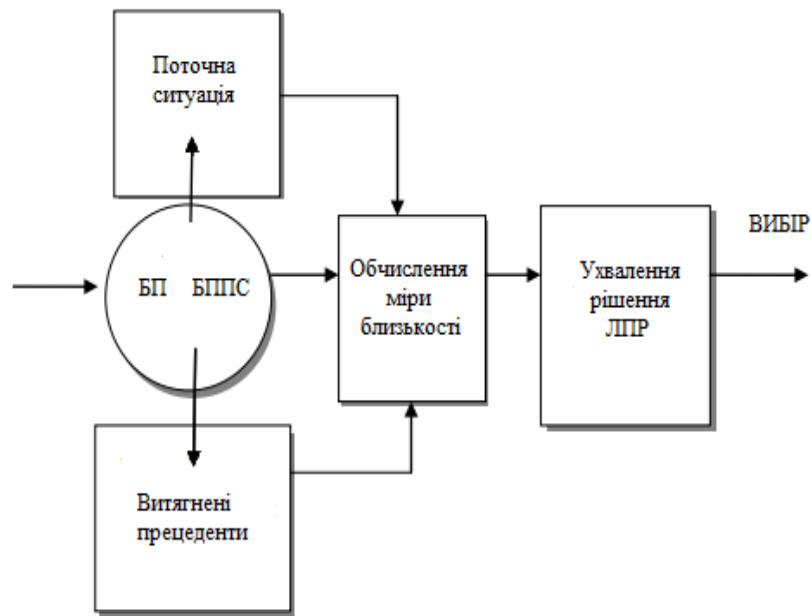


Рисунок 3.5 – Модуль пошуку інформації на основі попереднього досвіду

ЕС ІППР у такому разі може бути розділена на дві підсистеми, що взаємодіють – підсистему пошуку і підсистему адаптації [102].

Відібрані в такий спосіб викладачі та студенти є базою для подальшої диференційованої оцінки - ступеня близькості порівнюваних об'єктів.

3.4 Методи та метрики визначення схожості прецедентів

Другий етап CBR циклу (рис.3.5) – обчислення міри близькості з використанням обраних методів та метрик.

У роботі для подання прецедентів було використано просте параметричне подання, тобто подання прецеденту у вигляді набору параметрів із конкретними значеннями і рішення (набір параметрів, що характеризують запит користувача і відповідь на цей запит).

Для успішної реалізації міркувань на основі прецедентів необхідно забезпечити коректне вилучення прецедентів із БП системи.

Існують різні методи вилучення прецедентів із БП системи [124]: метод найближчого сусіда та його модифікації; метод пошуку на основі дерев рішень; метод вилучення на основі знань; метод вилучення з урахуванням застосовності прецедентів.

Метод, за допомогою якого здійснюватиметься обчислення міри подібності (схожості) прецедентів, задається під час створення CBR-системи розробниками. Найбільш популярним і часто використовуваним методом є пошук найближчого сусіда, в основі якого лежить спосіб вимірювання ступеня збігу значень атрибутів (властивостей), що визначають прецедент [124].

Метод найближчого сусіда (NN – Nearest Neighbor) – це найпоширеніший метод порівняння і вилучення прецедентів. Метод визначення найближчого сусіда (найближчих сусідів) також широко застосовується для розв’язання задач класифікації, кластеризації, регресії та розпізнавання образів [124].

Основними перевагами цього методу є простота реалізації та універсальність у сенсі незалежності від специфіки конкретної проблемної області. З метою визначення ступеня подібності на множині параметрів, використовуваних для опису прецедентів і поточної ситуації, вводиться певна метрика. Далі відповідно до обраної метрики визначається відстань від цільової точки, що відповідає поточній

проблемної ситуації, до точок, що представляють прецеденти з БП, і вибирається найближча до цільової точка.

Безумовно, ефективність методу найближчого сусіда багато в чому залежить від вибору метрики, тобто міри подібності. Основні метрики, які використовуються в методі найближчого сусіда [125]: евклідова відстань; міра подібності в метриці Мінковського; зважені міри подібності; відстань Хеммінга; відстань між неупорядкованими наборами; відстань Левенштейна; міра подібності Танімото.

Вибір відповідної метрики творче і доволі трудомістке завдання, від успішного вирішення якого безпосередньо залежить результативність пошуку та вилучення прецедентів. У кожному конкретному випадку цей вибір здійснюють по-різному, залежно від цілей користувача, фізичної та статистичної природи використовуваної інформації та інших обмежень і чинників, що впливають на процес інтелектуального пошуку. У деяких методах вибір відповідної метрики досягається за допомогою спеціальних алгоритмів перетворення вихідного простору ознак, в інших - експерт (адміністратор) сам визначає метрику, спираючись на власні знання про предметну область або експериментальні дані.

В контексті даного дослідження розглянемо визначення найближчого сусіда на прикладі індексу Жаккара та його модифікації коефіцієнта Танімото.

Метод (індекс) Жаккара і коефіцієнт Танімото є важливими інструментами в аналізі даних, зокрема у задачах, пов'язаних із порівнянням та класифікацією об'єктів.

Метод Жаккара, також відомий як індекс Жаккара, був розроблений французьким ботаніком Полем Жаккаром на початку 20-го століття. Цей метод використовується для вимірювання схожості між двома множинами шляхом порівняння їх спільних і загальних елементів.

Коефіцієнт Танімото, іноді називається коефіцієнтом Жаккара - Танімото, є розширенням індексу Жаккара. Розроблений Т. Танімото в 1950-х, але з розвитком інформаційних технологій метод почали використовувати в рекомендаційних

системах для аналізу схожості між інтересами користувачів або предметами, в текстовому аналізі при порівнянні документів або текстових фрагментів на основі використаних слів або фраз.

Методи Жаккара і Танімото допомагають виявляти структурні зв'язки між даними, що можуть бути неочевидними при поверхневому аналізі. Вони забезпечують кількісні міри схожості, які можуть бути використані для визначення ступеня збігу між різними об'єктами або сутностями, підвищуючи ефективність аналітичних рішень. Зазначимо, що міра подібності Танімото своїм походженням зобов'язана задачі порівняння множин.

Формула індексу Жаккара визначається наступним чином:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (3.1)$$

де:

A і B - це дві множини, які порівнюються;

$|A \cap B|$ - кількість елементів, які є спільними для обох множин (перетин множин);

$|A \cup B|$ - кількість унікальних елементів в обох множинах разом (об'єднання множин).

Індекс Жаккара вимірює схожість між двома множинами, вираховуючи відношення кількості елементів у їхньому перетині до кількості елементів у їхньому об'єднанні. Індекс Жаккара варіюється від 0 до 1, де 0 означає повну відмінність множин, а 1 - їхню ідентичність.

Коефіцієнт Танімото для бінарних векторів визначається так:

$$T(A, B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}, \quad (3.2)$$

де:

A і B - це бінарні вектори (наприклад, вектори характеристик об'єктів);

$|A \cap B|$ - кількість бітів, які одночасно мають значення 1 в обох векторах;

$|A|+|B|$ - загальна кількість бітів зі значенням 1 в кожному векторі.

Коефіцієнт Танімото також вимірює схожість, але він особливо корисний для бінарних даних. Він відображає відношення спільних характеристик до суми всіх унікальних характеристик обох об'єктів. Коефіцієнт Танімото також варіюється від 0 до 1, де 0 означає відсутність спільних характеристик, а 1 - повну ідентичність характеристик.

В даному дослідженні для оцінки близькості поточної ситуації і прецедента використовується індекс Жаккара - Танімото, який допомагає визначити міру схожості між двома множинами: множиною прецедентів та множиною поточних (проблемних) ситуацій.

Нехай поточна ситуація Q представлена набором атрибутів $Q=\{q_1, q_2, \dots, q_n\}$, де кожен q_i є бінарним атрибутом (наприклад, відсутній або присутній). Аналогічно, кожен прецедент C у базі прецедентів представлений як $C=\{c_1, c_2, \dots, c_n\}$, де c_i також є бінарним атрибутом.

Коефіцієнт Танімото, що вимірює схожість між Q і C , визначається як:

$$T(Q, C) = \frac{\sum_{i=1}^n \min(q_i, c_i)}{\sum_{i=1}^n \max(q_i, c_i)},$$

де:

$\sum_{i=1}^n \min(q_i, c_i)$ представляє кількість атрибутів, де відповідні значення в Q і C співпадають (обидва атрибути присутні).

$\sum_{i=1}^n \max(q_i, c_i)$ визначає загальну кількість унікальних атрибутів в Q і C (принаймні один з атрибутів присутній).

Отримане значення коефіцієнту Танімото вказує на ступінь схожості між поточною ситуацією і прецедентом. Значення близьке до 1 вказує на високу схожість, тоді як значення близьке до 0 свідчить про низьку схожість.

У результаті застосування цієї процедури отримаємо множину прецедентів, кожному з яких зіставлена оцінка схожості з поточною ситуацією, яка виражена у відсотках. Виходячи з цих даних, ЛПР може вибрати найбільш підходящий прецедент і отримати рішення для поточної ситуації.

Таким чином, сформовано ієрархічну структурно-параметричну модель функціонування ЕС ІППР, яка являє собою послідовність і взаємозв'язок процесів завантаження даних в систему та фаз CBR-циклу, спрямованих на реалізацію основної мети системи - вибору найкращого рішення зі структурованої бази прецедентів.

3.5 Приклад використання прототипу ЕС ІППР

Роботу запропонованого прототипу ЕС ІППР розглянемо на прикладі розв'язання задачі отримання експертної рекомендації з підтримки вибору ЛПР наукового керівника та теми дипломного проєкта для студента-випускника (рис.3.6).

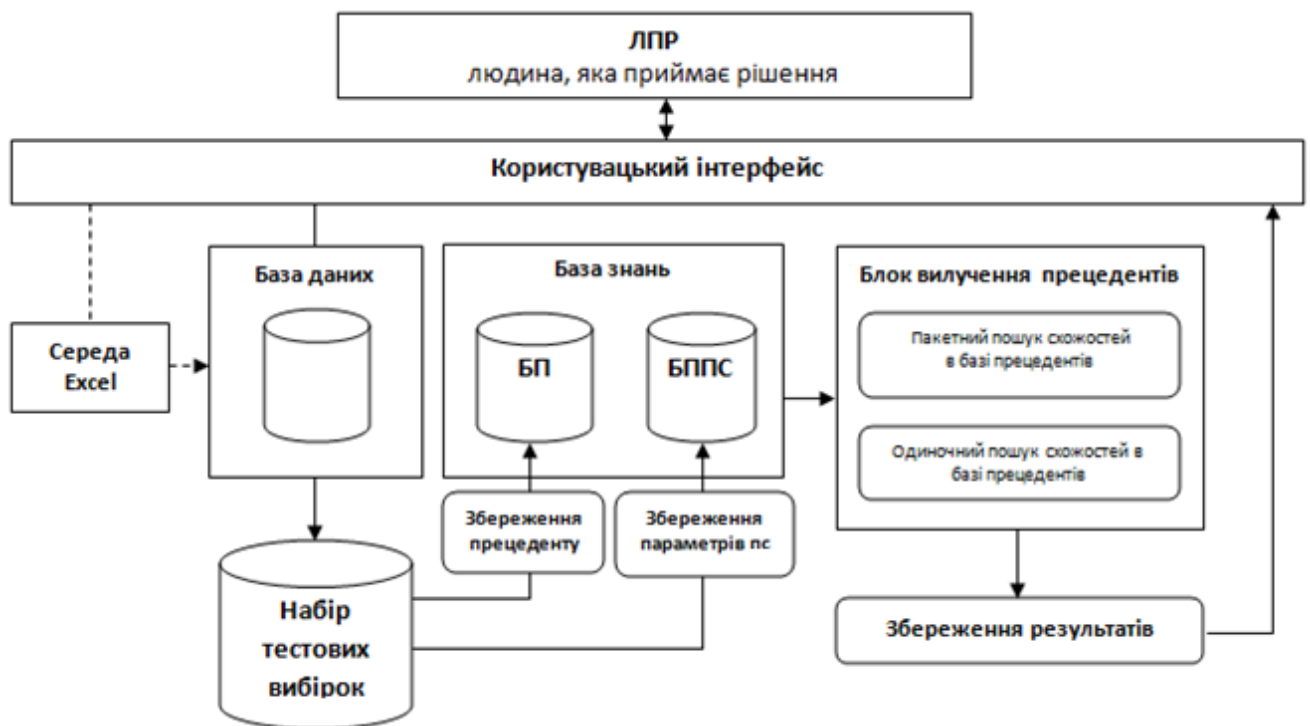


Рисунок 3.6 – Архітектура прототипу ЕС ІППР

1. ЛПР взаємодіє з системою, використовуючи інструменти для завантаження, нормалізації даних, створення та оцінки прецедентів.

Користувач системи завантажує необхідні Excel-шаблони задля підготовки даних для завантаження в ЕС ІППР (рис.3.7).

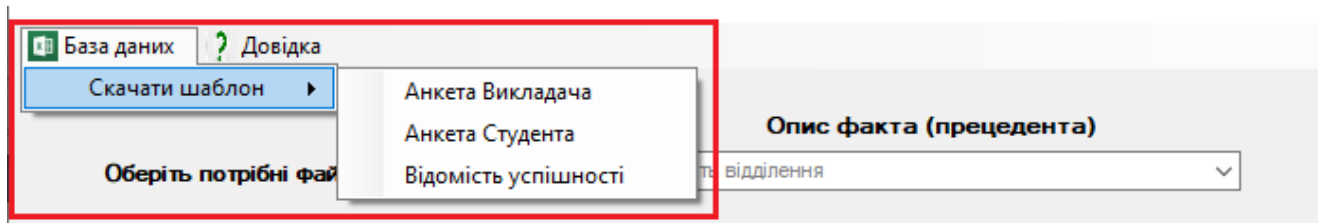


Рисунок 3.7 – Завантаження шаблонів Excel-файлів бази даних

2. Підготовлені Excel-звіти завантажуються для подальшого створення набору тестових вибірок, які будуть використані для створення прецедентів та опису параметрів поточної ситуації.

Далі, на основі тестових вибірок створюється база прецедентів (БП) та база параметрів поточної ситуації (БПС), де зберігається інформація, необхідна для аналізу, порівняння та вилучення прецедентів (рис.3.8).

Рисунок 3.8 – Створення бази прецедентів та параметрів поточної ситуації

Для полегшення роботи ЛПР, було автоматизовано вибір параметрів прецедента з бази знань відповідною функцією на мові С# (додаток А).

За допомогою мови С# створюється функція обробки та додавання прецедента до бази прецедентів (додаток А) із збереженням у ресурсний файл додатку у форматі .xlsx (додаток А). Також створена функція перевірки створеного прецедента на вже існуючий в базі прецедентів прецедент (додаток А).

За допомогою мови С# створюється функція обробки та додавання опису поточної ситуації для кожного студента (додаток А) із збереженням у ресурсний файл додатку у форматі .xlsx (додаток А). Також створена функція перевірки поточної ситуації на вже існуючий запис у файлі (додаток А).

Для зниження фактору суб'єктивності та невизначеності на ЛПР, було автоматизовано вибір параметрів прецедентів та параметрів поточної ситуації з бази знань відповідною функцією на мові С# (додаток А).

3. Здійснюється вилучення прецедентів з бази за допомогою алгоритмів пакетного та одиночного пошуку схожості для порівняння з поточною ситуацією.

Необхідно віднести поточну ситуацію до одного з відомих прецедентів: на вхід прототипу ЕС ІППР подається ситуація. CBR-цикл дає змогу обчислити міру схожості поточної ситуації та прецедентів із БП за метрикою індекс Жаккара для двох множин: множина А – база прецедентів, множина В – параметри поточної ситуації для всіх студентів, що навчаються (рис.3.9).

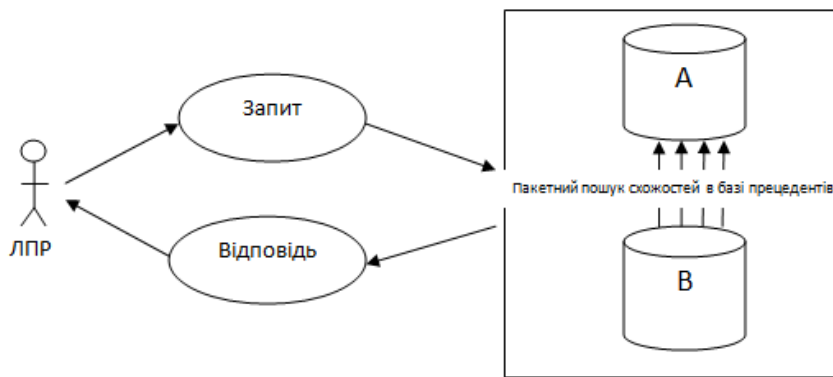


Рисунок 3.9 – Запит на агреговане порівняння кейсів з базою прецедентів

Студент	Case	%	Рекомендація системи
Ядренцев Іван Олегович	2	16,67%	Науковий керівник: Градинар Раїса Василівна. Тема ДП: Розробка інформаційної системи СТО
Шербань Володимир Вол...	3	22,22%	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка бази даних для сайту з наданн
Соломонюк Данило Воло...	3	10,53%	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка інтернет-магазину з продажу к
Рашупкін Павло Сергійович	5	16,67%	Науковий керівник: Полібіна Катерина Вячеславівна. Тема ДП: Чат-бот по доставці фастфуда
Нестеров Дмитро Олекса...	6	21,05%	Науковий керівник: Полібіна Катерина Вячеславівна. Тема ДП: Розробка 3D моделі "Кімната сту
Мельничук Олександр Ол...	1	19,05%	Науковий керівник: Градинар Раїса Василівна. Тема ДП: Розробка сайту з продажу меблів та тои
Ляхов Віталій Михайлович	3	13,64%	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка сайту-магазину товарів для дл
Кудас Ольга Сергіївна	10	15,38%	Науковий керівник: Шаповалов Володимир Миколайович. Тема ДП: Розробка комп'ютерної гри-т
Костюк Марко Євгенович	1	5,00%	Науковий керівник: Градинар Раїса Василівна. Тема ДП: Вплив нейромереж на графічний дизайн
Костерна Катерина Олек...	10	10,34%	Науковий керівник: Шаповалов Володимир Миколайович. Тема ДП: Розробка комп'ютерної гри
Коваль Олександр Віталій...	2	13,64%	Науковий керівник: Градинар Раїса Василівна. Тема ДП: Створення галереї "Cat notebook"
Книш Антон Володимиров...	3	13,64%	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка бази даних для телеграмм бот
Кім Дмитро Юрійович	5	13,64%	Науковий керівник: Полібіна Катерина Вячеславівна. Тема ДП: Розробка чат-боту у Телеграмі "Г

Рисунок 3.10 – Застосування метрики індекс Жаккара для пакетного пошуку

СВР-цикл дає змогу обчислити міру схожості поточної ситуації та прецедентів із БП за коефіцієнтом Танімото для двох множин: множина А – база прецедентів, множина В – параметри поточної ситуації для всіх студентів, що навчаються (рис.3.11).

Студент	Case	%	Рекомендація системи
Ядренцев Іван Олегович	2	16,67%	Науковий керівник: Градинар Раїса Василівна. Тема ДП: Розробка інформаційної системи СТО
Шербань Володимир Вол...	3	22,22%	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка бази даних для сайту з наданн
Соломонюк Данило Воло...	3	10,53%	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка інтернет-магазину з продажу к
Рашупкін Павло Сергійович	5	16,67%	Науковий керівник: Полібіна Катерина Вячеславівна. Тема ДП: Чат-бот по доставці фастфуда
Нестеров Дмитро Олекса...	6	21,05%	Науковий керівник: Полібіна Катерина Вячеславівна. Тема ДП: Розробка 3D моделі "Кімната сту
Мельничук Олександр Ол...	1	19,05%	Науковий керівник: Градинар Раїса Василівна. Тема ДП: Розробка сайту з продажу меблів та тои
Ляхов Віталій Михайлович	3	13,64%	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка сайту-магазину товарів для дл
Кудас Ольга Сергіївна	10	15,38%	Науковий керівник: Шаповалов Володимир Миколайович. Тема ДП: Розробка комп'ютерної гри-т
Костюк Марко Євгенович	1	5,00%	Науковий керівник: Градинар Раїса Василівна. Тема ДП: Вплив нейромереж на графічний дизайн
Костерна Катерина Олек...	10	10,34%	Науковий керівник: Шаповалов Володимир Миколайович. Тема ДП: Розробка комп'ютерної гри
Коваль Олександр Віталій...	2	13,64%	Науковий керівник: Градинар Раїса Василівна. Тема ДП: Створення галереї "Cat notebook"
Книш Антон Володимиров...	3	13,64%	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка бази даних для телеграмм бот
Кім Дмитро Юрійович	5	13,64%	Науковий керівник: Полібіна Катерина Вячеславівна. Тема ДП: Розробка чат-боту у Телеграмі "Г

Рисунок 3.11 – Застосування метрики коефіцієнт Танімото для пакетного пошуку

Як бачимо на рисунках 3.10 та 3.11, в обох випадках створюється ранжований список близькості прецедента та параметрів поточної ситуації. Індеси (коефіцієнти) виявляються однаковими, це може вказувати на консистентність або однорідність даних у рамках аналізованої вибірки.

Якщо людина ЛПР спостерігає, що коефіцієнти Жаккара та Танімото рівні, то може застосувати різні сценарії тлумачення отриманого рішення:

1. Це може вказувати на високу ступінь однорідності у інтересах студентів або у схожості дипломних робіт, які вони обирають.

2. Якщо алгоритм послідовно знаходить схожі або однакові результати для різних вибірок, це може свідчити про його надійність та валідність в контексті даної задачі.

3. ЛПР може зрозуміти, що опис поточної ситуації для може бути репрезентативним для всієї вибірки студентів, що вказує на загальні тенденції або уподобання серед студентської групи.

4. Якщо результати подібні, ЛПР може розглянути можливість стандартизації ресурсів або підходів, наприклад, при розподілі наукових керівників чи тем дипломних робіт, оскільки існує схожість у потребах студентів.

5. ЛПР може вирішити зосередитися на певних напрямках, які приваблюють студентів більше, та розробити стратегії для підтримки та розвитку цих сфер в освітньому закладі чи науковій сфері.

6. Рівність індексів може бути підставою для перегляду навчального плану або додаткових матеріалів для забезпечення більшої різноманітності чи спеціалізації, якщо вважається, що це необхідно для задоволення індивідуальних освітніх або професійних цілей студентів.

Узагальнюючи, ЛПР використовує ці дані для розуміння загальних тенденцій і індивідуальних інтересів у межах студентської спільноти, що дозволяє приймати обґрунтовані та ефективні рішення щодо курсу навчання, розподілу ресурсів та стратегічного розвитку освітніх програм.

Для того, щоб ЛПР мала можливість вибрати найбільш відповідний прецедент виходячи з множини оцінок подібності, то було додана можливість обробки прецедентів як множини прецедентів з бази прецедентів до окремого випадку поточної ситуації для окремо обраного студента (рис.3.12).

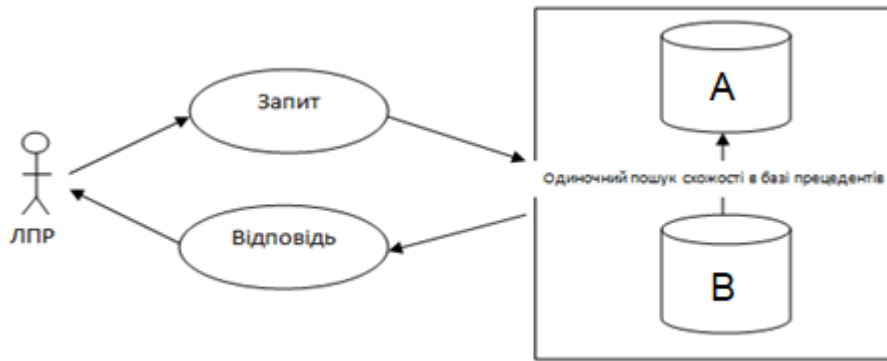


Рисунок 3.12 – Запит на визначення відповідності для конкретного кейсу

Для цього було використано метод триплетів. Цей підхід не є стандартною метрикою в методу найближчого сусіда, як індекс Жаккара або коефіцієнт Танімото, але може бути застосований у контексті обробки природної мови (NLP) для структурування та аналізу тексту. Тут триплети зазвичай представляють відносини між словами або концептами в межах речень або абзаців.

Для розуміння цього підходу, розглянемо його в контексті NLP. В NLP триплети часто використовують для представлення знань у форматі, який легко можна обробляти комп'ютерними системами. Триплет зазвичай складається з трьох частин: суб'єкт, предикат, об'єкт або в контексті нашої задачі: слово, контекст, значення.

Розглянемо метод оцінки схожості між двома текстами, розділивши кожен з них на слова та використавши ці слова для формування триплетів: спочатку поточна ситуація і база прецедентів підлягають попередній обробці - вони розбиваються на слова, видаляються стоп-слова, проводиться лематизація або стемінг [126].

Після попередньої обробки текстів і створення триплетів на основі слів, контексту та значення, можна застосувати алгоритми для оцінки схожості. Для кожного слова у поточній ситуації та у базі прецедентів можна визначити його контекст та значення, використовуючи різні методи, такі як векторизація тексту, TF-

IDF, word embeddings (наприклад, Word2Vec або GloVe), або навіть більш складні моделі, що використовують трансформери, такі як BERT [126].

Як було вказано вище, часто у NLP триплети представляють собою структури у форматі «суб'єкт-предикат-об'єкт». Але вище було виявлено однорідність текстових даних бази прецедентів, тому для даного завдання ми можемо використовувати простіший підхід, наприклад, генерування триграм із послідовностей слів.

Але ця реалізація не включає лематизацію або стемінг, які часто використовуються в NLP для зменшення кількості унікальних форм слів. Якщо це буде потрібно, то тоді треба інтегрувати бібліотеку NLP, наприклад, NLTK або spaCy. Інтеграція таких бібліотек у проєкт Visual Studio вимагає декількох кроків. Оскільки ці бібліотеки є бібліотеками Python, то потрібно буде використовувати Python у проєкті C# у Visual Studio. Один із способів зробити це — використання IronPython, який є реалізацією Python для .NET Framework.

Також, для обробки тексту у проєктах C# Windows Forms, можна скористатися бібліотеками .NET, які доступні через NuGet. Однією з таких бібліотек є System.Text.RegularExpressions для роботи з регулярними виразами, яка може допомогти у фільтрації та аналізі тексту. Для більш складного NLP можна використовувати бібліотеки, такі як Stanford.NLP для .NET.

На рисунку 3.13 представлені результати, які були отримані за допомогою методу триплетів (триграм). Формується ранжований список всіх рекомендацій для окремого студента (для прикладу визначмо студента «Соломонюк»). За попереднім методом, який використовував індекс Жаккара та коефіцієнт Танімото, відсотки схожості, які показані в колонці «%» = 10,53% та номер кейсу (пропоноване рішення) в колонці «Case»=3. Отриманий індивідуальний ранжований список методом триграм має інший кейс «Case»=4 та відсоток схожості «%»=27.01%. Це може свідчити про те, що метод триплетів (триграм) більш ефективно враховує контекст і структуру тексту, а також може краще вловлювати відтінки значень, що в

результаті призводить до більш високого рівня визначення схожості між даними студентів і прецедентами.

Студент	Case	%	Рекомендація системи	Рекомендовані дисципліни для вивчення	Повідомлення системи
Соломонюк Данило Воло...	4	27,01%	Маюк Руслан Андрійович	Web-технології	Немає додаткових рекомендацій
Соломонюк Данило Воло...	3	13,85%	Маюк Руслан Андрійович	Web-технології, Базы даних	Немає додаткових рекомендацій
Соломонюк Данило Воло...	10	10,87%	Шаповалов Володимир Миколайович	Базы даних, Людино-машинний інтерфейс	Студенту потрібно підвищити загал...
Соломонюк Данило Воло...	8	10,05%	Калугін Дмитро Валерійович	Операційні системи	Немає додаткових рекомендацій
Соломонюк Данило Воло...	2	9,41%	Градинар Раїса Василівна	Алгоритми та структури даних	Немає додаткових рекомендацій

Рисунок 3.13 – Розрахунок схожості методом триплетів

Таким чином зменшується фактор невизначеності при прийнятті рішення ЛПР. Також, це може бути особливо корисним у задачі вибору наукового керівника та теми диплома, де важливо не просто виявити ключові слова, але й зрозуміти більш глибокі семантичні взаємозв'язки між темами інтересів студента та спеціалізацією потенційних наукових керівників.

Зазвичай CBR-цикл виконується за безпосередньої взаємодії з ЛПР. Багато прецедентних систем тільки витягують зі сховища найдоречніші прецеденти і залишають процес адаптації на розсуд ЛПР, оскільки автоматизація процесу адаптації є складним завданням і практично не піддається узагальненню, а найчастіше в адаптації немає необхідності, оскільки вибрані прецеденти містять достатньо інформації для прийняття рішення [101].

4. Остаточні результати аналізу, включно з визначенням найбільш відповідних прецедентів та рекомендаціями, зберігаються для доступу та використання ЛПР.

При прийнятті рішення по вибору наукового керівника та теми дипломного проекту ЛПР з отриманого ранжованого списку рекомендацій може обрати будь-який варіант, ґрунтуючись на своїх уподобаннях. Але задля зменшення впливу факторів суб'єктивності та невизначеності на ЛПР, в системі накладені обмеження. Навіть, якщо відсоток схожості по кейсу є найвищим, але студент має незадовільний рівень знань, а викладач немає досвіду роботи з невстигаючими студентами, то рекомендаційна система не надасть можливості обрання цього кейсу для студента (рис.3.14). Ці обмеження реалізовані відповідною функцією на мові C# (додаток А).

Студент	Case	%	Рекомендація системи	Рекомендовані дисципліни для вивчення	Повідомлення системи
Барабаш Даніл Сергійович	9	31,21%	Шаповалов Володимир Миколайович	Інженерна та комп'ютерна графіка, Кон	Рекомендація не може бути створена
Барабаш Даніл Сергійович	10	27,04%	Шаповалов Володимир Миколайович	Бази даних, Людино-машинний інтерфей	
Барабаш Даніл Сергійович	6	22,64%	Полібіна Катерина Вячеславівна	Інженерна та комп'ютерна графіка, Об	
Барабаш Даніл Сергійович	5	18,18%	Полібіна Катерина Вячеславівна	Web-технології, Програмування для моб	
Барабаш Даніл Сергійович	3	17,74%	Маюк Рвслан Андрійович	Web-технології. Бази даних	

Рисунок 3.14 – Приклад процесу зниження факторів невизначеності на ЛПР

Після прийняття обраного рішення щодо вибору наукового керівника та теми дипломного проєкту, прийняте рішення зберігається в системі.

Далі, процес автоматизованої обробки прийнятого рішення для формування індивідуального плану студента по виконанню завдань дипломного проєктування, виконується без використання циклу CBR. Замість цього, для отримання результативного документа у формі індивідуального навчального плану студента використовуються прості емпіричні асоціації, що формуються на основі умовних правил типу «Якщо X, то Y» та їхніх модифікацій. Це означає, що система використовує певні заздалегідь встановлені правила або алгоритми, які автоматично застосовуються до даних про студента, обрану тему та керівника. На основі цих правил система визначає оптимальний навчальний план для студента, враховуючи його поточний стан, цілі та вимоги навчального закладу.

Такий підхід дозволяє швидко та ефективно створювати індивідуальні навчальні плани без необхідності використання складних аналітичних методів або великої кількості експертного втручання. Крім того, це дозволяє автоматизувати процес прийняття рішень у слабоформалізованих навчальних середовищах, де інформація може бути неповною або неоднозначною.

3.6 Висновки до третього розділу

Результати апробації створеної бази прецедентів свідчать про те, що запропонована структура має властивості консистентності та відсутності

надмірності, що є ключовими аспектами для ефективного функціонування системи підтримки прийняття рішень у невизначених та слабоформалізованих умовах.

Запропонований метод пошуку рішення за допомогою методу триплетів та обрахунком коефіцієнтів схожості, таких як Жаккар і Танімото, дозволяє створити ефективну систему для вирішення задач прийняття рішення, яка може виявити не лише явну, але й приховану схожість між різними випадками. Такий підхід підвищує якість рекомендаційної системи, забезпечуючи більш відповідні та точні рекомендації, при виборі наукових керівників та тем дипломних робіт для студентів, враховуючи їх індивідуальні інтереси та вже наявний досвід наукового керівника.

РОЗДІЛ 4

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПІДТРИМКИ ВИБОРУ ІНДИВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ

Тестування експертної інтелектуальної системи підтримки вибору індивідуальної освітньої траєкторії «Choice» здійснювалося в Новокаховському приладобудівному фаховому коледжі (далі НКПФК). Практична апробація створеного прототипу дозволила підвищити якість прийняття рішення індивідуалізованого вибору наукового керівника та теми дипломного проєкта студентів з урахуванням більш точного зіставлення їх академічних та професійних переваг з компетенціями та експертизою потенційних наукових керівників; запропоновано адаптивний вибір місця практики, який дозволяє студентам знаходити оптимальні варіанти розвитку своїх професійних навичок; здійснено інтеграцію даних та зворотного зв'язку, який забезпечує студентів актуальною інформацією, дозволяючи їм ухвалювати рекомендації експертної системи; проаналізовано вимоги ринку праці в галузі ІТ, який стрімко змінюється, що забезпечує студентам можливість вибору таких керівників та місць практики, які максимально відповідають сучасним вимогам ІТ індустрії. Акт впровадження наведено в додатку А.

4.1 Аналіз процесу обрання теми дипломного проєкта на відділенні інженерії програмного забезпечення в НКПФК

Згідно з Положенням «Про порядок створення та організацію роботи Державної екзаменаційної комісії НКПФК» атестацію проходить кожен студент після повного виконання ним навчального плану за відповідною спеціальністю. Атестація фахових молодших бакалаврів здійснюється згідно з вимогами освітньо-професійної програми та навчального плану у формі публічного захисту

кваліфікаційної роботи (дипломного проєкту). Випускні кваліфікаційні дипломні проєкти подаються студентами на випускове відділення у визначений випусковою цикловою комісією (далі ЦК) термін, але не пізніше ніж за тиждень до дня захисту в ДЕК [127].

Наведемо детальний аналіз аспектів діяльності, що цікавить: вибір теми випускної кваліфікаційної роботи (далі Дипломний проєкт (ДП)) на відділенні ІПЗ в НКПФК.

Основна взаємодія відбувається між викладачами відділення ІПЗ (наукові керівники, зав. відділенням) та студентами (рис. 4.1). У процесі складання списку тем на початковому етапі активно взаємодіють співробітники всередині відділення: завідувач відділення та викладачі відділення, які готові стати науковими керівниками.



Рисунок 4.1 – Організаційна модель

Завідувач відділенням ІПЗ відповідає за організацію всього процесу: надсилає запит про пропозицію нової теми викладачам, формує списки тем (спочатку без вказівки наукового керівника, а потім з вказівкою), затверджує остаточний список тем на засіданні циклової комісії програмної інженерії протоколом [127].

Наукові керівники після отримання запиту від завідувача відділенням ІПЗ, направляють свої пропозиції щодо можливих тем.

Студенти аналізують складений список тем, вибирають відповідну та оформляють заяву за встановленою формою у наукового керівника для затвердження його обраної теми. Студент також може запропонувати свою тему.

Функціонально-технологічна модель:

1. Запит з відділення ІПЗ викладачам про пропозицію теми для ДП (e-mail).
2. Направлення теми ДП з боку викладача на відділення ІПЗ (e-mail).
3. Формування на відділенні ІПЗ списку тем без зазначення наукових керівників.
4. Вибір студентом теми ДП зі списку або пропозиція своєї теми.
5. Формування на відділенні ІПЗ списку тем із зазначенням для кожної теми наукового керівника і студента.
6. Подання студентом заяви науковому керівникові про затвердження обраної теми.
7. Затвердження остаточного списку тем і формування службової записки на засіданні ЦК інженерії програмного забезпечення та наказом по НКПФК.

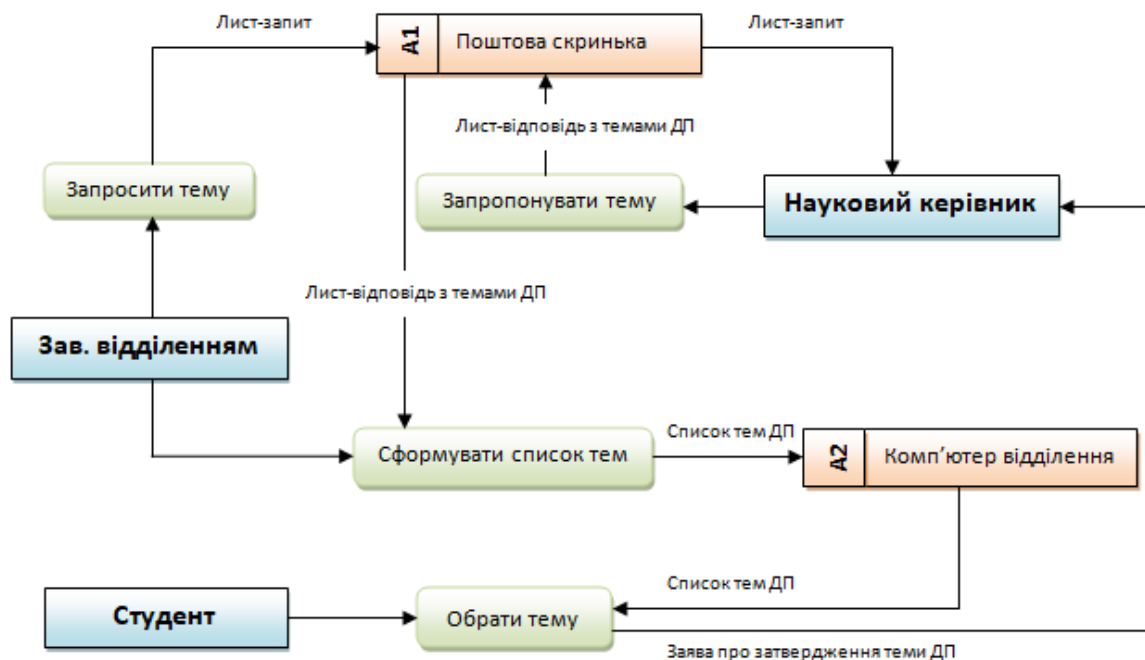


Рисунок 4.2 – Діаграма процесів вибору теми ДП

Запит про пропозицію тем ініціюється завідувачем відділенням ІПЗ (рис. 4.2). Наукові керівники пропонують варіанти тем ДП і на їх основі формується попередній список тем. Студент повинен вибрати тему, що його зацікавила, із запропонованого списку, або свій варіант, після чого він пише заяву на ім'я керівника на затвердження теми. Остаточне затвердження тем відбувається наказом по НКПФК.

У процесі вибору теми ДП студентом задіяні не так багато людей і документів, однак у поточній моделі процес має слабку інформаційну підтримку, не автоматизований та не дає можливості прийняття релевантних рішень. У поточній моделі занадто багато дій потрібно робити вручну (обмін інформацією за темами по e-mail, формування списку тем, опитування студентів для вибору тем, складання заяви тощо).

Застосування експертної інтелектуальної системи підтримки вибору наукового керівника та теми ДП може значно полегшити цей процес. Як було попередньо доведено, ЕС ІППР здатна за допомогою інтелектуального аналізу освітніх даних аналізувати різноманітні фактори, такі як інтереси студента, його академічні досягнення та попередні наукові дослідження, а також наявність наукових керівників, їх експертизу та досвід. На основі цих даних система може рекомендувати студентам наукових керівників та відповідні теми ДП, що відповідають їхнім потребам і можливостям та в подальшому коректувати їх індивідуальну освітню траєкторію.

У зв'язку з тим, що метою застосування ЕС ІППР є підвищення ефективності та оптимізація процедури прийняття рішень ЛПР, створення бази прецедентів вдалих випадків наукового керівництва з минулих років значно покращуватиме процес вибору успішного тандему наукового керівника та студента-випускника для подальшої роботи над ДП.

Використання запропонованого CBR циклу зіставлення поточної ситуації студента з базою прецедентів вдалих випадків наукового керівництва з минулих

років дозволить ЛПР за допомогою експертної системи аналізувати та порівнювати подібні сценарії та визначати оптимальні рішення для конкретного студента. Цей процес допоможе індивідуалізувати рекомендації щодо вибору наукового керівника та теми ДП та запропонувати подальший індивідуальний план навчання.

Виходячи з вищевказаного було модифіковано модель процесу підтримки ЛПР при виборі наукового керівника та теми ДП для студента (рис.4.3).

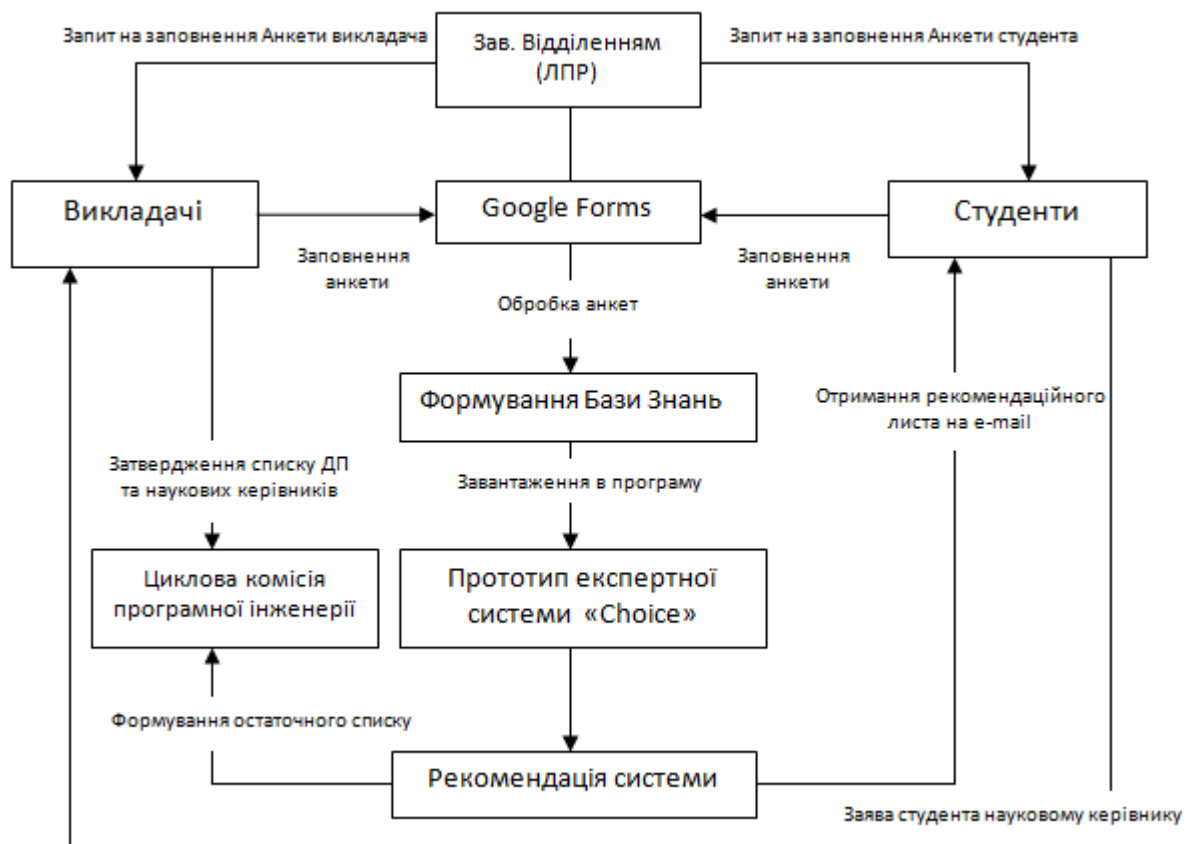


Рисунок 4.3 – Модифікована модель процесу підтримки ЛПР

1. Запит зав. відділення ІІЗ викладачам про заповнення особистої анкети (google forms).
2. Запит зав. відділення ІІЗ студентам про заповнення особистої анкети (google forms).

3. Аналіз анкет та заповнення спеціального шаблону в форматі файлу Excel «Анкета викладача» та «Анкета студента».
4. Завантаження підготовлених освітніх даних в ЕС ІППР «Choice».
5. Отримання рекомендацій ЕС ІППР «Choice».
6. Затвердження остаточного списку тем і формування службової записки на засіданні ЦК відділення програмної інженерії.
7. Остаточне затвердження тем ДП наказом по НКПФК.

В зв'язку з вимушеним переходом на дистанційне навчання в умовах воєнного стану в Україні, коли спостерігається утруднення комунікації між студентом та викладачем, зниження відповідальності студентів та невідання організувати самонавчання, доцільно передати процес вибору теми дипломного проєкту та наукового керівника до рук однієї особи, людини, яка приймає рішення (ЛПР).

Поділ відповідальності та передача процесу вибору теми дипломного проєкту та наукового керівника під контроль однієї особи може бути обґрунтованим з кількох причин [128]:

1. В умовах обмеженої комунікації та можливих складнощів з координацією між різними структурними одиницями коледжу, централізована особа, яка ухвалює рішення, може спростити та прискорити процес вибору теми та керівника для студентів.
2. Централізований підхід дає змогу забезпечити дотримання стандартів якості у виборі теми та наукового керівника, тому що одна особа може слідкувати за дотриманням критеріїв і вимог, встановлених коледжем.
3. Одна людина, яка ухвалює рішення, може запобігти можливим конфліктам між студентами, а також звести до мінімуму ризик непорозуміння або неправильної інтерпретації вимог.
4. В умовах ускладненої комунікації та невизначеності, студенти можуть відчувати додаткову напругу під час вибору теми та керівника. Централізований підхід може зняти з них цю відповідальність і спростити процес вибору.

У контексті вибору наукового керівника та теми дипломного проєкту ЛПР в особі завідуючого відділенням може відігравати роль ключового фасилітатора і координатора цього процесу. ЛПР організовує і керує процесом вибору, забезпечуючи ефективну реалізацію завдання вибору наукового керівника для студентів та теми ДП.

Для того щоб належним чином розуміти та керувати цим процесом, необхідно створити діаграму прецедентів, яка детально відобразить всі можливі взаємодії між ЛПР та ЕС ІППР. Ця діаграма буде відображати різноманітні сценарії та можливі варіанти дій, які може виконати ЛПР у процесі вибору (рис.4.4).

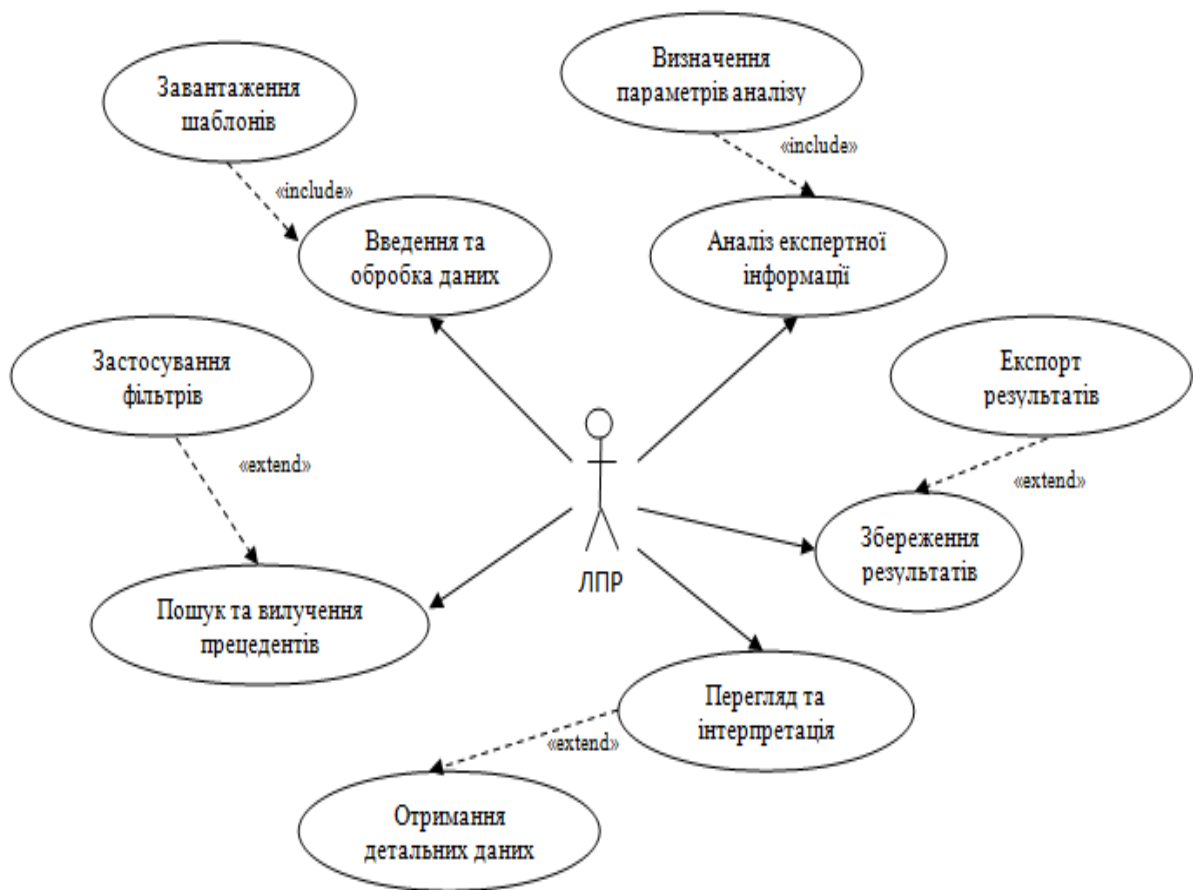


Рисунок 4.4 – Діаграма прецедентів взаємодії ЛПР з системою ЕС «Choice»

Актор - завідуючий відділенням (ЛПР), користувач системи, який відповідає за вибір наукового керівника та тем дипломних проєктів для студентів.

Основні випадки використання:

- введення та обробка даних - ЛПР завантажує дані анкетування, навантаження викладачів та успішність студентів, тип зв'язку – include. «Завантаження шаблонів» - перед введенням даних система надає шаблони для організації даних;
- аналіз експертної інформації - система обробляє введені дані, використовуючи алгоритми класифікації, тип зв'язку – include. «Визначення параметрів аналізу» - ЛПР вказує критерії та параметри для аналізу даних;
- пошук та вилучення прецедентів - ЛПР ініціює пошук прецедентів з бази прецедентів на основі поточної ситуації, тип зв'язку – extend. «Застосування фільтрів» - ЛПР може застосовувати фільтри для уточнення пошуку;
- збереження результатів - система зберігає рекомендації та аналітичні звіти, тип зв'язку – extend. «Експорт результатів» - результати можуть бути експортовані в різні формати;
- перегляд та інтерпретація результатів - ЛПР переглядає результати аналізу та виробляє рішення, тип зв'язку – extend. «Отримання детальних даних» - ЛПР може запитувати детальну інформацію за конкретними пунктами аналізу.

Отже, на основі розроблених вище моделей, методів та алгоритмів створено прототип ЕС ІППР «Choice», який дозволяє вирішити низку важливих прикладних завдань в освітньому процесі в НКПФК – підвищення ефективності та оптимізація процедури прийняття рішень взагалі та інформаційної підтримки ЛПР, зниження впливу людського фактору на процес прийняття рішення у невизначених ситуаціях .

4.2 Вимоги до прототипу ЕС ІППР «Choice»

Розроблений прототип ЕС ІППР «Choice» складається з чотирьох підсистем:

- підсистема автоматизації завантаження освітніх даних;
- підсистема аналітичного аналізу освітніх даних;

- підсистема підтримки рішення ЛПР;
- підсистему візуалізації прийнятого рішення ЛПР.

Функціональні вимоги:

- інтуїтивно зрозумілий та зручний інтерфейс для взаємодії з ЛПР, що включає функціонал введення даних, пошуку та відображення результатів;
- можливість ефективного зчитування, зберігання та обробки даних з файлів формату MS Excel;
- автоматизована обробка та аналіз експертної інформації для створення бази прецедентів та параметрів поточної ситуації;
- реалізацію та налаштування методів для точного вилучення прецедентів;
- функціонал для зберігання результатів, рекомендацій тощо.

Система повинна мати адекватну обробку помилок і не допускати пошкодження даних у їх результаті.

Вимоги до продуктивності:

- забезпечення швидкої обробки великих обсягів даних без значних затримок або збоїв у роботі;
- оптимізація алгоритмів для забезпечення точних і швидких розрахунків.

Вимоги до підтримки:

- можливість легкого оновлення бази даних і алгоритмів обробки без необхідності значних змін у програмі;
- наявність технічної підтримки для користувача у разі виникнення проблем або питань.

Вимоги до зручності користування:

- забезпечення зрозумілості та легкості навігації у інтерфейсі для користувачів без технічного фону;
- надання чітких інструкцій або керівництв для користувачів щодо використання системи, довідники;

– можливість налаштування деяких аспектів інтерфейсу або функціоналу згідно з потребами користувачів.

Виконання цих вимог забезпечить створення ефективної, зручної у використанні та надійної експертної системи, яка зможе задовольнити потреби ЛПР і сприяти підвищенню якості прийнятого їм рішення.

В розробці та програмній реалізації прототипу ЕС ІППР «Choice» були використані мова програмування C#, середовище програмування MS Visual Studio Enterprise 2017, бібліотеки пакету EPPlus – OfficeOpenXml, бібліотеки пакету Microsoft.Office.Interop.Word, який входить до Microsoft Office Primary Interop Assemblies (PIA) і забезпечує інтерфейс для взаємодії з Microsoft Word за допомогою COM-взаємодії, середовище MS Microsoft Excel.

Прототип ЕС ІППР «Choice» створена для навчального закладу НКПФК та розрахована на використання трьома користувачами — завідувачами відділеннями. Ця обмеженість кількості користувачів відображена у функціональності програми: передбачена можливість реєстрації лише для трьох осіб. Після реєстрації третього користувача кнопка реєстрації стає недоступною, тим самим запобігаючи подальшому додаванню нових користувачів (рис. 4.5).

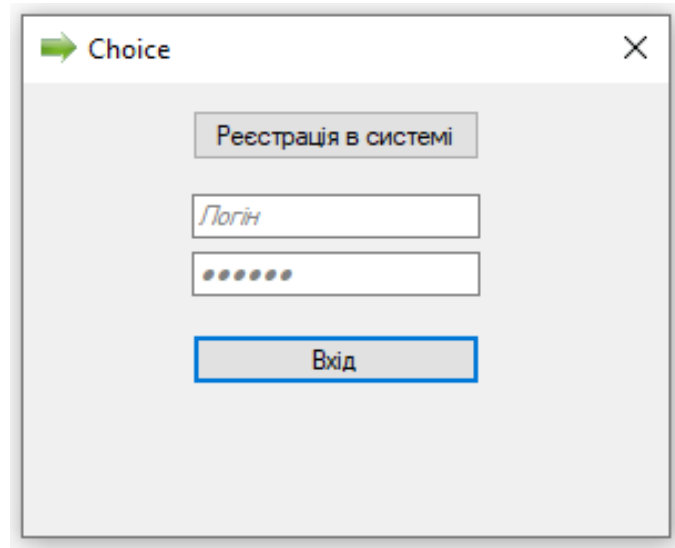


Рисунок 4.5 – Вікно реєстрації в ЕС ІППР «Choice»

Логіни та паролі користувачів зберігаються в ресурсному файлі програми, що забезпечує надійне та безпечне зберігання цих важливих даних. Авторизація в системі виконується на основі цих логінів і паролів, що гарантує, що доступ до програми мають лише уповноважені особи.

Після успішної авторизації користувач отримує доступ до спеціалізованого інтерфейсу (рис.4.6), розробленого з урахуванням його потреб та завдань:

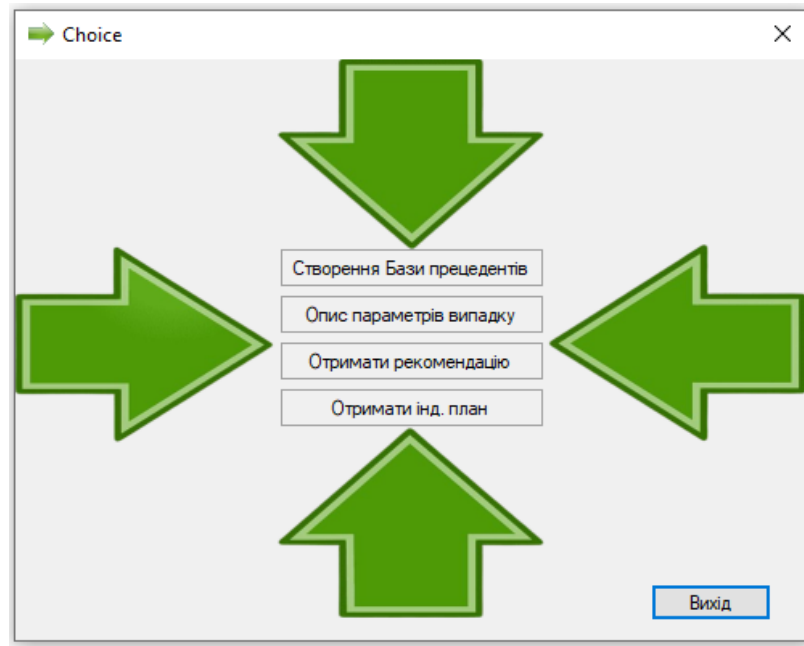


Рисунок 4.6 – Спеціалізований інтерфейс користувача програми

- створення бази прецедентів. Користувач може додавати до бази прецедентів інформацію про теми попередніх дипломних проектів, наукових керівників, їх особисті характеристики, експертизи, якими володіють, галузь наукових інтересів;
- опис параметрів поточної ситуації. Користувач може вводити дані про поточну ситуацію, що описує досягнення студента, рівень знань, досвід попередніх досліджень, бажану тему ДП, рівень знань тощо;
- отримання рішення. Після введення поточної ситуації користувач запрошує програму зробити пропозиції щодо створення тандему наукового керівника та

студента для подальшої роботи над бажаною темою ДП. Також користувач має можливість перегляду списку запропонованих рішень та обрати остаточне;

– отримання індивідуального навчального плану. Після вибору релевантного рішення, користувач запрошує програму підготувати індивідуальний навчальний план для студента, який включатиме індивідуальний графік навчального процесу та запропоноване місце практики тощо.

Кожна з цих функцій спрямована на полегшення процесу вибору наукового керівника та теми дипломного проекту та надання студентам належної підтримки у виконанні індивідуального навчального плану.

4.2.1 Підсистема автоматизації завантаження освітніх даних

У НКПФК основні процеси, такі як: ведення бази даних студентів з інформацією про персональні дані, успішність, відвідуваність та інші параметри, створення розкладу занять для студентів і викладачів, організація роботи над дипломними проектами, курсовими роботами та іншими навчальними проектами, створення звітів, аналіз даних успішності тощо, здійснюються за допомогою програми Microsoft Office Excel. Нижче представлено модель документообігу (таблиця 4.1) процесу обрання наукового керівника та теми дипломного проекту на відділенні інженерії програмного забезпечення в НКПФК з використанням електронних таблиць Microsoft Office Excel.

Таблиця 4.1

Модель документообігу

	Вихідний документ	Операція	Виконавець	Вхідні документи	Накопичувач
1	Анкета викладача	Створення Анкети google forms	Завідувач відділенням		Google Drive (D)
2		Заповнення Анкети викладача	Викладачі	Анкета викладача	Поштова скринька (A1)

Продовження Таблиці 4.1

3	Анкета Студента	Створення Анкети google forms	Завідувач відділенням		Google Drive (D)
4		Заповнення Анкети Студента	Студенти	Анкета студента	Поштова скринька (A1)
5	Анкети викладачів та студентів	Аналіз та обробка даних анкет	Завідувач відділенням	Файли формату .xlsx «TeacherProfile» «StudentProfile»	Комп'ютер відділення (A2)
6	Анкети викладачів та студентів Навантаження викладачів Відомість успішності	Завантаження даних в прототип ЕС «Choice»	Завідувач відділенням	Файли формату .xlsx «TeacherProfile» «StudentProfile» «Навантаження викладачів» «Відомість успішності»	Комп'ютер відділення (A2)
	Наказ про призначення наукових керівників та тем ДП Лист-рекомендація Індивідуальний план студента		Завідувач відділенням		Комп'ютер відділення (A2)

Аналіз освітньої практики показує, що найчастіше онлайн-анкети будуються на основі загальнодоступних сервісів. Найбільш поширеним є сервіс Google Forms. Цей сервіс входить до складу мережевого офісного пакета, дає змогу створювати онлайн-форми та опитування. Враховуючи те, що анкети, створені в Google Forms, можуть бути легко експортовані у форматі .xlsx, вони практично не вимагають ручної обробки чи додаткового аналізу. Для забезпечення ефективної інтеграції з рекомендаційною системою достатньо просто зберегти ці анкети на комп'ютері відділення, де встановлено додаток (прототип експертної системи), з логічним та зрозумілим іменем файлу, наприклад, «TeacherProfile.xlsx». Таким чином, файл анкети готовий до безпосереднього завантаження у систему, що забезпечує швидке, зручне та автоматизоване використання зібраних даних.

Отже, анкетування студентів та викладачів дозволила ефективно виявити їх поточні інтереси, створити та завантажити базу знань в прототип ЕС ІППР по обранню наукового керівника та теми ДП.

Для ефективної роботи з прототипом ЕС ІППР необхідно підготувати низку файлів, які будуть включати дані анкетування студентів та викладачів, навантаження викладачів, а також відомості про успішність студентів. У програмі реалізована можливість завантаження спеціально підготовлених шаблонів (рис.4.6), які призначені для заповнення даних у Excel-файлах за певною стандартизованою структурою.

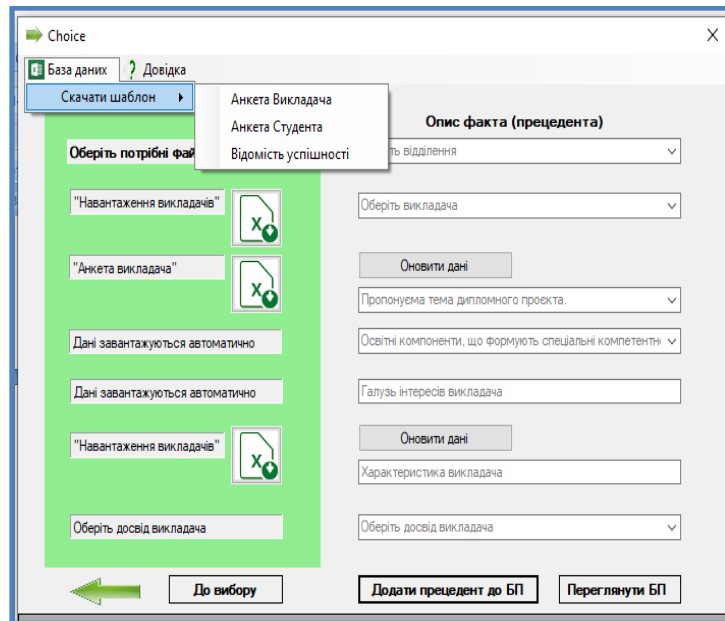


Рисунок 4.6 – Завантаження шаблонів Excel-файлів бази даних

Ця функція забезпечує, що всі дані будуть організовані у єдиному та зрозумілому форматі, що спрощує їх подальше використання у системі. Застосування шаблонів дозволяє ЛПП легко і швидко обирати необхідні дані без помилок, що можуть виникати при ручному введенні. Але і ручне введення даних залишається доступним.

Крім того, ця функція надає можливість автоматизувати створення тестових вибірок для формування бази прецедентів та параметрів поточної ситуації. Такий

підхід значно полегшує процес аналізу даних і дозволяє ЛПР більш ефективно використовувати систему для прийняття обґрунтованих рішень.

4.2.2 Підсистема аналітичного аналізу освітніх даних

За час збору, обробки та обліку анкетних даних викладачів і студентів зібралася невелика інформаційна база анкетних даних, для збереження та обробки яких було застосовано електронні таблиці Excel.

Щоб мінімізувати можливість помилок та неправильного введення даних з боку ЛПР, була впроваджена система (рис.4.7), що обмежує їхню участь у завантаженні даних. Автоматизація процесу та встановлені обмежень були реалізовані за допомогою спеціально розроблених функцій на мові програмування C# (додаток А).

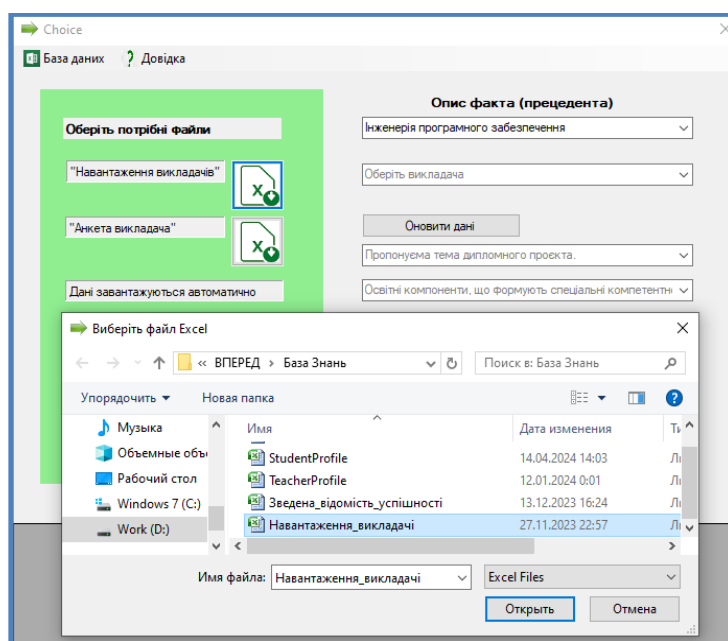


Рисунок 4.7 – Автоматизоване завантаження даних в ЕС ІППР

Автоматизація створення тестової вибірки є ключовим аспектом ефективної роботи прототипу ЕС ІППР. У цьому процесі важливу роль відіграють як експерт предметної області, так і розробник системи. Вони працюють разом для

встановлення відповідних обмежень та параметрів, що дозволяють автоматизувати пошук та реалізувати автоматичні підстановки з бази даних (рис 4.8). Це сприяє створенню точних та відповідних вибірок даних, необхідних для формування бази прецедентів.

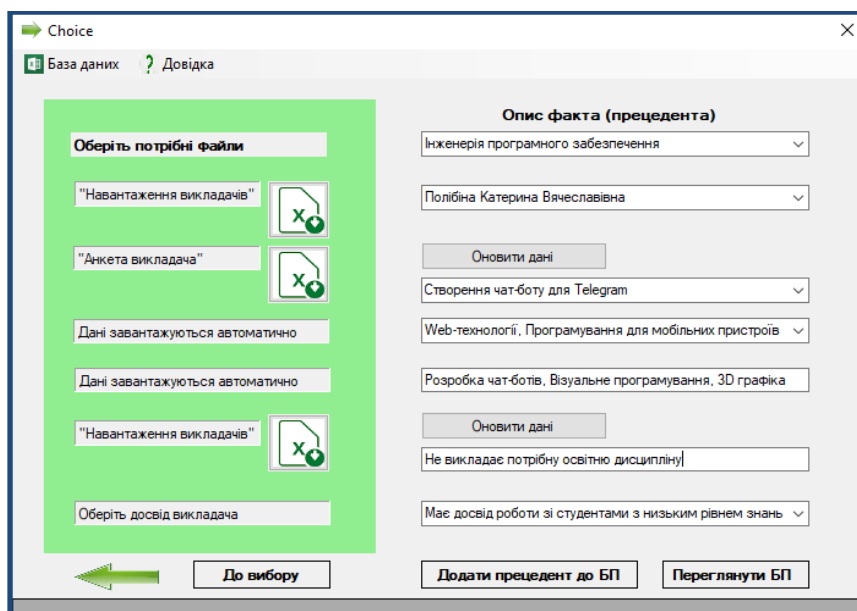


Рисунок 4.8 – Автоматичне створення вибірки за обраним викладачем

Ці функції виконують такі завдання, як перевірка введених даних на предмет помилок, повторювання чи невідповідностей.

Автоматизація ключових аспектів збору даних не тільки спрощує роботу ЛПР, але й підвищує якість та надійність інформації, що використовується системою для прийняття обґрунтованих рішень. Завдяки цим методам, процес створення бази прецедентів та бази параметрів для кожної поточної ситуації стає значно більш точним, швидким та ефективним.

Створення БП поділяється на такі етапи (рис.4.9):

а) створення прецедента за допомогою тестової вибірки. Як було зазначено вище, створення тестової вибірки було автоматизовано. З використанням обмежень, встановлених експертом предметної області та розробником системи, формується вибірка даних, яка використовується для створення прецедентів (рис.4.8). Ці

прецеденти слугують як вхідними даними для системи, так і використовуються для аналізу, прийняття рішень та вивчення можливих сценаріїв у контексті вибору наукових керівників та тем дипломних проектів;

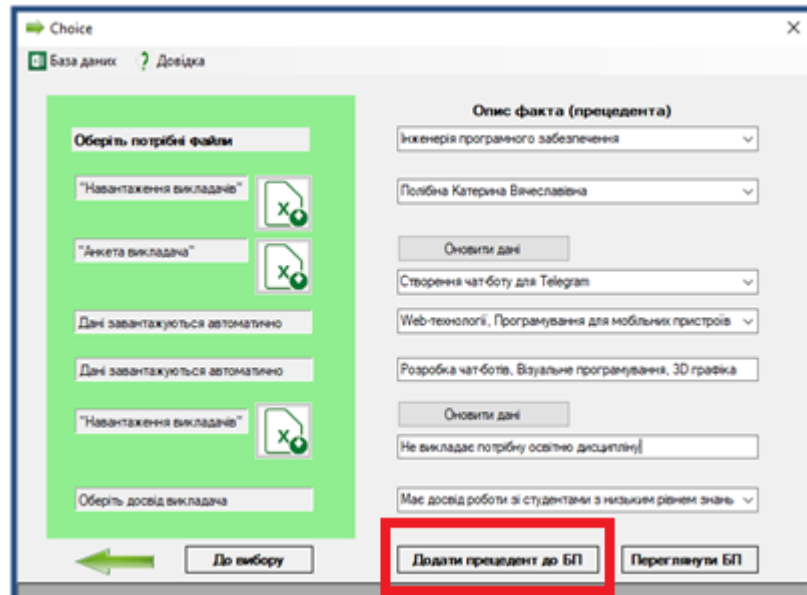


Рисунок 4.9 – Створення бази прецедентів

б) перевірка доданих прецедентів до бази прецедентів на дублікат. Кожен прецедент, який додається до бази прецедентів (ресурсного файлу системи), перевіряється на наявність дублікатів. Цей процес забезпечує, що кожен новий прецедент є унікальним і не повторює вже існуючі випадки у базі даних. Перевірка на дублікати виконується для кожного прецедента окремо, що дозволяє підтримувати високу якість та достовірність даних у системі. Ці функції реалізовано на мові програмування C# (Додаток А). Завдяки цим функціям, процес роботи з базою прецедентів стає більш автоматизованим та ефективним. В результаті, ЛПР отримують точний та корисний інструмент для аналізу поточних ситуацій та прийняття обґрунтованих рішень.

Створення БППС поділяється на такі етапи:

а) створення опису параметрів поточної ситуації за допомогою тестової вибірки (рис.4.10);

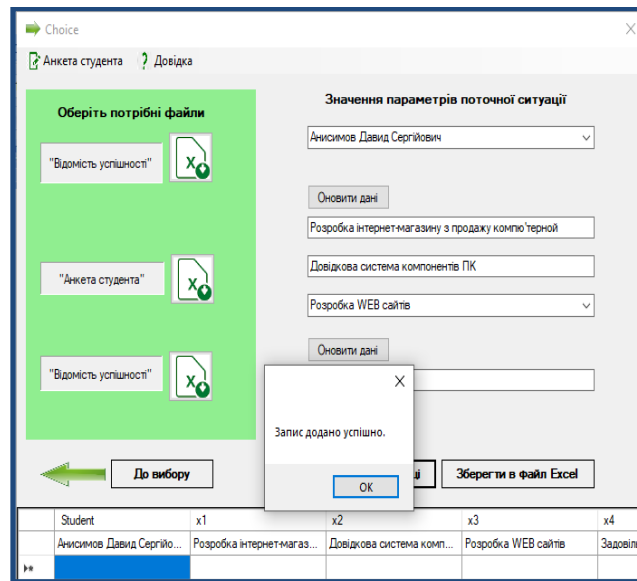


Рисунок 4.10 – Створення тестової вибірки параметрів поточної ситуації

б) перевірка доданих параметрів до БППС (перевірка здійснюється для кожного доданого параметра окремо) при збереженні опису параметрів поточної ситуації у базу параметрів поточної ситуації (ресурсний файл системи). Ці функції також реалізовано на мові програмування C# (Додаток А). Завдяки цим функціям, процес роботи з базою параметрів поточної ситуації стає більш автоматизованим та ефективним. В результаті, ЛПР отримують точний та корисний інструмент для аналізу поточних ситуацій та прийняття обґрунтованих рішень (рис.4.11).

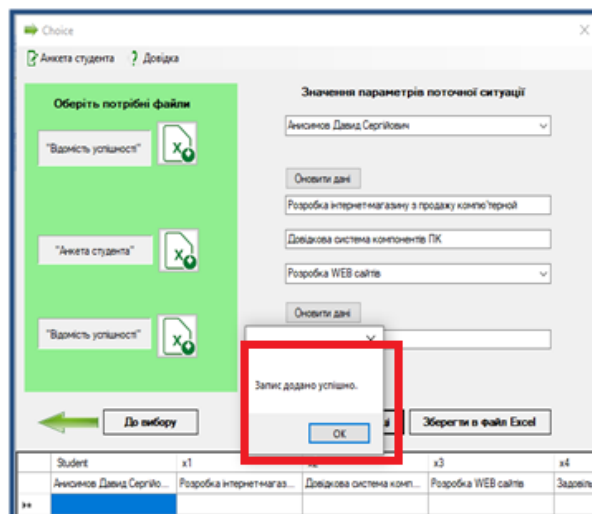


Рисунок 4.11 – Створення бази параметрів поточної ситуації

В контексті розробленої експертної системи, ключовим елементом є наявність бази прецедентів та бази параметрів поточної ситуації, які зберігаються в ресурсних файлах програми. Ці бази знань є обов'язковими для функціонування системи, оскільки вони містять необхідну інформацію для аналізу, виведення, прийняття та (якщо треба) перегляду рішень ЛПР.

4.2.3 Підсистема підтримки рішення ЛПР

Блок отримання рішення (рис.4.12), який є центральним компонентом системи, складається з двох таблиць у графічному інтерфейсі програми. Ці таблиці призначені для відображення та обробки даних, завантажених із зазначених баз знань, які зберігаються в програмі.

Перша таблиця використовується для представлення даних з бази прецедентів, тоді як друга таблиця містить дані з бази параметрів поточної ситуації. Це дозволяє користувачам системи ефективно візуалізувати та аналізувати інформацію, необхідну для прийняття обґрунтованих рішень.

Choice

Завантажити Базу Прецедентів

Case	x1	x2	x3	x4	x5	x6
1	Градинар Раїса Василівна	Розробка бази даних (обраної пред...	Бази даних, Основи прогр...	Бази даних, Про...	Викладає потрібну освітн...	Має досвід роботи зі ст...
2	Градинар Раїса Василівна	Розробка інформаційної системи (о...	Алгоритми та структури д...	Бази даних, Про...	Викладає потрібну освітн...	Має досвід роботи зі ст...
3	Маюк Руслан Андрійович	Розробка web-сайту	Web-технології, Бази даних	Розробка web-с...	Викладає потрібну освітн...	Має досвід роботи зі ст...
4	Маюк Руслан Андрійович	Розробка Інтернет магазину (прода...	Web-технології	Розробка web-с...	Викладає потрібну освітн...	Має досвід роботи зі ст...

Завантажити параметри поточної ситуації

Оберіть студента: Оберіть вибірову дисципліну: Отриманий рівень знань:

Student	x1	x2	x3	x4
Анисимов Давид Сергійо...	Розробка інтернет-магазину з прод...	Довідкова система компонентів ПК	Розробка WEB сайтів	Задовільний
Барабаш Даніл Сергійович	Розробка комп'ютерної гри на двиг...	"Двигуни для розробки комп'ютерн...	Розробка комп'ютерної гри	Незадовільний
Гаврашенко Богдан Сергі...	Розробка сайту інтернет магазин п...	Система для торгівельного предста...	Розробка WEB сайтів	Задовільний
Кальян Ігор Олександрович	Розробка багатофункціонального д...	Управління студією звукозапису	Програмування для мобільних прис...	Задовільний

Отримати сценарій рішення | Прийняти рішення | Очистити | Отримати рекомендацію | Отримати список | Створити наказ | До вибору

№ варіанту | Студент | Case | % | Повідомлення системи

Рисунок 4.12 – Завантаження баз знання для отримання рішення

Для реалізації цього підходу були розроблені спеціальні функції на мові програмування C# (додаток А). Ці функції забезпечують завантаження, обробку та представлення даних у графічному інтерфейсі, що включає функціонал для зчитування інформації з ресурсних файлів, її відображення в таблицях, а також можливості для подальшого аналізу та використання даних у процесі прийняття рішень.

Такий підхід до реалізації блоку отримання рішень не лише полегшує взаємодію користувачів з системою, але й забезпечує гнучкість та ефективність у роботі з великим обсягом інформації, що є важливим для успішної реалізації завдань експертної системи.

У програмі реалізовано два основні функціонали, що дозволяють ЛПР ефективно працювати над пошуком рішення за виводом по прецедентах.

По-перше, програма надає ЛПР можливість отримати сценарій рішення, який складається з декілька планів.

При заповненні анкети Студент обирає дисципліни за вибором, які йому цікаві або важливі для його майбутньої кар'єри. При створенні БППС система враховує ці данні.

Далі додаток аналізує вибрані студентом дисципліни та порівнює їх з темами дипломів із бази прецедентів. На основі збігів між вибраними дисциплінами та темами дипломів обчислюється коефіцієнт близькості за методом триплетів. В залежності від отриманого коефіцієнту близькості та інших параметрів - рівень знань студента та науковий досвід попередніх робіт), програма розраховує різні сценарії призначення наукового керівника.

Остаточне рішення, яке обирає ЛПР, зберігається в ресурсний файл системи для подальшої роботи (рис.4.13).

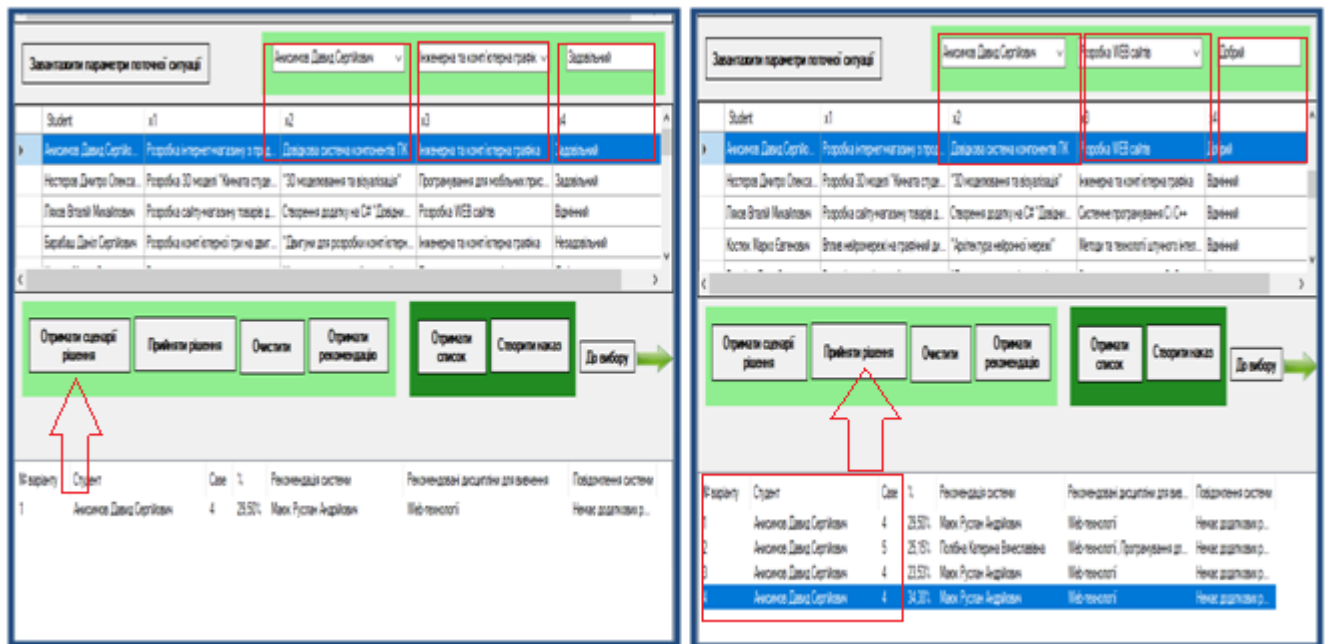


Рисунок 4.13 – Отримання сценарію рішення

В результаті одиночного пошуку по базі прецедентів, ЛПР отримує відсоток схожості поточної ситуації окремого студента з наявними прецедентами. Для розрахунку цієї схожості використовується метод триплетів, методика якого детально описана у другому розділі дослідження. В результаті, ЛПР отримує сценарій рішення з планами за відсотком схожості, які включають рекомендованого наукового керівника та рекомендації стосовно того, на які дисципліни студенту слід звернути увагу при роботі над дипломним проектом. Крім того, сформований список має обмеження вибору рішення для ЛПР, щодо індивідуальної характеристики викладача, який не має досвіду роботи зі студентами, які не встигають. Такий підхід дозволяє сформувані індивідуальні рекомендації, які враховують унікальні особливості та потреби кожного студента. Система може запропонувати не тільки найбільш відповідних наукових керівників, які спеціалізуються на схожих тематиках досліджень або мають необхідний досвід для підтримки студента у його науковій роботі, але і рекомендувати студенту набір основних предметів, які варто вивчати для глибокого розуміння обраної теми дипломного проекту.

По-друге, програма надає ЛПР можливість здійснити оцінку схожості бази прецедентів з масивом записів поточної ситуації для всіх студентів групи одночасно. Отримані результати виводяться у вигляді відсотка схожості масиву записів поточної ситуації з наявними прецедентами за метрикою коефіцієнту Танімото

Критично оцінюючи такий підхід, слід зазначити, що такий підхід має формальний характер і обмежений у глибокому аналізі індивідуальних потреб студентів. Він лише забезпечує можливість автоматизованого формування вихідного документу – наказу про призначення наукових керівників та тем ДП. Але при проведенні експерименту було отримано достатньо близький результат до попереднього (рис.4.14).

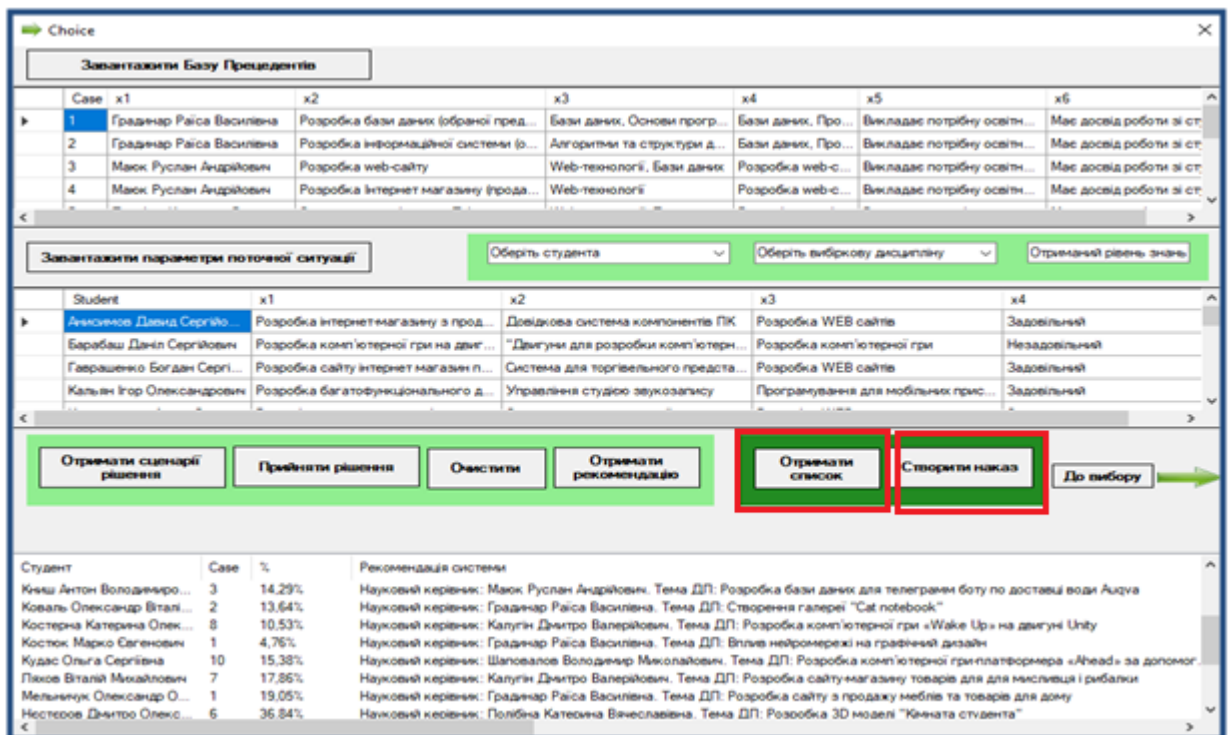


Рисунок 4.14 – Пакетний пошук рішення за прецедентами

Отже, для досягнення більш високої ефективності і точності в процесі вибору рішення, необхідно шукати схожості та враховувати індивідуальні характеристики кожного студента окремо.

Остаточне рішення, яке обирає ЛПР, зберігається в ресурсний файл системи за допомогою спеціальної функції на мові програмування С# (додаток А) для подальшої роботи.

4.2.4 Підсистема візуалізації прийнятого рішення ЛПР

Підсистема візуалізації включає три функціонала:

- автоматична генерація чернеток наказу про призначення наукових керівників та тем ДП;
- автоматична генерація рекомендаційного листа студенту, який можна відіслати на e-mail;
- автоматична генерація індивідуального плану студента на II семестр навчального року.

Вище вказані документи генеруються прототипом експертної системи у вигляді файлів формату MS Word (Додаток Б). Ці файли можна переглядати, редагувати та зберігати на комп'ютері завідуючого відділенням для подальшого опрацювання. Автоматизація цих процесів реалізована за допомогою функцій на мові С# (додаток А).

Отримання індивідуального плану виконується окремою підсистемою ЕС ІППР (рис.4.15).

Блок генерації індивідуального плану студента складається з двох таблиць у графічному інтерфейсі програми. Перша таблиця призначена для відображення даних із ресурсного файлу, що містить інформацію про прийняті рішення ЛПР.

Друга таблиця призначена для завантаження даних з Excel-файлу, що містить інформацію про місця проходження практики студентами у минулих роках. Цей файл містить дані про підприємства, установи або організації, де студенти вже працювали на практиці, а також відповідні контактні данні у вигляді адрес сайтів цих установ.

Попередній досвід підтвердив успішність проходження студентами практики на зазначених підприємствах та сприяння удосконаленню їхніх знань та практичних навичок. Такий Excel-файл виступає своєрідним примітивним аналогом бази прецедентів минулого року.

Завантажити рекомендацію

Student	x1	x2
Шербань Володимир Володимирович	Маюк Руслан Андрійович	Web-технології, Базы даних
Ядренцев Іван Олегович	Градинар Раїса Василівна	Алгоритми та структури даних
Соломонок Данило Володимирович	Маюк Руслан Андрійович	Web-технології
Рашупкін Павло Сергійович	Полібіна Катерина Вячесл...	Web-технології, Програмування для
Нестеров Дмитро Олександрович	Полібіна Катерина Вячесл...	Інженерна та комп'ютерна графіка,

Завантажити базу практик

Підприємство	Сайт
ТОВ "ВЕБ ПРОДАКШН"	сайт:https://wproduct.net/games-development
ТОВ «КВАРТ-СОФТ»	https://opendatabot.ua
ТОВ «СОФТПАЙН ІТ»	https://www.softline.kiev.ua
ТОВ «ВЕБАГЕНСІЯ ЧЕРВОНИЙ ХАМЕЛЕОН»	https://redchameleon.com.ua
ТОВ «ЮБІСОФТ УКРЕЙН»	https://www.ubisoft.com

До вибору Зробити вибір Отримати індивідуальний план

Студент	Викладач	Дисципліни	Підприємство	Сайт
Ядренцев Іван Олегович	Градинар Раїса Василівна	Алгоритми та с...	ТОВ «ВЕБАГЕ...	https://redcham...
Рашупкін Павло Сергійович	Полібіна Катерина Вячесл...	Web-технології,...	ТОВ "ВЕБ ПРО...	сайт:https://wpr...
Шербань Володимир Вол...	Маюк Руслан Андрійович	Web-технології,...	ТОВ «КВАРТ-С...	https://opendat...

Рисунок 4.15 – Генерація індивідуального плану студента

Програма не виконує інтелектуального аналізу для оцінки схожості обраного рішення та місця практики для кожного окремого студента. Замість цього система накладає обмеження, що дозволяє обирати лише одне місце практики для кожного студента (відповідна функція мовою C# наведена в додатку А). Такий підхід спрощує процес вибору та уникнення можливих конфліктів у призначенні місць практик, але при цьому не враховує всі можливі варіанти для кожного студента в індивідуальному порядку, бо не потребує цього.

Після отримання місця практики для кожного студента, система генерує індивідуальний навчальний план на основі простого правила «якщо – то». Це означає, що для кожного студента формулюються конкретні навчальні завдання та активності залежно від обраного наукового керівника. Індивідуальний навчальний план має наступні складові:

- ПІБ студента;
- графік навчального процесу у II семестрі поточного навчального року;
- місце практики, контактні дані керівника практики;
- рекомендовані дисципліни для вивчення;
- які навички та вміння набуваються під час практики;
- де і як можна застосувати набути навички та вміння;
- посилання на методичні матеріали;
- контактні дані наукового керівника та графік консультацій.

Ці складові допомагають студентам систематизувати свої академічні зусилля та планувати навчальні та практичні заходи для досягнення успіху у своєму навчанні та підготовці до майбутньої кар'єри. Також, індивідуальний навчальний план стає ефективним інструментом для досягнення успіху у навчанні та дослідженнях під керівництвом обраного наукового керівника. Всі генеровані вихідні дані представлено в додатку Б.

Таким чином, прототип ЕС ІППР «Choice», яка описана вище, є добре структурованою та включає всі необхідні компоненти для ефективної підтримки прийняття рішень ЛПР у навчальному процесі.

Інтерфейс користувача дозволяє легко навігувати по системі, вводити дані, виконувати пошук та переглядати результати, що робить процес прийняття рішень більш ефективним.

Використання файлів MS Excel для зберігання даних робить систему доступною та легкою для розуміння. Це дозволяє швидко змінювати та оновлювати інформацію без потреби в складних маніпуляціях з даними.

Включення експертної інформації з коректними рішеннями є ключовим для розробки надійної бази прецедентів та бази параметрів поточної ситуації. Це допомагає забезпечити точність та релевантність рекомендацій, які надає система.

Інтеграція бази прецедентів і бази параметрів поточної ситуації є фундаментальною для визначення оптимальних рішень. Використання різних метрик, таких як коефіцієнт Танімото та метод триграм, дозволяє системі точно ідентифікувати та аналізувати релевантні прецеденти, забезпечуючи високу точність рекомендацій.

Можливість зберігання результатів у ресурсні файли та файли MS Word додає гнучкості системі, дозволяючи легко документувати та ділитися рекомендаціями.

У цілому, прототип ЕС ІППР «Choice» надає необхідні інструменти для підтримки прийняття обґрунтованих рішень у слабоформалізованих умовах навчального процесу.

4.3 Вплив впровадження ЕС ІППР «Choice» на навчальний процес

Автоматизація процесу вибору наукового керівника та тем дипломних робіт в НКПФК має на меті вирішити ряд завдань, які полегшують і оптимізують прийняття рішень для різних зацікавлених сторін:

- керівникам відділень ефективно управляти розподілом ресурсів та завантаженістю викладачів. Це допомагає впевнитися, що студенти отримують керівників, які найкраще відповідають їхнім науковим інтересам та навчальним потребам, водночас забезпечуючи баланс робочого навантаження серед викладацького складу. Експертні системи можуть також сприяти забезпеченню прозорості та справедливості процесу вибору;
- викладачам краще планувати свою наукову та педагогічну діяльність, розподіляючи час між науковим керівництвом та іншими завданнями. Викладачі можуть ефективніше керувати своїми обов'язками, знаючи, яка кількість студентів і з якими темами дипломних робіт вони будуть працювати;
- студентам надає інструменти для вибору наукового керівника та теми дипломної роботи, які найкраще відповідають їх особистим і професійним цілям,

більш відповідально та усвідомлено підходити до формування своєї освітньої траєкторії, максимізуючи користь від навчального процесу. Зокрема, система може враховувати попередній науковий досвід студентів, їх академічні успіхи, інтереси та переваги, сприяючи більш точному підбору керівника, що в свою чергу може позитивно вплинути на якість дипломної роботи та подальшу кар'єру.

З метою порівняння традиційного способу прийняття рішення та автоматизованої підтримки прийняття рішення ЛПР, в особі завідуючого відділенням, було створено таблицю (таблиця 4.2).

Таблиця 4.2

Порівняння процесів прийняття рішення

Критерій	Традиційний процес прийняття рішень	Автоматизований процес прийняття рішень
Час	Високозатратний за часом процес	Низькозатратний за часом процес
Точність	Залежить від досвіду ЛПР	Висока точність результатів
Об'єктивність	Залежить від суб'єктивного судження ЛПР	Об'єктивна, неупереджена оцінка
Ефективність	Залежить від швидкості та досвіду ЛПР	Виключає фактори суб'єктивності
Споживання ресурсів	Великі витрати людських ресурсів	Менші витрати людських ресурсів
Складність реалізації	Не вимагає спеціальних знань	Не вимагає спеціальних знань від ЛПР
Масштабованість	Обмежена, складно масштабувати	Легко масштабується для великої кількості даних та користувачів

Ця таблиця надає порівняльний огляд традиційного та автоматизованого процесів підтримки прийняття рішень ЛПР за різними критеріями.

З метою отримання об'єктивної інформації про рівень задоволення здобувачів передфахової вищої освіти методами навчання і якістю викладання, організаційною, інформаційною та консультативною підтримкою наприкінці навчального року у

НКПФК проводиться анкетування. Анкетування учасників освітнього процесу здійснюється за допомогою електронних анкет. Опрацювання анкет відбувається автоматично в електронному форматі. Анкетування здебільшого є анонімним і вибіркоким, до процедури якого долучається не менше ніж 60-70% учасників освітнього процесу.

Відповіді на запитання оцінювалися за десятибальною шкалою (1-10), де: «1» – незадовільно, «10» – відмінно.

Питання блоку, які стосуються даного дослідження:

1. Наскільки ви були задоволені швидкістю та ефективністю процесу призначення наукового керівника?
2. Чи був процес вибору наукового керівника прозорим та зрозумілим для вас?
3. Чи виникала необхідність у перегляді прийнятого рішення?
4. Чи отримали ви додаткові рекомендації?
5. Чи була співпраця з науковим керівником корисна та плідна для вас?
6. Чи задоволені ви призначеним місцем практики?
7. Чи вважаєте ви, що впровадження експертної системи забезпечить більш об'єктивний та справедливий відбір наукового керівника в порівнянні з вибором вручну?

В проведеному експерименті взяли участь студенти-випускники двох відділень НКПФК.

На відділенні Комп'ютерної інженерії призначення наукових керівників проєктів та місць проходження практики виконувалося в традиційний спосіб. Кількість студентів - 38

На відділенні Програмної інженерії цей процес відбувався за допомогою розробленої експертної системи «Choice». Кількість студентів – 32.

Студентів не було проінформовано про проведення експерименту призначення наукових керівників в різний спосіб.

Результати анкетування (рис.4.16) які відокремлені від загальних результатів анкетування, вказують на значне покращення швидкості та ефективності процесу призначення наукового керівника з використанням експертної системи.

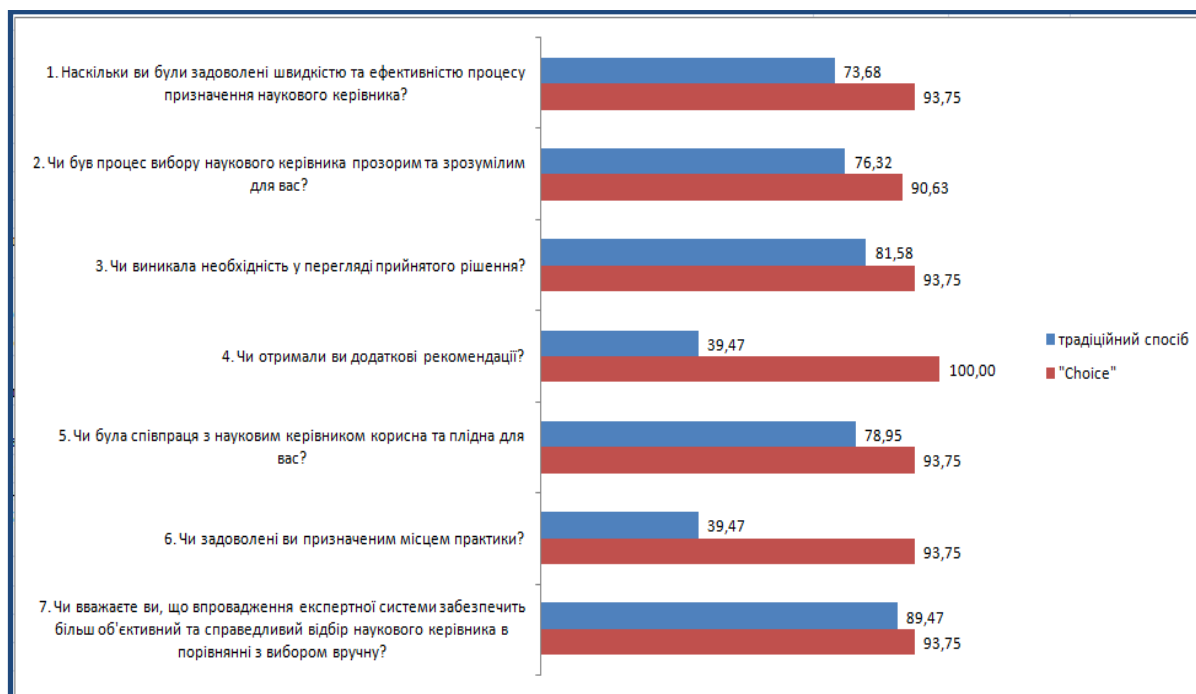


Рисунок 4.16 – Результати анкетування

Учасники анкетування високо оцінили прозорість та зрозумілість цього процесу, що свідчить про вдале впровадження нової системи. Більшість відповідей не відчували потреби у перегляді прийнятого рішення, що свідчить про його адекватність та відповідність вимогам студентів.

Деякі студенти отримали додаткові рекомендації, що може свідчити про індивідуальний підхід при виборі наукового керівника та підтримку з боку системи. Співпраця з науковим керівником була визнана більш корисною та плідною для студентів у випадку використання експертної системи.

Щодо задоволеності призначеним місцем практики, результати були розділені. Проте, загальний рівень задоволеності серед користувачів експертної системи був вищий.

Найважливіше, що більшість студентів вважає, що впровадження експертної системи забезпечить більш об'єктивний та справедливий відбір наукового керівника в порівнянні з вибором в традиційний спосіб.

Доцільність впровадження ЕС ІППР «Choice» в навчальний процес НКПФК ґрунтується на порівнянні результатів захистів дипломних робіт до та після впровадження автоматизованого процесу підтримки прийняття рішення.

Згідно звітів «Про роботу Державної екзаменаційної комісії із захисту дипломних проєктів за спеціальністю 121 Інженерія програмного забезпечення» в НКПФК [127] перед впровадженням ЕС ІППР «Choice», результати захистів наведено нижче (рис.4.17).

Результати захисту Дипломних проєктів на відділенні ПІ:

Групи/Показник	Якісна успішність %	Абсолютна успішність%
П18-1Д	40%	90%
П18-2Д	60%	100%
Загальний показник захисту ДП	50%	95%

Результати захисту Дипломних проєктів на відділенні ПІ:

Групи/Показник	Успішність «5» %	Успішність «4» %	Успішність «3» %
П18-1Д	4 – 20%	4 – 20%	10 – 50%
П18-2Д	5 – 27%	6 – 33%	7 – 38%
Загальний показник захисту ДП	23,5%	26,5%	44%

Рисунок 4.17 – Результати захисту ДП на відділення ПІ у 2021-2022 н.р.

В контексті даного дослідження, результати можна розглядати як результат формування невдалих тандемів наукового керівника та студента, які виникли через суб'єктивні уявлення ЛПР. Це підтверджується тим, що двом студентам не вдалося успішно виконати дипломну роботу (абсолютна успішність в однієї з груп 90%).

Проте після впровадження в наступному навчальному році ЕС ІППР «Choice» індивідуалізованого підбору наукових керівників було відзначено підвищення якості виконання та захисту дипломних робіт та зростання академічних досягнень студентів (отримання дипломів з відзнакою). В контексті даного дослідження, можна казати про те, що це стало можливим завдяки поліпшенню якості вибору наукових керівників, що в свою чергу сприяло утворенню успішних тандемів наукового керівника та студента. (рис.4.18) .

Результати захисту Дипломних проєктів на відділенні ПІ:

Групи/Показник	Якісна успішність %	Абсолютна успішність%
П19-1Д	58%	100%
П19-2Д	79%	100%
Загальний показник захисту ДП	71%	100%

Результати захисту Дипломних проєктів на відділенні ПІ:

Групи/Показник	Успішність «5» %	Успішність «4» %	Успішність «3» %
П19-1Д	7 – 37%	4 – 21%	8 – 42%
П19-2Д	5 – 36%	6 – 43%	3 – 21%
Загальний показник захисту ДП	36,5%	32%	33%

Диплом «з відзнакою» отримали 7 студентів, що складає 21% від загальної кількості студентів.

Рисунок 4.18 – Результати захисту ДП на відділення ПІ у 2022-2023 н.р.

Діаграма порівняння показників успішності захисту дипломних проєктів в 2021-2022 та 2022-2023 навчальних роках наведено нижче (рис.4.19) .

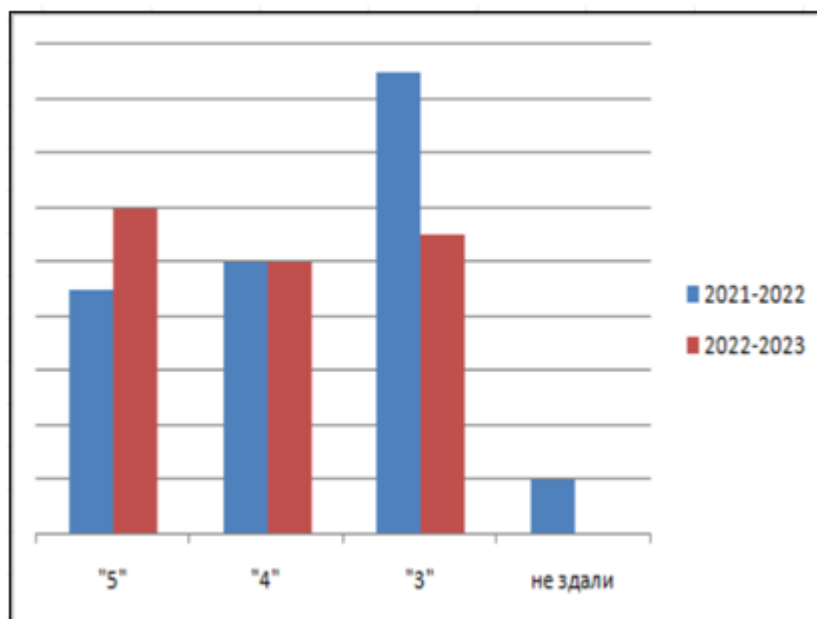


Рисунок 4.19 – Діаграма порівняння показників успішності

Діаграми порівняння абсолютних та якісних показників захисту дипломних проєктів в 2021-2022 та 2022-2023 навчальних роках (рис.4.20) .

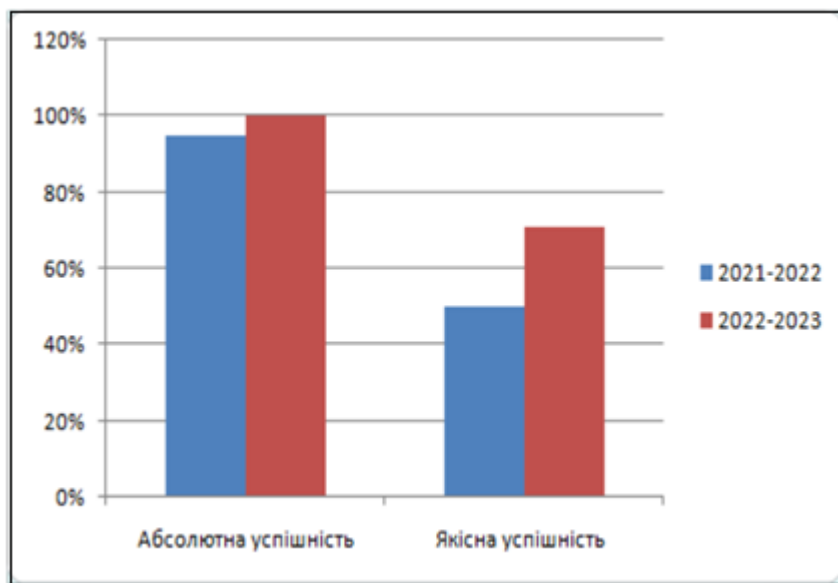


Рисунок 4.20 – Діаграма порівнянь абсолютних та якісних показників

4.4 Висновки до четвертого розділу

Підсумовуючи четвертий розділ дисертаційного дослідження з впровадження ЕС ІППР «Choice» по вибору наукового керівника та теми дипломного проєкту студента, можна зробити наступні висновки.

Розроблена та впроваджена в навчальний процес НКПФК система дозволяє ефективно вирішувати завдання автоматичної генерації ранжованого списку рішень ЛПР без суб'єктивного впливу на нього. Також запропонований прототип дозволяє автоматично генерувати освітні документи, такі як накази, рекомендаційні листи та індивідуальні плани студентів без витрачання надлишкового часу.

В процесі вирішення цього завдання враховуються як об'єктивні параметри, так і індивідуальні потреби кожного студента.

Проведені експерименти показали, що в результаті автоматизованої підтримки прийняття рішення ЛПР, зменшується ймовірність невдалих тандемів між студентами та керівниками, а також підвищується якість та академічні досягнення студентів. Такий підхід сприяє оптимізації навчального процесу, забезпечуючи кращу відповідність між компетенціями наукових керівників та потребами студентів.

ВИСНОВКИ

Дисертаційна робота вирішує важливе і актуальне науково-прикладне завдання підтримки вибору індивідуальної освітньої траєкторії шляхом розробки нових моделей і методів побудови експертних систем інтелектуальної підтримки прийняття рішень в освіті.

Основні наукові і практичні результати полягають у наступному:

1. Встановлено, що на поточний момент часу існує нагальна потреба в створенні сучасних експертних інтелектуальних систем підтримки прийняття рішення в слабоформалізованій освітньої галузі, які враховують інтелектуальний аналіз даних та попередній досвід людини, що приймає рішення, і забезпечують прийняття нею рішень в умовах невизначеності; які здатні ефективно вирішувати окрему частку питань, пов'язаних із процесами формування рішень при управлінні навчанням та зниженням впливу людського фактору на цей процес.

2. З метою підвищення ефективності функціонування системи підтримки вибору індивідуальної освітньої траєкторії запропоновано зменшувати обсяг інформації, яка повинна бути прийнята до обробки, шляхом експертних ухвалень. Запропонований метод класифікації даних надає можливість використання інтелектуального аналізу освітніх даних без надлишкового «занурення» людини, яка приймає рішення, у деталі базових даних або проміжної аналітики.

3. Запропоновано підхід до створення експертної інтелектуальної системи підтримки рішення на основі інтеграції методів інтелектуального аналізу освітніх даних та сценарно-прецедентного підходу, що дозволяє накопичувати, систематизувати, інтегрувати та ефективно використовувати ретроспективні і поточні дані та досвід експертів. Ця інтеграція сприяє поліпшенню процесу формування бази прецедентів із використанням модифікованого CBR циклу. Модифікований CBR цикл, що використовує експертну інформацію (тестові набори даних) для отримання прецедентів, підвищує релевантність отриманих рішень на

основі прецедентів за рахунок формування бази вдалих (підходящих) прецедентів у процесі виконання CBR циклу.

4. Запропонований алгоритм отримання рішення на основі метрики коефіцієнта Жаккара-Танімото дає змогу реалізувати ефективний метод оцінювання схожості поточного прецедента з кожним прецедентом у базі даних на основі їх аналогії з більш схожими випадками з минулого.

5. Запропоновано використання комбінованої методики формування функції вибору ЛПР, що ґрунтується на методу триплетів, для отримання ранжируваного списку релевантних рішень та зниження впливу фактору невизначеності на ЛПР.

6. Вперше розроблено інформаційну технологію підтримки індивідуалізованого вибору ЛПР наукового керівника дипломного проєкта студентів з урахуванням більш точного зіставлення їх академічних та професійних переваг з компетенціями та експертизою потенційних наукових керівників.

7. Запропоновано адаптивний вибір ЛПР місця практики, який дозволяє студентам знаходити оптимальні варіанти розвитку своїх професійних навичок.

8. Практична апробація створеного прототипу дозволила підвищити якість прийняття рішення індивідуалізованого вибору наукового керівника та теми дипломного проєкта студентів з урахуванням більш точного зіставлення їх академічних та професійних переваг з компетенціями та експертизою потенційних наукових керівників; запропоновано адаптивний вибір місця практики, який дозволяє студентам знаходити оптимальні варіанти розвитку своїх професійних навичок; здійснено інтеграцію даних та зворотного зв'язку, який забезпечує студентів актуальною інформацією, дозволяючи їм ухвалювати рекомендації експертної системи; проаналізовано вимоги ринку праці в галузі ІТ, який стрімко змінюється, що забезпечує студентам можливість вибору таких керівників та місць практики, які максимально відповідають сучасним вимогам ІТ індустрії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Парпан У. Модернізація національної системи освіти у контексті сучасного євроінтеграційного поступу України. *Вісник Національного університету «Львівська політехніка»*. 2017. № 865. С. 296-302.
2. Головань М.С. Аналіз тенденцій розвитку вищої освіти в умовах глобалізації, інтеграції та інформатизації суспільства. *Проблеми сучасної педагогічної освіти*. 2011. №31. С. 203-211.
3. Про схвалення Стратегія розвитку вищої освіти в Україні на 2022—2032 роки: Розпорядження Кабінету Міністрів України від 23.02.2022 р. № 286-р. URL: <https://www.kmu.gov.ua/npas/pro-shvalennya-strategiyi-rozvitku-vishchoyi-osviti-v-ukrayini-na-20222032-roki-286-> (Дата звернення 20.03.2023)
4. Стюарт Рассел. Сумісний з людиною. Штучний інтелект і проблема контролю. Видавництво: BookChef, 2020. 416 с.
5. Кай-Фу Лі, Чень Цюфань. Штучний інтелект 2041: 10 передбачень для майбутнього. Видавництво: BookChef, 2022. 464 с.
6. Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи. Видавництво: Львівська політехніка, 2012. 168 с.
7. Баклан І.В. Експертні системи. Курс лекцій: Навчальний посібник. К.: НАУ, 2012. 132 с.
8. Schank R.C., Abelson R.P. Scripts, Plans, Goals and Understanding. Erlbau, 1977. 220 p.
9. Schank R.C. Dynamic Memory: A theory of reminding and learning in computers and people. Cambridge University Press, 1982. 200 p.
10. Watson I.D., Abdullah S. Developing case-based reasoning systems: a case study in diagnosing building defects. *IEE Colloquium on Case-Based Reasoning: Prospects for Applications*. 1994. №57. pp. 1-13.
11. Bench-Capon T.M., Coenen F. The maintenance of legal knowledge based systems. *AI Review*, 1992, pp. 129-143.

12. Nilsson M., Sollenborn M. Advancements and Trends in medical case-based reasoning: An overview of systems and system development. *17th Internatonal FLAIRS Conference*. 2004. pp. 178–183.

13. Simpson R.L. A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation. *Technical Report GIT-ICS*. 2015. pp. 180-185.

14. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник. Запоріжжя: ЗНТУ, 2008. 341 с.

15. Таран Т.А. Штучний інтелект. Теорія і застосування: Навчальний посібник. Луганськ: СНУ ім. В. Даля, 2006. 240 с.

16. Смірнов О.А. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем. ИНФРА-М, 2010. 65 с.

17. Досин Д.Г., Литвин В.В., Нікольський Ю.В., Пасічник В.В. Інтелектуальні системи, базовані на онтологіях. Львів: «Цивілізація», 2009. 414 с.

18. Мейтус В. Ю. Інтелектуальні системи, онтології та онтологічні простори. Київ, 2009. С. 414.

19. Розробка загальної апаратно-програмної архітектури розподіленої версії онтологічного порталу. *Інформаційні технології в освіті*. Херсон, 2009. №4. С. 164–178.

20. Запровадження інтелектуальних інформаційних технологій для систем підтримки прийняття рішень. *Інформаційні технології*. 2003. № 3. С. 36–41.

21. Setlak G. Intelligent decision support systems. *Вісник Одеського політехнічного університету*. 2001. №4. С. 219–222.

22. Інтелектуальні системи підтримки прийняття рішень: Навчальний посібник. Європейський університет. Київ, 2007. 335 с.

23. IBM Watson. Morning Dough - веб-сайт. URL: <https://www.morningdough.com/uk/ai-tools/ibm-wat> (Дата звернення 22.03.2023)

24. Вивчення нового Gemini від Google DeepMind: про що йдеться? UNITE.AI – веб-сайт. URL: <https://www.unite.ai/uk> (Дата звернення 21.12.2023)

25. Що таке Microsoft Azure. Найважливіше про глобальну хмарну платформу. KyivStar Bisness Hub – веб-сайт. URL: <https://hub.kyivstar.ua/news> (Дата звернення 10.02.2022)

26. Коротка історія ChatGPT та компанії Open AI. DOU – веб-сайт спільноти програмістів. URL: <https://dou.ua/forums/topic/44088> (Дата звернення 28.06.2023)

27. Чим займається компанія Palantir? AIN.UA - веб-сайт. URL: <https://ain.ua/2022/06/03/chym-zajmayetsya-kompaniya-palantir-ta-za-shho-yiyi-krytykuyut> (Дата звернення 03.06.2022)

28. Що таке Salesforce і чим вона цікава для досвідчених розробників. DOU – веб-сайт спільноти програмістів. URL: <https://dou.ua/lenta/articles/what-salesforce-is> (Дата звернення 31.01.2019)

29. Адаптивні технології: досвід Knewton. EdEra blog – веб-сайт. URL: <http://blog.ed-era.com/knewton> (Дата звернення 09.11.2023)

30. Coursera – веб-сайт. <https://www.coursera.org> (Дата звернення 20.11.2023)

31. Що таке Impact Canvas і чому обирати цю модель? Дія. Бізнес – веб-портал. URL: <https://business.diia.gov.ua> (Дата звернення 28.11.2023)

32. Інтелектуальні тьюторські системи і навчальні середовища. Stud.com.ua – веб-сайт. URL: <https://stud.com.ua/101446> (Дата звернення 30.11.2023)

33. Українська LMS-система. HRASSIST – веб-сайт. URL: <https://hrassist.pro/uk/employee-training> (Дата звернення 22.03.2023)

34. Quizlet – Сервіс для створення навчальних карток, тестів та ігор. ВЧИМО – веб-сайт. URL: <https://vchymo.com/application/Quizlet> (Дата звернення 22.03.2024)

35. Ігнатович О.М. Технології кар'єрного консультування викладачів та студентів професійно-педагогічного коледжу. *Digital Library NAES of Ukraine*. 2024. С. 1-8.

36. Освітня аналітика України – веб сайт. URL: <https://science.iea.gov.ua> (Дата звернення 22.03.2024)

37. Бурдаєв В.П. Системи навчання з елементами штучного інтелекту. Монографія. Харків: Вид. ХНЕУ, 2009. 400 с.
38. Verbeke, G. E-Learning in the workplace. *Journal of Workplace Learning*. 2003. №15, P. 312-318.
39. Verbeke, G., DeGeeter, K. Predictors of e-learning success: an exploratory study. *Computers in Human Behavior*. 2005. №21. P. 957-980.
40. Verbeke, G., DeGeeter, K. The impact of ICT on learning: A review of research. *Education and Technology for a Better World*. 2006. P. 699-704.
41. Verbeke, G., DeGeeter, K. Evaluating e-learning success: an integrative framework. *International Journal of Learning Technology*. 2007. №4. P. 412-432.
42. Toru Iyoshi and M. S. Vijaya Kumar. Opening Education: Collective Educational Development through Open Technology, Open Content and Open Knowledge. MIT Press, 2008. 400 p.
43. Holsapple, C.W. Decision Support Systems (a knowledge based approach). New York: West Publishing Company, 1983. 860p.
44. Гагарін О.О., Титенко С.В. Дослідження і аналіз методів та моделей інтелектуальних систем безперервного навчання. *Наукові вісті НТУУ «КПІ»*. 2007. №6. С. 37–48.
45. Терещук Г.В. Індивідуалізація і диференціація навчання студентів в умовах впровадження Інтернету речей в процесі навчання. *Науковий часопис Національного педагогічного університету імені М.П. Драгоманова*. 2019. №11. С. 108-112.
46. Терещук Г. В. Компетентісний підхід у роботі педагогічного університету. *Освіта*. 2013. № 45. С. 4.
47. Євтушенко Л. І., Спірін, О. М. Принципи та методи побудови індивідуальної освітньої траєкторії учня на основі компетентнісного підходу. *Педагогіка формування творчої особистості*. 2015. № 49. С.62-68.

48. Євтушенко Л. І. Побудова індивідуальної освітньої траєкторії як шлях до формування творчої особистості. *Науковий вісник Херсонського державного університету*. 2018. №2. С. 54-59.

49. Іванюк І. Персоналізація навчання в умовах використання інформаційних технологій. *Науковий вісник Миколаївського державного університету імені В. О. Сухомлинського*. 2020. №39. С. 131-135.

50. Литвиненко О., Яковенко О. Використання інтелектуальних технологій для розробки індивідуальної траєкторії навчання. *Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, досвід, перспективи*. 2021. №19. С. 79-83.

51. Вакалюк Т. В. Індивідуалізація навчання іноземної мови у вищих навчальних закладах засобами комп'ютерної технології. *Вісник Чернівецького національного університету імені Юрія Федьковича*. 2017. №7. С. 63-69.

52. Городніча Т. В. Розвиток індивідуальної освітньої траєкторії учнів з іноземних мов у вищих навчальних закладах. *Вісник Херсонського державного університету*. 2018. №2. С. 106-110.

53. Хорошкова О. Реалізація індивідуально-орієнтованого підходу у навчанні програмування на основі компетентнісного підходу. *Наукові записки Національного університету «Острозька академія»*. 2017. № 3(47). С. 33-38.

54. Бугиль М. Використання індивідуальної освітньої траєкторії в навчанні програмування на основі методики диференційованого навчання. *Вісник Житомирського державного університету імені Івана Франка*. 2020. № 2(96). С.92-96.

55. Іванова І. Використання індивідуально-орієнтованого навчання у процесі підготовки фахівців з програмування. *Науковий вісник Миколаївського національного університету імені В. О. Сухомлинського*. 2018. №2(2). С. 68-72.

56. Лазарева Ю., Скрипник І. Використання індивідуальної освітньої траєкторії у процесі навчання програмування з використанням онлайн-курсів. *Вісник Одеського національного університету імені І. І. Мечникова*. 2019. №24(2). С. 46-53.

57. Діденко О. Використання індивідуальної освітньої траєкторії у навчальному процесі з програмування. *Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми*. 2017. №1(51). С. 102-107.

58. Гончар О. В. Індивідуалізація процесу підготовки студентів до участі у конкурентній боротьбі на ринку праці. *Науковий вісник Національного університету біоресурсів і природокористування України*. 2019. №29. С. 164-171.

59. Іванова О. М. Індивідуальна освітня траєкторія як засіб забезпечення конкурентоспроможності випускника вищого навчального закладу. *Науковий вісник Миколаївського державного університету імені В. О. Сухомлинського*. 2016. №3(1). С.111-117.

60. Мальований О. В. Індивідуалізація освітнього процесу як засіб забезпечення конкурентоспроможності студентів вищих навчальних закладів. *Наукові записки Кіровоградського національного технічного університету*. 2018. №30. С.168-174.

61. Беспалько О. Персоналізація навчання на основі штучного інтелекту. *Вісник Чернігівського національного педагогічного університету імені Т. Г. Шевченка*. 2020. № 170(1). С. 83-89.

62. Данилюк О. Особистісно орієнтований підхід у системі вищої освіти на основі технологій штучного інтелекту. *Освіта та розвиток обдарованої особистості*. 2019. №12(31). С.31-36.

63. Гуцол С., Хорошкевич В. Використання технологій штучного інтелекту в освіті. *Східноєвропейський журнал передових технологій*. 2020. №4(10). С.20-25.

64. Гавриленко Н. С., Деріга Л. М. Формування індивідуальної освітньої траєкторії учня засобами віртуального навчального середовища. *Інформаційні технології і засоби навчання*. 2018. №68(6). С.14-23.

65. Калинова І. В., Шаповал О. В. Розвиток індивідуальної освітньої траєкторії студента на основі застосування технологій віртуальної реальності. *Науковий вісник Миколаївського національного університету імені В.О. Сухомлинського*. 2020. № 1 (22). С. 114-119.

66. Благодар І. В., Войтенко Т. В., Запорожець О. О. Індивідуалізація та диференціація навчання студентів в умовах впровадження Інтернету речей в процесі навчання. *Науково-педагогічні проблеми інформатизації*. 2019. №39. С. 59-71.

67. Плаксина, О. О., Шевелева, Л. В. Інтернет-речі в освіті: практика та перспективи застосування. *Науковий вісник Національного університету біоресурсів і природокористування України*. 2018. № 34(2). С. 154-163.

68. Kasik D. J., Minas M. K. The evolution of software engineering education: A global perspective. *IEEE TransactionsoncEducation*. 2017. №60(1). P. 3-10.

69. Diesta O, Juristo N, Martínez-Fernández S. Improving software engineering education through model-based and technology-based learning: A systematic review. *IEEE Transactionson Education*. 2017. №60(4). P. 271-280.

70. COURSERA - веб-сайт. URL: <https://www.coursera.org/learn> (Дата звернення 19.06.2023)

71. UDACITY – веб-сайт. URL: <https://www.udacity.com> (Дата звернення 19.06.2023)

72. Cioca, L.-I. Sustainability of youth careers in Romania - a study of the relationship between personal interests of students with the chosen specialty at the university. *Sustainability*. 2021. №13.P. 15-20.

73. Archibald DE, Graham CR, Larsen R. Validation of a blended teaching readiness instrument for primary/secondary school teachers. *British Journal of Educational Technology*. 2021. № 52. P. 536-551.

74. Федорова В. О. Критичний аналіз моделей реалізації індивідуальної освітньої траєкторії здобувача вищої освіти. *Вісник Черкаського національного університету імені Богдана Хмельницького*. 2020. №4. С. 152–157.

75. Гуменюк Н. Створення індивідуальної освітньої траєкторії для студентів-програмістів на основі системи дистанційного навчання. *Наукові записки Тернопільського національного педагогічного університету імені Володимира Гнатюка*. 2020. №2. С.20-25.

76. Батрак О. В., Грушевська Г. В., Лавренюк І. А. Особливості застосування системи ІОТ у навчальному процесі ВНЗ з викладання дисципліни ООП. *Наукові записки НаУКМА. Технічні науки*. 2018. №14. С.10-17.

77. Григор'єв О., Шерстюк О. Формування індивідуальної освітньої траєкторії студента в процесі підготовки кадрів за спеціальністю «Програмна інженерія». *Науковий вісник Міжнародного гуманітарного університету*. 2020. №4(8). С.24-30.

78. Корольчук Т., Коваленко А. Досвід реалізації індивідуальної освітньої траєкторії в навчанні програмістів. *Наука і освіта*. 2017. №5(10). С.15-20.

79. Закон України про вищу освіту (Відомості Верховної Ради (ВВР), 2014, № 37-38,ст.2004). Верховна Рада України – веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/1556-18#Text> (Дата звернення 18.08.2023)

80. Закон України про фахову передвищу освіту (Відомості Верховної Ради (ВВР), 2019, № 30, ст.119). Верховна Рада України – веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/2745-19#Text> (Дата звернення 18.08.2023)

81. Затвердженні стандарти. 015 Професійна освіта. (Наказ № 802 від 13.07.2021 р.). Міністерство освіти і науки України – веб-сайт. URL: <https://mon.gov.ua/ua/osvita/fahova-peredvisha-osvita/sektor-fahovoyi-peredvishoyi-osviti/zatverdzheni-standarti> (Дата звернення 18.08.2023)

82. Національна рамка кваліфікацій. *Європейський фонд освіти*. 2021. С.1-24.

83. Дудикевич В. Б. Багаторівневі інтелектуальні системи керування: гарантоздатність, безпека об'єктів. *Системи обробки інформації*. 2015. №4. С. 92–95.

84. Максименко В. П. Дидактика: курс лекцій: Навчальний посібник. ХмЦНП, 2013. 222 с.

85. Полібіна К.В. Дослідження моделей, методів та інформаційних технологій оцінки кваліфікації програміста. *Проблеми інформаційних технологій*. 2020. № 27, С. 77-88.

86. Данчук В. Д. Інтелектуальні інформаційні системи управління науковими проектами. *Управління проектами, систем. аналіз і логістика*. 2013. №11. С. 21–27.

87. Лубко Д.В., Зінов'єва О.Г., Шаров С.В. Проектування та розробка експертної системи діагностування несправностей транспортних засобів. *Системи обробки інформації*. 2019. №1(156). С. 15–21.

88. Барабаш О. В., Ленков С. В., Лепіх Я. І. Універсальна модель процесу функціонування інформаційного об'єкта в інтелектуальній системі управління. *Сенсорна електроніка і мікросистемні технології*. 2010. № 3. С. 87–90.

89. Шерстюк В.Г. Моделі, методи та інформаційні технології сценарно-прецедентного управління полієргатичними системами. Автореферат: дисертація доктора технічних наук. Херсонський національний технічний університет. Херсон, 2014. 48 с.

90. Лисецький Ю. М. Інформаційні системи і технології в менеджменті. Інститут проблем математичних машин і систем. Монографія. Київ : Логос, 2014. 417 с.

91. Сукач М. К. Прикладні системи обробки інформації : навчальний посібник для студентів галузей знань «Економіка і підприємництво», «Менеджмент і адміністрування», «Культура». Київ : КСУ, 2011. 209 с.

92. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. *Перша науково-технічної конференція, 13-14 груд. 2010 р.* Харків, 2010. 104 с.

93. Пашкевич М. С. Когнітивна економіка: проблеми та перспективи для регіонів. *Бізнес Інформ*. 2014. № 12. С. 8–13.

94. Інтелектуальні системи прийняття рішень та прикладні аспекти інформаційних технологій. ISDMIT'2007 : зб. наук. пр. міжнар. наук. конф., 14-18 трав. 2007 р. Євпаторія, 2007. № 2. 260 с.

95. Гриценко В. І. Інтелектуальні інформаційні технології: перспективи розвитку та застосування. *HTI*. 2000. № 3. С. 9–10.
96. Кравцова Н.В., Ходаківська О.В., Кравцов Г.О. Роль штучного інтелекту як інтелектуального агента в інтерактивному навчанні студентів. *The III International Science Conference on E-Learning and Education*. Lisbon, Portugal. С. 255–258.
97. Кузіков Б.О. Модель адаптивної навчальної системи із використанням інтелектуальних агентів та технології Google Wave. *Вісник СумДУ. Серія «Технічні науки»*. 2010. №3. С. 68–77.
98. Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту : зб. наук. пр. міжнар. наук. конф. ISDMCI'2013. НАН України, Ін-т кібернетики ім. В.М. Глушкова, Півд. наук. центр НАН України, Херсон. нац. техн. ун-т, Люблін. ун-т технологій. Херсон : ХНТУ, 2013. 579 с.
99. Шаров С.В. Сучасний стан розвитку інтелектуальних інформаційних систем. *Вісник Чернігівського національного педагогічного університету. Серія: Педагогічні науки*. 2015. №130. С. 111–114.
100. Метешкін К. О. Методологічні основи автоматизованого навчання фахівців з використанням інтелектуальних інформаційних технологій. Національний аерокосмічний університет ім. М.Є. Жуковського. Харків, 2006. 33 с.
101. Шерстюк В.Г. Формальна модель гібридної сценарно-прецедентної СППР. *Автоматика. Автоматизація. Електротехнічні комплекси та системи*. 2004. № 1. С. 114-122.
102. Шерстюк В.Г., Жарікова М.В. Модель індивідуальних траєкторій навчання в інтелектуальній навчальній системі підтримки академічної сумлінності. *Вісник Херсонського національного технічного університету*. 2016. № 4(59). С.218-224.
103. Федорук П. І. Адаптивна система дистанційного навчання та контролю знань на базі інтелектуальних Інтернет-технологій. Прикарпат. нац. ун-т ім. В. Стефаника. Івано-Франківськ, 2008. 326 с.

104. Федорук П. І. Використання адаптивних та інтелектуальних технологій в системах дистанційного навчання. *Управляючі системи та машини*. 2006. № 5. С. 68–76, 84.
105. Піоварчик О. В. Інтелектуальні help-системи як засіб навчання технологіям програмування. *Інформаційні технології в освіті*. 2010. №5. С. 139–144.
106. Довбиш А.С., Васильєв А.В., Любчак В.О. Інтелектуальні інформаційні технології в електронному навчанні. Суми: Сумський державний університет, 2013. 177 с.
107. Прийма С.М., Строкань О.В., Лубко Д.В., Мозговенко А.А. Інформаційна система семантичного опрацювання результатів неформального й інформального навчання. *Праці ТДАТУ*. 2020. №20(3). С. 282–292.
108. Кравцова Н.В., Ходаківська О.В., Кравцов Г.О. Роль штучного інтелекту як інтелектуального агента в інтерактивному навчанні студентів. *The III International Science Conference on E-Learning and Education (February 2 – 5, 2021, Lisbon, Portugal)*. С. 255–258.
109. Кирик В. В. Комп'ютерно-інтегровані технології управління на основі нечіткої логіки. Київ, 2008. 198 с.
110. Братушка С.М., Новак С.М, Хайлу С.О. Системи підтримки прийняття рішень: навчальний посібник для самостійного вивчення дисципліни. Суми : ДВНЗ «УАБС НБУ», 2010. 265 с.
111. Галузинський, Г. П. Сучасні технологічні засоби обробки інформації. К. : КНЕУ, 1998. 224 с.
112. Пушкар, О. І. Системи підтримки прийняття рішень. Харків : Інжек, 2006. 304 с.
113. Ситник, В. Ф. Системи підтримки прийняття рішень. К. : КНЕУ, 2003.
114. Raymond McLeon Jr., George Schell. *Management Information Systems*. Hall, Upper Sadle River, New Jersey 07458, 2001.

115. Raymond McLeon Jr., George Schell. Management Information Systems. Hall, Upper Sadle River, New Jersey 07458, 2001.
116. Kimball R., Caserta J. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Confirming and Delivering Data. Wiley, 2004. 525 p.
117. Kimball R., Ross M. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. Wiley, 2002. 421 p.
118. Olson J. Data Quality Accuracy Dimension. Morgan Kauffmann Publishers, 2003. 293 p.
119. Woolf B. Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-learning. San Francisco, CA: Elsevier Inc., Morgan Kauffman, 2009. 440 p.
120. EDM 2014. URL: <http://www.educationaldatamining.org/EDM2014/> (дата обращения: 19.01.2015)
121. Апостолук В. О. Інтелектуальні системи керування : конспект лекцій. Київ, 2008. 88 с.
122. Бабенко Н.І., Крючковський Д.А., Маломуж Т.В. Методи зниження впливу факторів суб'єктивності та невизначеності у системах підтримки прийняття рішень. *Автоматика. Автоматизація. Електротехнічні комплекси та системи*. 2004. № 1. С. 114-122.
123. Polibina K. Model of Supporting an Individual Training Pathway for Software Engineers. *IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT)*. 2021. Lviv, Ukraine.
124. Aamodt A. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*. 1994. P.39–59.
125. Ульшин В.О. Моделі представлення знань експертних систем підтримки прийняття рішень при діагностуванні. *Вісник ЧНУ ім. В. Даля*. 2009. №5 (135). С. 21–25.

126. Глибовець А. М., Точицький В. В. Алгоритм токенизації та стемінгу для текстів українською мовою. Національний університет «Києво-Могилянська академія». URL: <https://ekmair.ukma.edu.ua/server/api/core/bitstreams/a756c821-adae-4dda-8642-712e278bff3e/content> (Дата звернення 25.09.2023)

127. Новокаховський приладобудівний фаховий коледж - веб-сайт. Правові документи. URL: <https://nkpfk.ks.ua/pravova-osnova> (Дата звернення 05.09.2023)

128. Полібіна К.В. Ефективність дистанційних освітніх технологій в умовах пандемії. *Інфокомунікаційні та комп'ютерні технології. Вісник Університету «Україна»*. Київ. 2021. № 2 (02), С.280-293.

ДОДАТОК А

Лістинг коду програми

Функція завантаження Excel-шаблонів

```
private void анкетаВикладачаToolStripMenuItem1_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.Filter = "Excel Files (*.xlsx)|*.xlsx";
    saveFileDialog.FileName = "TeacherProfile.xlsx";

    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        byte[] fileBytes = Properties.Resources.TemplateTeacherProfile;
        File.WriteAllBytes(saveFileDialog.FileName, fileBytes);
        MessageBox.Show("Файл збережено.");
    }
}

private void анкетаСтудентаToolStripMenuItem_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.Filter = "Excel Files (*.xlsx)|*.xlsx";
    saveFileDialog.FileName = "НазваФайлу.xlsx";

    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        byte[] fileBytes = Properties.Resources.TemplateStudentProfile;
        File.WriteAllBytes(saveFileDialog.FileName, fileBytes);
        MessageBox.Show("Файл збережено.");
    }
}

private void відомістьУспішностіToolStripMenuItem_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.Filter = "Excel Files (*.xlsx)|*.xlsx";
    saveFileDialog.FileName = "НазваФайлу.xlsx";
    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        byte[] fileBytes = Properties.Resources.TemplateЗведена_відомість_успішності;
        File.WriteAllBytes(saveFileDialog.FileName, fileBytes);
        MessageBox.Show("Файл збережено.");
    }
}
```

Функція завантаження Excel- файлів в додаток

```
private void button3_Click(object sender, EventArgs e)// файл Навантаження викладачів
{
    if (comboBox3.SelectedItem == null)
    {
        MessageBox.Show("Виберіть відділення для фільтрації викладачів.", "Помилка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    string selectedDepartment = comboBox3.SelectedItem.ToString();
    using (OpenFileDialog openFileDialog = new OpenFileDialog())
```

```

{
    openFileDialog.Filter = "Excel Files|*.xls;*.xlsx;*.xlsm";
    openFileDialog.Title = "Виберіть файл Excel";
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        string selectedFilePath = openFileDialog.FileName;
        if (IsFileAlreadyOpen(selectedFilePath))
        {
            MessageBox.Show("Закрийте файл перед використанням.");
            return;
        }
    }

    else if (openFileDialog.ShowDialog() == DialogResult.Cancel)
    {
        MessageBox.Show("Виберіть файл Excel");
        return;
    }

    using (var stream = File.Open(openFileDialog.FileName, FileMode.Open,
FileAccess.Read))
    {
        using (IExcelDataReader reader = ExcelReaderFactory.CreateReader(stream))
        {
            DataSet result = reader.AsDataSet();
            DataTable table = result.Tables[0];
            DataRow headers = table.Rows[0];
            int teachersColumnIndex = -1;
            int departmentColumnIndex = -1;
            for (int i = 0; i < headers.ItemArray.Length; i++)
            {
                if (headers.ItemArray[i].ToString().ToLower() == "викладачі")
                {
                    teachersColumnIndex = i;
                }
                else if (headers.ItemArray[i].ToString().ToLower() == "відділення")
                {
                    departmentColumnIndex = i;
                }
            }
            if (teachersColumnIndex != -1 && departmentColumnIndex != -1)
            {
                comboBox2.Items.Clear();

                for (int i = 1; i < table.Rows.Count; i++)
                {
                    if (table.Rows[i].ItemArray.Length > teachersColumnIndex &&
table.Rows[i].ItemArray.Length > departmentColumnIndex &&
table.Rows[i].ItemArray[departmentColumnIndex].ToString().ToLower() ==
selectedDepartment.ToLower())
                    {
                        string teacherName =
table.Rows[i].ItemArray[teachersColumnIndex].ToString();
                        if (!string.IsNullOrEmpty(teacherName) &&
!comboBox2.Items.Contains(teacherName))
                        {
                            comboBox2.Items.Add(teacherName);
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    }
}

private bool IsFileAlreadyOpen(string filePath)
{
    try
    {
        FileStream fileStream = File.Open(filePath, FileMode.Open, FileAccess.ReadWrite,
FileShare.None);
        fileStream.Close();
    }
    catch (IOException)
    {
        return true;
    }
    return false;
}

private void button5_Click(object sender, EventArgs e) Анкета викладача
{
    using (OpenFileDialog openFileDialog = new OpenFileDialog())
    {
        openFileDialog.Filter = "Excel Files|*.xls;*.xlsx;*.xlsm";
        openFileDialog.Title = "Select an Excel File";

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            selectedFilePath1 = openFileDialog.FileName;
            LoadExcelData(selectedFilePath1);
            {
                string selectedFilePath1 = openFileDialog.FileName;

                if (IsFileAlreadyOpen(selectedFilePath1))
                {
                    MessageBox.Show("Закрийте файл перед використанням.");
                    return;
                }
            }

            else if (openFileDialog.ShowDialog() == DialogResult.Cancel)
            {
                MessageBox.Show("Виберіть файл Excel");
                return;
            }
            LoadExcelData(selectedFilePath1);
            MessageBox.Show("Анкету викладачів завантажено!", "Інформація",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}

private void LoadExcelData(string filePath)
{
    using (var stream = File.Open(filePath, FileMode.Open, FileAccess.Read))

```

```

{
    using (var reader = ExcelReaderFactory.CreateReader(stream))
    {
        var result = reader.AsDataSet(new ExcelDataSetConfiguration()
        {
            ConfigureDataTable = (_) => new ExcelDataTableConfiguration() {
UseHeaderRow = true }
        });

        var table = result.Tables[0];
        string selectedTeacher = comboBox2.SelectedItem.ToString();
        comboBox4.Items.Clear();
        comboBox5.Items.Clear();
        textBox1.Clear();
        int teacherIndex = table.Columns.IndexOf("Викладач");
        int dpIndex1 = table.Columns.IndexOf("Тема ДП1");
        int dpIndex2 = table.Columns.IndexOf("Тема ДП2");
        int disciplineIndex1 = table.Columns.IndexOf("Дисципліни для вивчення1");
        int disciplineIndex2 = table.Columns.IndexOf("Дисципліни для вивчення2");
        int fieldIndex = table.Columns.IndexOf("Галузь інтересів викладача");
        foreach (DataRow row in table.Rows)
        {
            string teacherName = row[teacherIndex].ToString();
            if (teacherName.Equals(selectedTeacher,
StringComparison.OrdinalIgnoreCase))
            {
                string dpTopic1 = row[dpIndex1].ToString();
                if (!string.IsNullOrEmpty(dpTopic1) &&
!comboBox4.Items.Contains(dpTopic1))
                {
                    comboBox4.Items.Add(dpTopic1);
                }
                string dpTopic2 = row[dpIndex2].ToString();
                if (!string.IsNullOrEmpty(dpTopic2) &&
!comboBox4.Items.Contains(dpTopic2))
                {
                    comboBox4.Items.Add(dpTopic2);
                }
                string discipline1 = row[disciplineIndex1].ToString();
                if (!string.IsNullOrEmpty(discipline1) &&
!comboBox5.Items.Contains(discipline1))
                {
                    comboBox5.Items.Add(discipline1);
                }
                string discipline2 = row[disciplineIndex2].ToString();
                if (!string.IsNullOrEmpty(discipline2) &&
!comboBox5.Items.Contains(discipline2))
                {
                    comboBox5.Items.Add(discipline2);
                }

                textBox1.Text = row[fieldIndex].ToString();
            }
        }
    }
}

```

Функція обробки та додавання прецедента до бази прецедентів

```

private void button1_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) ||
        string.IsNullOrEmpty(comboBox2.Text) || string.IsNullOrEmpty(comboBox4.Text) ||
        string.IsNullOrEmpty(comboBox5.Text) || string.IsNullOrEmpty(comboBox6.Text))
    {
        MessageBox.Show("Перед збереженням заповніть усі поля!");
        return;
    }
    string directoryPath = Path.Combine(Application.StartupPath, "Temp");
    Directory.CreateDirectory(directoryPath);

    string filePath = Path.Combine(directoryPath, "TempDataBasePrecedent.xlsx");

    if (PrecedentExists(filePath))
    {
        MessageBox.Show("Такий прецедент вже існує!");
        return;
    }
    else
    {
        using (var package = new ExcelPackage(new FileInfo(filePath)))
        {
            ExcelWorksheet worksheet = package.Workbook.Worksheets.First();

            int nextRow = worksheet.Dimension?.Rows + 1 ?? 1;

            worksheet.Cells[nextRow, 1].Value = nextRow - 1;
            worksheet.Cells[nextRow, 2].Value = comboBox2.Text;
            worksheet.Cells[nextRow, 3].Value = comboBox4.Text;
            worksheet.Cells[nextRow, 4].Value = comboBox5.Text;
            worksheet.Cells[nextRow, 5].Value = textBox1.Text;
            worksheet.Cells[nextRow, 6].Value = textBox2.Text;
            worksheet.Cells[nextRow, 7].Value = comboBox6.Text;
            package.Save();
        }
        MessageBox.Show("Прецедент збережено в базі прецедентів.");
        ClearTextBoxes();
    }
}

// Функція для перевірки наявності дублікату у файлі
private bool PrecedentExists(string filePath)
{
    using (var package = new ExcelPackage(new FileInfo(filePath)))
    {
        ExcelWorksheet worksheet = package.Workbook.Worksheets.First();

        for (int row = 2; row <= worksheet.Dimension?.Rows; row++)
        {
            if (worksheet.Cells[row, 2].Value?.ToString() == comboBox2.Text &&
                worksheet.Cells[row, 3].Value?.ToString() == comboBox4.Text &&
                worksheet.Cells[row, 4].Value?.ToString() == comboBox5.Text &&
                worksheet.Cells[row, 5].Value?.ToString() == textBox1.Text &&
                worksheet.Cells[row, 6].Value?.ToString() == textBox2.Text &&
                worksheet.Cells[row, 7].Value?.ToString() == comboBox6.Text)
            {
                return true;
            }
        }
    }
}

```

```

        return false;
    }
}

```

Функція завантаження Excel- файлів в додаток

```

private void button3_Click(object sender, EventArgs e) Відомість успішності
{
    if (comboBox3.SelectedItem == null)
    {
        MessageBox.Show("Виберіть відділення для фільтрації викладачів.", "Помилка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    string selectedDepartment = comboBox3.SelectedItem.ToString();

    using (OpenFileDialog openFileDialog = new OpenFileDialog())
    {
        openFileDialog.Filter = "Excel Files|*.xls;*.xlsx;*.xlsm";
        openFileDialog.Title = "Виберіть файл Excel";

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            string selectedFilePath = openFileDialog.FileName;

            if (IsFileAlreadyOpen(selectedFilePath))
            {
                MessageBox.Show("Закрийте файл перед використанням.");
                return;
            }
        }

        else if (openFileDialog.ShowDialog() == DialogResult.Cancel)
        {
            MessageBox.Show("Виберіть файл Excel");
            return;
        }

        using (var stream = File.Open(openFileDialog.FileName, FileMode.Open,
        FileAccess.Read))
        {
            using (IExcelDataReader reader = ExcelReaderFactory.CreateReader(stream))
            {
                DataSet result = reader.AsDataSet();
                DataTable table = result.Tables[0];
                DataRow headers = table.Rows[0];

                int teachersColumnIndex = -1;
                int departmentColumnIndex = -1;

                for (int i = 0; i < headers.ItemArray.Length; i++)
                {
                    if (headers.ItemArray[i].ToString().ToLower() == "викладачі")
                    {
                        teachersColumnIndex = i;
                    }
                }
            }
        }
    }
}

```

```

    }
    else if (headers.ItemArray[i].ToString().ToLower() == "відділення")
    {
        departmentColumnIndex = i;
    }
}

if (teachersColumnIndex != -1 && departmentColumnIndex != -1)
{
    comboBox2.Items.Clear();

    for (int i = 1; i < table.Rows.Count; i++)
    {
        if (table.Rows[i].ItemArray.Length > teachersColumnIndex &&
table.Rows[i].ItemArray.Length > departmentColumnIndex &&
table.Rows[i].ItemArray[departmentColumnIndex].ToString().ToLower() ==
selectedDepartment.ToLower())
        {
            string teacherName =
table.Rows[i].ItemArray[teachersColumnIndex].ToString();
            if (!string.IsNullOrEmpty(teacherName) &&
!comboBox2.Items.Contains(teacherName))
            {
                comboBox2.Items.Add(teacherName);
            }
        }
    }
    MessageBox.Show("Список викладачів з відділенням \"" + selectedDepartment
+ "\" завантажено!", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
}

private bool IsFileAlreadyOpen(string filePath)
{
    try
    {
        FileStream fileStream = File.Open(filePath, FileMode.Open, FileAccess.ReadWrite,
FileShare.None);
        fileStream.Close();
    }
    catch (IOException)
    {
        return true;
    }

    return false;
}

```

Функція обробки та додавання опису поточної ситуації

```

private void button1_Click(object sender, EventArgs e)
{
    if (comboBox1.Text != "" && textBox1.Text != "" && textBox4.Text != "" &&
textBox2.Text != "" && textBox3.Text != "")
    {
        string student = comboBox1.Text;

```

```

        string x1 = textBox1.Text;
        string x2 = textBox4.Text;
        string x3 = textBox3.Text;
        string x4 = textBox2.Text;

        if (IsDuplicateEntry(student, x1, x2, x3, x4))
        {
            MessageBox.Show("Такий запис вже існує у таблиці.");
        }
        else
        {
            dataGridView1.Rows.Add(student, x1, x2, x3, x4);
            MessageBox.Show("Запис додано успішно.");
            comboBox1.Text = textBox1.Text = textBox4.Text = textBox2.Text = textBox3.Text
= "";
        }
    }
    else
    {
        MessageBox.Show("Заповніть всі поля перед збереженням.");
    }
}

private bool IsDataGridViewCleared()
{
    return dataGridView1.Rows.Count == 0;
}

```

Функція збереження в ресурсний файл

```

private void button9_Click(object sender, EventArgs e)
{
    string directoryPath = Path.Combine(Application.StartupPath, "Temp");
    string filePath = Path.Combine(directoryPath, "Parameters.xlsx");

    if (!File.Exists(filePath))
    {
        MessageBox.Show("Файл Excel не знайдено.");
        return;
    }

    SaveToExcel(filePath);
}

private void SaveToExcel(string filePath)
{
    FileInfo file = new FileInfo(filePath);

    using (ExcelPackage package = new ExcelPackage(file))
    {
        ExcelWorksheet worksheet = package.Workbook.Worksheets["Data"];

        if (worksheet == null)
        {
            MessageBox.Show("Лист 'Data' не знайдено у файлі Excel.");
            return;
        }

        int lastRow = worksheet.Dimension.End.Row;
    }
}

```

```

for (int rowIndex = dataGridView1.Rows.Count - 1; rowIndex >= 0; rowIndex--)
{
    var gridViewRow = dataGridView1.Rows[rowIndex];
    if (gridViewRow.IsNewRow)
    {
        continue;
    }

    bool isDuplicate = false;
    for (int excelRow = 1; excelRow <= lastRow; excelRow++)
    {
        bool rowMatch = true;
        for (int colIndex = 1; colIndex <= worksheet.Dimension.End.Column;
colIndex++)
        {
            string excelValue = worksheet.Cells[excelRow, colIndex].Text;
            string gridViewValue = gridViewRow.Cells[colIndex -
1].Value?.ToString() ?? string.Empty;

            if (!excelValue.Equals(gridViewValue,
StringComparison.InvariantCultureIgnoreCase))
            {
                rowMatch = false;
                break;
            }
        }

        if (rowMatch)
        {
            MessageBox.Show("Такі данні вже існують.");
            isDuplicate = true;
            dataGridView1.Rows.RemoveAt(rowIndex);
            break;
        }
    }

    if (!isDuplicate)
    {
        lastRow++;
        for (int colIndex = 0; colIndex < gridViewRow.Cells.Count; colIndex++)
        {
            worksheet.Cells[lastRow, colIndex + 1].Value =
gridViewRow.Cells[colIndex].Value;
        }
    }
}

package.Save();
MessageBox.Show("Дані додано до файлу Excel.");
ClearDataGridViewData();
}

private void ClearDataGridViewData()
{
    dataGridView1.DataSource = null;
    dataGridView1.Rows.Clear();
    dataGridView1.Refresh();
}

```

Функція завантаження БП в додаток

```

private void button1_Click(object sender, EventArgs e)
{
    string filePath = Path.Combine("Temp", "TempDataBasePrecedent.xlsx");
    LoadPrecedentsFromFile(filePath);
}
private string GetCellContentsLower(int rowIndex, string columnName)
{
    return dataGridView1.Rows[rowIndex].Cells[columnName].Value.ToString().ToLower();
}

private void LoadParametersFromFile(string filePath)
{
    try
    {
        using (var package = new ExcelPackage(new FileInfo(filePath)))
        {
            ExcelWorksheet worksheet = package.Workbook.Worksheets[0];

            System.Data.DataTable dataTable = new System.Data.DataTable();
            foreach (var firstRowCell in worksheet.Cells[1, 1, 1,
worksheet.Dimension.End.Column])
            {
                dataTable.Columns.Add(firstRowCell.Text);
            }

            for (var rowNumber = 2; rowNumber <= worksheet.Dimension.End.Row; rowNumber++)
            {
                var row = worksheet.Cells[rowNumber, 1, rowNumber,
worksheet.Dimension.End.Column];
                var newRow = dataTable.NewRow();
                foreach (var cell in row)
                {
                    newRow[cell.Start.Column - 1] = cell.Text;
                }
                dataTable.Rows.Add(newRow);
            }

            dataGridView2.DataSource = dataTable;

            dataGridView2.Columns["Student"].Width = 150;
            dataGridView2.Columns["x1"].Width = 200;
            dataGridView2.Columns["x2"].Width = 200;
            dataGridView2.Columns["x3"].Width = 200;
            dataGridView2.Columns["x4"].Width = 200;

            foreach (DataGridViewColumn column in dataGridView2.Columns)
            {
                column.ReadOnly = true;
                column.SortMode = DataGridViewColumnSortMode.NotSortable;
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("An error occurred while loading the file: " + ex.Message);
    }
}

```

Функція завантаження параметрів поточної ситуації в додаток

```
private void button2_Click(object sender, EventArgs e)
{
    string filePath = Path.Combine("Temp", "Parameters.xlsx");
    LoadParametersFromFile(filePath);
}

private void LoadParametersFromFile(string filePath)
{
    try
    {
        using (var package = new ExcelPackage(new FileInfo(filePath)))
        {
            ExcelWorksheet worksheet = package.Workbook.Worksheets[0];

            System.Data.DataTable dataTable = new System.Data.DataTable();
            foreach (var firstRowCell in worksheet.Cells[1, 1, 1,
worksheet.Dimension.End.Column])
            {
                dataTable.Columns.Add(firstRowCell.Text);
            }

            for (var rowNumber = 2; rowNumber <= worksheet.Dimension.End.Row; rowNumber++)
            {
                var row = worksheet.Cells[rowNumber, 1, rowNumber,
worksheet.Dimension.End.Column];
                var newRow = dataTable.NewRow();
                foreach (var cell in row)
                {
                    newRow[cell.Start.Column - 1] = cell.Text;
                }
                dataTable.Rows.Add(newRow);
            }
            dataGridView2.DataSource = dataTable;
            dataGridView2.Columns["Student"].Width = 150;
            dataGridView2.Columns["x1"].Width = 200;
            dataGridView2.Columns["x2"].Width = 200;
            dataGridView2.Columns["x3"].Width = 200;
            dataGridView2.Columns["x4"].Width = 200;

            foreach (DataGridViewColumn column in dataGridView2.Columns)
            {
                column.ReadOnly = true;
                column.SortMode = DataGridViewColumnSortMode.NotSortable;
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("An error occurred while loading the file: " + ex.Message);
    }
}
```

Функція розрахунку індексу Танімото при пакетному пошуку

```
private void button3_Click(object sender, EventArgs e)
{
    listView1.Columns[4].Width = 0;
    listView1.Columns[5].Width = 0;
```

```

        //CompareRowsFromDataGridViews();
        CompareAllRowsFromDataGridViews();
    }
private void CompareAllRowsFromDataGridViews()
{
    listView1.Items.Clear();
    foreach (DataGridViewRow rowDgv2 in dataGridView2.Rows)
    {
        double highestTanimotoCoefficient = 0;
        string highestTanimotoCaseNumber = string.Empty;
        string highestTanimotoRecommendation = string.Empty;

        HashSet<string> wordsSet2 = ExtractWordsSetFromRow(rowDgv2, "Student", "x4");

        foreach (DataGridViewRow rowDgv1 in dataGridView1.Rows)
        {
            HashSet<string> wordsSet1 = ExtractWordsSetFromRow(rowDgv1, "Case", "x1",
"x5", "x6");
            double currentTanimotoCoefficient = CalculateTanimotoCoefficient(wordsSet1,
wordsSet2);

            if (currentTanimotoCoefficient > highestTanimotoCoefficient)
            {
                highestTanimotoCoefficient = currentTanimotoCoefficient;
                highestTanimotoCaseNumber = rowDgv1.Cells["Case"].Value?.ToString();
                highestTanimotoRecommendation = FindRecommendation(dataGridView1,
highestTanimotoCaseNumber, rowDgv2);
            }
        }

        if (highestTanimotoCoefficient > 0)
        {
            string studentName = rowDgv2.Cells["Student"].Value?.ToString();
            string[] rowValues = { studentName, highestTanimotoCaseNumber,
$"{{highestTanimotoCoefficient * 100:F2}}%", highestTanimotoRecommendation };
            var listViewItem = new ListViewItem(rowValues);
            listView1.Items.Add(listViewItem);
        }
    }

    if (listView1.Items.Count == 0)
    {
        MessageBox.Show("Не знайдено жодної відповідності.");
    }
}
private double CalculateTanimotoCoefficient(HashSet<string> set1, HashSet<string> set2)
{
    var intersection = new HashSet<string>(set1);
    intersection.IntersectWith(set2);
    var union = new HashSet<string>(set1);
    union.UnionWith(set2);

    double tanimoto = intersection.Count / (double)(set1.Count + set2.Count -
intersection.Count);
    return tanimoto;
}

```

Функція виводу рекомендації

```

private string FindRecommendation(DataGridView dataGridView, string caseNumber, DataGridViewRow
selectedRow)
{
    var matchingRowDgv1 = dataGridView.Rows
        .Cast<DataGridViewRow>()
        .FirstOrDefault(row => row.Cells["Case"].Value?.ToString() == caseNumber);

    string x1DataDgv2 = selectedRow.Cells["x1"].Value?.ToString() ?? "";
    if (matchingRowDgv1 != null)
    {
        string x1DataDgv1 = matchingRowDgv1.Cells["x1"].Value?.ToString() ?? "";
        return "Науковий керівник: " + x1DataDgv1 + ". Тема ДП: " + x1DataDgv2;
    }

    return "";
}

```

Функція методу триграм

```

private void button5_Click(object sender, EventArgs e)
{
    listView1.Columns[1].Width = 50;
    listView1.Columns[3].Width = 200;
    listView1.Columns[4].Width = 250;
    listView1.Columns[5].Width = 200;
    CompareRowsUsingTriplets();
}
private void CompareRowsUsingTriplets()
{
    listView1.Items.Clear();

    if (dataGridView2.SelectedRows.Count == 0 || dataGridView1.Rows.Count == 0)
    {
        MessageBox.Show("Оберіть рядок параметрів студента!");
        return;
    }

    DataGridViewRow selectedRowFromDataGridview2 = dataGridView2.SelectedRows[0];

    HashSet<string> triplets2 = ExtractTripletsFromRow(selectedRowFromDataGridview2,
"Student", "x4");

    List<Tuple<string, double>> jaccardResults = new List<Tuple<string, double>>();

    foreach (DataGridViewRow row in dataGridView1.Rows)
    {
        HashSet<string> triplets1 = ExtractTripletsFromRow(row, "Case", "x1", "x5", "x6");
        double jaccardIndex = CalculateJaccardIndex1(triplets1, triplets2);

        string caseNumber = row.Cells["Case"].Value?.ToString();
        jaccardResults.Add(new Tuple<string, double>(caseNumber, jaccardIndex));
    }

    var topFiveResults = jaccardResults.OrderByDescending(t => t.Item2).Take(5);

    string studentName = selectedRowFromDataGridview2.Cells["Student"].Value?.ToString();
    foreach (var result in topFiveResults)
    {

```

```

        string recommendation = FindRecommendation1(dataGridView1, result.Item1,
selectedRowFromDataGridView2);

        string teacherLastName = GetTeacherLastName(dataGridView1, result.Item1);

        string note = recommendation;

        string x3DataDgv1 = dataGridView1.Rows
            .Cast<DataGridViewRow>()
            .FirstOrDefault(row => row.Cells["Case"].Value?.ToString() == result.Item1)
            ?.Cells["x3"].Value?.ToString();

        string[] rowValues = new string[] {
studentName,
result.Item1,
$"{{result.Item2 * 100:F2}}%",
teacherLastName,
x3DataDgv1,
note
};

        var listViewItem = new ListViewItem(rowValues);

        if (recommendation.Contains("Студенту потрібно підвищити загальний рівень знань"))
        {
            listViewItem.BackColor = Color.LightCoral;
        }
        if (recommendation.Contains("Викладач не викладає вказані освітні дисципліни"))
        {
            listViewItem.BackColor = Color.LightGreen;
        }

        listView1.Items.Add(listViewItem);
    }
}

private string GetTeacherLastName(DataGridView dataGridView, string caseValue)
{
    foreach (DataGridViewRow row in dataGridView.Rows)
    {
        if (row.Cells["Case"].Value?.ToString() == caseValue)
        {
            return row.Cells["x1"].Value?.ToString();
        }
    }
    return "Викладача не знайдено";
}

private HashSet<string> ExtractTripletsFromRow(DataGridViewRow row, params string[]
excludedColumns)
{
    HashSet<string> triplets = new HashSet<string>();
    StringBuilder letters = new StringBuilder();

    foreach (DataGridViewCell cell in row.Cells)
    {
        if (cell.OwningColumn == null || excludedColumns.Contains(cell.OwningColumn.Name))
            continue;

        string cellText = cell.Value?.ToString().ToLower().Trim() ?? string.Empty;
        var lettersOnly = new string(cellText.Where(char.IsLetter).ToArray());
    }
}

```

```

        letters.Append(lettersOnly);
    }

    for (int i = 0; i < letters.Length - 2; i++)
    {
        triplets.Add(letters.ToString().Substring(i, 3));
    }

    return triplets;
}

private double CalculateJaccardIndex1(HashSet<string> set1, HashSet<string> set2)
{
    var intersection = new HashSet<string>(set1);
    intersection.IntersectWith(set2);
    var union = new HashSet<string>(set1);
    union.UnionWith(set2);
    return union.Count > 0 ? intersection.Count / (double)union.Count : 0;
}

private string FindRecommendation1(DataGridView dataGridView, string caseNumber,
DataGridViewRow selectedRow)
{
    string recommendation = "";
    var matchingRowForFirstCondition = dataGridView.Rows
        .Cast<DataGridViewRow>()
        .FirstOrDefault(row => row.Cells["Case"].Value?.ToString() == caseNumber);

    if (matchingRowForFirstCondition != null)
    {
        string x6DataDgv1 =
matchingRowForFirstCondition.Cells["x6"]?.Value?.ToString().ToLower().Trim();
        string x4DataDgv2 = selectedRow.Cells["x4"]?.Value?.ToString().ToLower().Trim();

        if (x6DataDgv1?.Contains("не має досвіду роботи зі студентами з низьким рівнем
знань") == true &&
            x4DataDgv2?.Contains("незадовільний") == true)
        {
            recommendation = "Студенту потрібно підвищити загальний рівень знань";
        }
    }

    var matchingRowForSecondCondition = dataGridView.Rows
        .Cast<DataGridViewRow>()
        .FirstOrDefault(row => row.Cells["Case"].Value?.ToString() == caseNumber);

    if (matchingRowForSecondCondition != null)
    {
        string x5DataDgv1 =
matchingRowForSecondCondition.Cells["x5"]?.Value?.ToString().ToLower().Trim();
        string x3DataDgv1 = matchingRowForSecondCondition.Cells["x3"]?.Value?.ToString();

        if (x5DataDgv1?.Contains("не викладає потрібну освітню дисципліну") == true)
        {
            recommendation = "Викладач не викладає вказані освітні дисципліни, але має
експертизу по темі дипломного проектування";
        }
        return recommendation != "" ? recommendation : "Немає додаткових рекомендацій";
    }
}

```

```

        return recommendation != "" ? recommendation : "Немає додаткових рекомендацій";
    }

```

Функція збереження обраного рішення в ресурсний файл

```

private void button8_Click(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count == 0)
    {
        MessageBox.Show("Оберіть варіант");
        return;
    }
    ListViewItem selectedItem = listView1.SelectedItems[0];
    string student = selectedItem.SubItems[0].Text;
    string recommendation = selectedItem.SubItems[3].Text;
    string disciplines = selectedItem.SubItems[4].Text;
    string message = selectedItem.SubItems[5].Text;
    if (message.Contains("Студенту потрібно підвищити загальний рівень знань"))
    {
        MessageBox.Show("Рекомендація не може бути збережена");
        return;
    }
    string filePath = Path.Combine("Temp", "Recommendation.xlsx");
    if (!File.Exists(filePath))
    {
        using (ExcelPackage package = new ExcelPackage())
        {
            ExcelWorksheet worksheet = package.Workbook.Worksheets.Add("Recommendation");
            worksheet.Cells["A1"].Value = "Student";
            worksheet.Cells["B1"].Value = "x1";
            worksheet.Cells["C1"].Value = "x2";
            worksheet.Cells["A2"].Value = student;
            worksheet.Cells["B2"].Value = recommendation;
            worksheet.Cells["C2"].Value = disciplines;
            package.SaveAs(new FileInfo(filePath));
        }
        MessageBox.Show("Дані були збережені");
        return;
    }

    using (ExcelPackage package = new ExcelPackage(new FileInfo(filePath)))
    {
        ExcelWorksheet worksheet = package.Workbook.Worksheets["Recommendation"];
        int rowCount = worksheet.Dimension.Rows;
        for (int row = 2; row <= rowCount; row++)
        {
            if (worksheet.Cells[row, 1].Value?.ToString() == student &&
                worksheet.Cells[row, 2].Value?.ToString() == recommendation &&
                worksheet.Cells[row, 3].Value?.ToString() == disciplines)
            {
                MessageBox.Show("Такий запис вже існує у файлі");
                return;
            }
        }
        int newRow = rowCount + 1;
        worksheet.Cells[newRow, 1].Value = student;
        worksheet.Cells[newRow, 2].Value = recommendation;
        worksheet.Cells[newRow, 3].Value = disciplines;
        package.Save();
    }
}

```

```

    }

    MessageBox.Show("Дані були збережені");
}

```

Функція створення наказу

```

private void button6_Click(object sender, EventArgs e)
{
    string docPath = @"Resources\Наказ_ДП.doc";
    if (!File.Exists(docPath))
    {
        MessageBox.Show($"Файл {docPath} не знайдено");
        return;
    }

    var dataTable1 = ListViewToDataTable(listView1, "Студент", "Рекомендація системи");
    ExportToWord(dataTable1, docPath);
}

public System.Data.DataTable ListViewToDataTable(ListView listView, string column1Name,
string column2Name)
{
    System.Data.DataTable dataTable = new System.Data.DataTable();
    dataTable.Columns.Add(column1Name);
    dataTable.Columns.Add(column2Name);
    int columnIndex1 = -1, columnIndex2 = -1;
    for (int i = 0; i < listView.Columns.Count; i++)
    {
        if (listView.Columns[i].Text == column1Name)
            columnIndex1 = i;
        else if (listView.Columns[i].Text == column2Name)
            columnIndex2 = i;
    }
    if (columnIndex1 == -1 || columnIndex2 == -1)
    {
        MessageBox.Show($"Стовпчик ('{column1Name}', '{column2Name}') не знайдено");
        return dataTable;
    }
    foreach (ListViewItem item in listView.Items)
    {
        DataRow row = dataTable.NewRow();
        row[column1Name] = item.SubItems[columnIndex1].Text;
        row[column2Name] = item.SubItems[columnIndex2].Text;
        dataTable.Rows.Add(row);
    }

    return dataTable;
}

private void ExportToWord(System.Data.DataTable dataTable, string docPath)
{
    if (dataTable.Rows.Count > 0)
    {
        Microsoft.Office.Interop.Word.Application word = new
Microsoft.Office.Interop.Word.Application();
        Document document = word.Documents.Open(docPath);
        Range range = document.Content;
    }
}

```

```

        range.Find.ClearFormatting();
        if (range.Find.Execute("Рекомендовано:"))
        {
            range = range.Document.Range(range.End, range.End);
        }
        Table table = document.Tables.Add(range, dataTable.Rows.Count + 1,
dataTable.Columns.Count);
        table.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleSingle;
        table.Borders.InsideLineStyle = WdLineStyle.wdLineStyleSingle;
        float fontSize = 10;
        for (int i = 0; i < dataTable.Columns.Count; i++)
        {
            table.Cell(1, i + 1).Range.Text = dataTable.Columns[i].ColumnName;
            table.Cell(1, i + 1).Range.Font.Size = fontSize;
        }
        for (int i = 0; i < dataTable.Rows.Count; i++)
        {
            for (int j = 0; j < dataTable.Columns.Count; j++)
            {
                table.Cell(i + 2, j + 1).Range.Text = dataTable.Rows[i][j].ToString();
                table.Cell(i + 2, j + 1).Range.Font.Size = fontSize;
            }
        }

        word.Visible = true;
    }
    else
    {
        MessageBox.Show("Немає даних для експорту");
    }
}

```

Функція створення рекомендаційного листа

```

private void button7_Click(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count == 0)
    {
        MessageBox.Show("Оберіть варіант");
        return;
    }
    ListViewItem selectedItem = listView1.SelectedItems[0];
    string student = selectedItem.SubItems[0].Text;
    string recommendation = selectedItem.SubItems[3].Text;
    string disciplines = selectedItem.SubItems[4].Text;
    string message = selectedItem.SubItems[5].Text;
    if (message.Contains("Студенту потрібно підвищити загальний рівень знань"))
    {
        MessageBox.Show("Рекомендація не може бути створена");
        return;
    }
    Word.Application wordApp = new Word.Application();
    Document document = wordApp.Documents.Open(@"Resources\Лист_рекомендація.docx");

    InsertTextAfterWord("Шановний студент", student, document);
    InsertTextAfterWord("Ваш науковий керівник: ", recommendation, document);
    InsertTextAfterWord("Додаткове повідомлення системи:", message, document);
    InsertTextAfterWord(" наступних дисциплін: ", disciplines, document);
    wordApp.Visible = true;
}

```

```

private void InsertTextAfterWord(string word, string textToInsert, Word.Document document)
{
    Word.Range range = document.Content;
    range.Find.ClearFormatting();

    bool found = range.Find.Execute(FindText: word);
    while (found)
    {
        int wordEnd = range.Start + word.Length;
        Word.Range insertRange = document.Range(wordEnd, wordEnd);
        insertRange.Text = textToInsert;
        range.SetRange(insertRange.End, document.Content.End);
        found = range.Find.Execute(FindText: word);
    }
}

```

Функція скорегованого рішення за місцем практики

```

private void button3_Click(object sender, EventArgs e)
{
    if (selectedRowsDGV1.Count == 0 || selectedRowsDGV2.Count == 0)
    {
        MessageBox.Show("Оберіть варіант");
        return;
    }
    DataGridViewRow row1 = selectedRowsDGV1[selectedRowsDGV1.Count - 1];
    DataGridViewRow row2 = selectedRowsDGV2[selectedRowsDGV2.Count - 1];
    string student = row1.Cells["Student"].Value.ToString();
    string x1 = row1.Cells["x1"].Value.ToString();
    string x2 = row1.Cells["x2"].Value.ToString();
    string company = row2.Cells["Підприємство"].Value.ToString();
    string site = row2.Cells["Сайт"].Value.ToString();
    ListViewItem item = new ListViewItem(student);
    item.SubItems.Add(x1);
    item.SubItems.Add(x2);
    item.SubItems.Add(company);
    item.SubItems.Add(site);
    listView1.Items.Add(item);
    row1.ReadOnly = true;
    row2.ReadOnly = true;
    row1.DefaultCellStyle.BackColor = Color.LightGray;
    row2.DefaultCellStyle.BackColor = Color.LightGray;

    row1.Selected = false;
    row2.Selected = false;
}

private void dataGridView1_CellClick_1(object sender, DataGridViewCellEventArgs e)
{
    DataGridViewRow selectedRow = dataGridView1.Rows[e.RowIndex];
    selectedRowsDGV1.Add(selectedRow);
}

private void dataGridView2_CellClick_1(object sender, DataGridViewCellEventArgs e)
{
    DataGridViewRow selectedRow = dataGridView2.Rows[e.RowIndex];
    selectedRowsDGV2.Add(selectedRow);
}

```

```

private void dataGridView1_SelectionChanged(object sender, EventArgs e)
{
    if (dataGridView1.SelectedRows.Count > 0 &&
selectedRowsDGV1.Contains(dataGridView1.SelectedRows[0]))
    {
        dataGridView1.ClearSelection();
        MessageBox.Show("Цей рядок вже вибрано і додано.");
    }
}

private void dataGridView2_SelectionChanged(object sender, EventArgs e)
{
    if (dataGridView2.SelectedRows.Count > 0 &&
selectedRowsDGV2.Contains(dataGridView2.SelectedRows[0]))
    {
        dataGridView2.ClearSelection();
        MessageBox.Show("Рядок вже було додано. Зробіть інший вибір");
    }
}

```

Функція створення індивідуального плану

```

private void button4_Click(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count == 0)
    {
        MessageBox.Show("Оберіть варіант");
        return;
    }
    ListViewItem selectedItem = listView1.SelectedItems[0];
    string student = selectedItem.SubItems[0].Text;
    string teacher = selectedItem.SubItems[1].Text;
    string disciplines = selectedItem.SubItems[2].Text;
    string place = selectedItem.SubItems[3].Text;
    string site = selectedItem.SubItems[4].Text;
    Microsoft.Office.Interop.Word.Application wordApp = new
Microsoft.Office.Interop.Word.Application();
    Document document = wordApp.Documents.Open(@"Resources\TeacherCard1.doc");

    InsertTextAfterWord("ПІБ студента:", student, document);
    InsertTextAfterWord("Науковий керівник:", teacher, document);
    InsertTextAfterWord("Додаткові дисципліни для вивчення:", disciplines, document);
    InsertTextAfterWord("Місце проходження практики:", place, document);
    InsertTextAfterWord("Контактні данні:", site, document);
    wordApp.Visible = true;
}

private void InsertTextAfterWord(string word, string textToInsert,
Microsoft.Office.Interop.Word.Document document)
{
    Microsoft.Office.Interop.Word.Range range = document.Content;
    range.Find.ClearFormatting();

    bool found = range.Find.Execute(FindText: word);
    while (found)
    {
        int wordEnd = range.Start + word.Length;
        Microsoft.Office.Interop.Word.Range insertRange = document.Range(wordEnd,
wordEnd);

        insertRange.Text = textToInsert;
        range.SetRange(insertRange.End, document.Content.End);
        found = range.Find.Execute(FindText: word);
    }
}

```

```

    }
}
}
Функція завантаження прийнятих рішень
private void button1_Click(object sender, EventArgs e)
{
    string filePath = Path.Combine("Temp", "Recommendation.xlsx");
    LoadPrecedentsFromFile(filePath);
}
private void LoadPrecedentsFromFile(string filePath)
{
    try
    {
        using (var package = new ExcelPackage(new FileInfo(filePath)))
        {
            ExcelWorksheet worksheet = package.Workbook.Worksheets[0];
            System.Data.DataTable dataTable = new System.Data.DataTable();
            foreach (var firstRowCell in worksheet.Cells[1, 1, 1,
worksheet.Dimension.End.Column])
            {
                dataTable.Columns.Add(firstRowCell.Text);
            }
            for (var rowNumber = 2; rowNumber <= worksheet.Dimension.End.Row; rowNumber++)
            {
                var row = worksheet.Cells[rowNumber, 1, rowNumber,
worksheet.Dimension.End.Column];
                var newRow = dataTable.NewRow();
                foreach (var cell in row)
                {
                    newRow[cell.Start.Column - 1] = cell.Text;
                }
                dataTable.Rows.Add(newRow);
            }
            dataGridView1.DataSource = dataTable;
            dataGridView1.Columns["Student"].Width = 200;
            dataGridView1.Columns["x1"].Width = 150;
            dataGridView1.Columns["x2"].Width = 250;
            foreach (DataGridViewColumn column in dataGridView1.Columns)
            {
                column.ReadOnly = true;
                column.SortMode = DataGridViewColumnSortMode.NotSortable;
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("An error occurred while loading the file: " + ex.Message);
    }
}
}

```

Функція завантаження бази практик

```

private void button2_Click(object sender, EventArgs e)
{
    using (OpenFileDialog openFileDialog = new OpenFileDialog())
    {
        openFileDialog.Filter = "Excel Files|*.xls;*.xlsx;*.xlsm";
        openFileDialog.Title = "Select an Excel File";

        if (openFileDialog.ShowDialog() == DialogResult.OK)

```

```

{
    string selectedFilePath = openFileDialog.FileName;
    if (IsFileAlreadyOpen(selectedFilePath))
    {
        MessageBox.Show("Закрийте файл перед використанням.");
        return;
    }
    try
    {
        using (ExcelPackage package = new ExcelPackage(new FileInfo(selectedFilePath)))
        {
            ExcelWorksheet worksheet = package.Workbook.Worksheets[0];
            System.Data.DataTable dt = new System.Data.DataTable();
            dt.Columns.Add("Підприємство");
            dt.Columns.Add("Сайт");
            int rowCount = worksheet.Dimension.Rows;
            for (int row = 2; row <= rowCount; row++)
            {
                string company = worksheet.Cells[row, 1].Value?.ToString();
                string website = worksheet.Cells[row, 2].Value?.ToString();
                if (!string.IsNullOrEmpty(company) &&
!string.IsNullOrEmpty(website))
                {
                    dt.Rows.Add(company, website);
                }
            }
            dataGridView2.DataSource = dt;
            dataGridView2.Columns["Підприємство"].Width = 300;
            dataGridView2.Columns["Сайт"].Width = 300;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка при завантаженні даних з Excel: {ex.Message}");
    }
}

private bool IsFileAlreadyOpen(string filePath)
{
    try
    {
        FileStream fileStream = File.Open(filePath, FileMode.Open, FileAccess.ReadWrite,
FileShare.None);
        fileStream.Close();
    }
    catch (IOException)
    {
        return true;
    }

    return false;
}
List<DataGridViewRow> selectedRowsDGV1 = new List<DataGridViewRow>();
List<DataGridViewRow> selectedRowsDGV2 = new List<DataGridViewRow>();

```

ДОДАТОК Б

Вихідні документи



Індивідуальний план студента

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НОВОКАХОВСЬКИЙ ПРИЛАДОБУДІВНИЙ ФАХОВИЙ КОЛЕДЖ

Індивідуальний план студента групи П20-1,2Д

на II семестр 2023-2024 навчального року

Відділення	Інженерія програмного забезпечення
ПІБ Студента	Кім Дмитро Юрійович
Технологічна практика (8 тижнів)	з 05.02.2024 по 29.03.2024
Переддипломна практика (6 тижнів)	з 01.04.2024 по 10.05.2024
Місце проходження практики	ТОВ «КВАРТ-СОФТ»
Контактні данні	https://www.softline.kiev.ua
Дисципліни для вивчення	Сучасні технології розробки WEB застосувачів на платформі Microsoft.NET
Вимоги до знань, навичок та вмінь	<u>Обов'язково:</u> програмування – навички процедурного та об'єктно-орієнтованого програмування, наприклад, на мові C#; знання алгоритмів та структур даних. <u>Бажано:</u> об'єктно-орієнтоване проектування; базове знання C#; реляційні бази даних та SQL; базовий Web – основи HTML та CSS, протокол http
Що буде вивчатися на практиці	Устрій .Net коду та синтаксичні засоби його побудови: узагальнені типи даних, делегати, події, лямбда-вирази та інші. Принципи роботи прибиральника сміття та керування пам'яттю. Технології роботи з даними: ADO.Net, Entity Framework. Багатотопочне та асинхронне програмування та інші.
Чому це треба вчити	Аналіз пропозицій працевлаштування в IT-галузі на загальновідомих інтернет-агрегаторах DOU.ua та інших дає змогу зрозуміти, що розробники програмного забезпечення з використанням технологій Microsoft займають одні з перших позицій щодо попиту на їх знання та вміння, до того ж, якщо є хист та бажання вивчати сучасні мови та технології програмування, C# та .Net є досить молодим та стрімко змінюючимся напрямком створення програмного забезпечення, що надає широкі та цікаві перспективи як у навчанні, так і у подальшому працевлаштуванні

Чому можна навчитися	Будувати архітектуру програмних систем із компонентів зі слабкою зв'язаністю (багатошарову, мікросервісну, доменно-орієнтовану). Розробляти програмні модулі різного призначення – бізнес-логіка, компоненти доступу до даних, десктопні та web-інтерфейси користувача, компоненти багаторазового використання, розташовані як локально, так і на ресурсах глобального призначення (наприклад, nuget.org), сервіси, розгорнуті у www.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Дисципліна надає набір вмінь та компетенцій, що є мінімально необхідними для студента, який бажає в майбутньому стати C#.Net-розробником програмного забезпечення, а саме: створювати програмні продукти на мові C#, що працюють на платформах сімейства .Net, розробляти бази даних для СУБД MS SQL Server та застосовувати їх у програмних системах .Net, створювати і розгорнути web-сервіси та інші програмні компоненти для використання їх у системах з ервісноорієнтованою та мікросервісною архітектурою, писати комп'ютерні ігри на платформі unity, розробляти настільне та web- сучасне програмне забезпечення, створювати компоненти та інтегрувати їх в існуючі продукти Microsoft: Office, SharePoint
Дипломне проєктування (4 тижні)	Науковий керівник: Полібіна Катерина Вячеславівна Консультації: кожного вівторка з 14.00 по 18.00 Контактні данні: Телеграм @KatPol
Захист дипломних проєктів (3 тижня)	з 10.06.2024 по 28.06.2024

Завідувачка відділення ІПЗ

Світлана ГОВОРУН

Завідувачка циклової комісії ІІ

Катерина ПОЛІБІНА

Наказ на призначення наукових керівників та тем дипломного проєкта (частина)



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НОВОКАХОВСЬКИЙ ПРИЛАДОБУДІВНИЙ ФАХОВИЙ КОЛЕДЖ Н А К А З

« » 20__ № / / 23__
Про затвердження тем і керівників
дипломного проєктування

Відповідно до Положення про організацію освітнього процесу НКПФК, навчального плану на 202_-202_ н.р., освітньо-професійної програми підготовки молодших спеціалістів та фахових молодших бакалаврів; на підставі протоколу № __ від « » 20__ р. засідання циклової комісії дисциплін професійної підготовки за галуззю знань 12 Інформаційні технології зі спеціальності 121 Інженерія програмного забезпечення

Н А К А З У Ю:

1. Затвердити теми дипломних проєктів і призначити керівників дипломного проєктування за наступними студентами відділення.

Рекомендовано:

Студент	Рекомендація системи
Яценко Іван Олександрович	Науковий керівник: Градичар Раїса Василівна. Тема ДП: Розробка інформаційної системи СТО
Щербак Володимир Володимирович	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка бази даних для сайту з надання стоматологічних послуг
Соломонюк Давид Сергійович	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка інтернет-магазину з продажу кормів для _____

Коваль Олександр Віталійович	Науковий керівник: Градичар Раїса Василівна. Тема ДП: Створення галереї "Cat notebook"
Клиш Антон Володимирович	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка бази даних для телеграм-боту по доставці води Аюча
Кім Дмитро Юрійович	Науковий керівник: Полбіна Катерина Вячеславівна. Тема ДП: Розробка чат-боту у Telegram "Помічник: з гри Dota2"
Карчевський Артур Євгенович	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка сайту фітнес клубу
Каталь Ігор Олександрович	Науковий керівник: Катутін Дмитро Валерійович. Тема ДП: Розробка багатофункціонального додатку для прослуховування радіо
Гаврищенко Богдан Сергійович	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка сайту Інтернет магазину по продажу будівельних та садових інструментів "Серп і молот"
Барабаш Данил Сергійович	Науковий керівник: Полбіна Катерина Вячеславівна. Тема ДП: Розробка комп'ютерної гри на двигуні Сіксмап фізюп
Анісимов Давид Сергійович	Науковий керівник: Маюк Руслан Андрійович. Тема ДП: Розробка інтернет-магазину з продажу комп'ютерної техніки

2. Заступнику директора з навчальної роботи Бутовий Л.В. підготувати процедуру рецензування дипломних проєктів відповідно до законодавства.

3. Голові циклової комісії програмування та інформаційних технологій Полбіній К.В. разом з керівниками дипломного проєктування скласти графік роботи над дипломними проєктами з призначенням проміжних перевірок та графік засідань експертної комісії з захисту дипломних проєктів.

4. Завідувачу відділення Говорун С.М. підготувати документацію для роботи експертної комісії з захисту дипломних проєктів.

Лист рекомендація студенту

Шановний студент Анісимов Давид Сергійович

Рекомендаційна система «Choice» проаналізувала ваші дані і надає наступну рекомендацію*:

Ваш науковий керівник: Маюк Руслан Андрійович

Тема дипломного проєкта: Розробка Інтернет магазину з продажу комп'ютерної техніки

Додаткове повідомлення системи: немає додаткових повідомлень

Для плідної роботи над дипломним проєктом та отримання знань та навичок зі спеціальності «Інженерія програмного забезпечення» система рекомендує вам зосередитись на вивченні

Наступних дисциплін: Web-технології

Важливо пам'ятати, що глибоке знання цих дисциплін допоможе вам не тільки у навчальному процесі, але й стане міцним фундаментом для вашої майбутньої кар'єри. Приділіть увагу також додатковим матеріалам та джерелам, які можуть бути корисними в цьому процесі.

Ми віримо у ваш успіх та бажаємо продуктивного навчання!

З повагою,

зав. відділення Інженерії програмного забезпечення _____ Говорун С.М.

*Рекомендація системи може бути переглянута.

ДОДАТОК В

Акти впровадження та використання



АКТ

впровадження у навчальний процес результатів дисертаційної роботи аспірантки кафедри програмних засобів і технологій Херсонського національного технічного університету Полібіної Катерини Вячеславівни, поданої на здобуття наукового ступеня доктора філософії, на тему «Сценарно-прецедентні моделі підтримки індивідуальної освітньої траєкторії підготовки фахівців з інженерії програмного забезпечення»

Комісія у складі:

Голова комісії - заступник директора з НВР - **Бутова Людмила Володимирівна**,

Члени комісії - заступник директора з ВР **Загнйотова Валентина Іванівна**

- завідувач відділення «Інженерія програмного забезпечення»

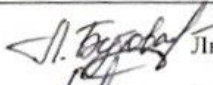
Говорун Світлана Михайлівна

цим Актом засвідчує, що результати дисертаційного дослідження Полібіної К.В. були впроваджені у навчальний процес Новокаховського приладобудівного фахового коледжу та отримали практичну реалізацію у вигляді створеного в дисертаційній роботі програмного забезпечення «Choice».

У навчальний процес впроваджено наступні результати дисертаційного дослідження:

- індивідуалізовано вибір керівника дипломного проєкту, який точно враховує професійні переваги та переваги академічної успішності здобувачів освіти, порівнюючи їх з компетентностями та експертизою викладання потенційних наукових керівників;
- розроблено механізм, який дозволяє студентам знаходити оптимальні варіанти розвитку своїх професійних навичок, враховуючи їхні індивідуальні потреби;
- здійснено оновлення інформації відповідно до сучасних вимог IT-індустрії, що допомагає здобувачам освіти зрозуміти їхні сильні і слабкі сторони, інтереси та вподобання; сприяє більш ефективному подальшому вибору професійної діяльності в галузі інформаційних технологій (ІТ), успішному формуванню власної кар'єри в майбутньому.

Зазначені результати дисертаційної роботи мають практичну цінність, що підтверджено результатами її апробації - покращення процесу підготовки фахівців з інженерії програмного забезпечення та підвищення якості захисту дипломних проєктів у 2022-2023 навчальному році.

Голова комісії -  Людмила БУТОВА

Члени комісії -  Валентина ЗАГНЬОТОВА

 Світлана ГОВОРУН



Міністерство освіти і науки України

Відокремлений структурний підрозділ

**«ХЕРСОНСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ОДЕСЬКА ПОЛІТЕХНІКА»**

вул. Небесної сотні, 23, м.Херсон, 73020, Україна, тел./факс.: +380675835663
www: <http://kspk.op.edu.ua>, e-mail: kspk@op.edu.ua, код ЄДРПОУ: 43868648

від 20 березня 2023 р. № 1-8/113в



АКТ

про використання в навчальному процесі результатів дисертаційної роботи
аспірантки кафедри програмних засобів і технологій Херсонського
національного технічного університету Полібіної Катерина Вячеславівни,
поданої на здобуття наукового ступеня доктор філософії, на тему «Сценарно-
прецедентні моделі підтримки індивідуальної освітньої траєкторії підготовки
фахівців з інженерії програмного забезпечення

Комісія у складі: голова - заступник директора з НР, кандидат економічних наук Ротань Наталя Вікторівна; члени комісії: завідувач відділення кандидат педагогічних наук – Подозьорова Анжела Володимирівна, голова циклової комісії, кандидат технічних наук Сафонов Михайло Сергійович підтвердила, що на відділенні «Комп'ютерної та програмної інженерії» впроваджені результати дисертаційного дослідження Полібіної К.В. в навчальних дисциплінах спеціальності 121 «Інженерія програмного забезпечення»: «Конструювання програмного забезпечення», «Бази даних», «Людино-машинний інтерфейс», «Основи науково-дослідницької роботи». Використання в навчальному процесі результатів дисертаційного дослідження Полібіної К.В. дає можливість ознайомити студентів з новими підходами та технологіями конструювання програмного забезпечення інтелектуальної підтримки прийняття рішень в галузі освіти, сценарно-прецедентними моделями зокрема.

Заступник директора з НР, к.ekon.n.
Завідувач відділення, к.пед.n.
Голова циклової комісії, к.т.n.

Наталя РОТАНЬ
Анжела ПОДОЗЬОРОВА
Михайло САФОНОВ

ЗАТВЕРДЖУЮ

Директор Відокремленого
структурного підрозділу
Новокаховський політехнічний
фаховий коледж Національного
університету «Одеська політехніка»

Віталій ВІТКОВ

12 березня 2023 р.

**АКТ**

про використання в навчальному процесі результатів дисертаційної роботи аспірантки кафедри програмних засобів і технологій Херсонського національного технічного університету Полібіної Катерини Вячеславівни, поданої на здобуття наукового ступеня доктор філософії, на тему «Сценарно-прецедентні моделі підтримки індивідуальної освітньої траєкторії підготовки фахівців з інженерії програмного забезпечення»

Комісія у складі: голова – директор НПФК Одеської політехніки, кандидат технічних наук – **Вітков Віталій Володимирович**; члени комісії: заступник директора з навчальної роботи – **Войтюк Лілія Миколаївна**, голова циклової комісії з інженерії програмного забезпечення – **Тодоріко Євгенія Сергіївна**, підтвердила, що на відділенні Економіки та програмування впроваджені результати дисертаційного дослідження Полібіної К.В. в навчальних дисциплінах спеціальності 121 Інженерія програмного забезпечення, освітньо-професійної програми Розробка програмного забезпечення з тем «Інформаційно-аналітичні системи», «Системи ІППР», «Бази даних», «Об'єктно-орієнтоване програмування».

Використання в навчальному процесі результатів дисертаційного дослідження Полібіної К.В. дає можливість ознайомити студентів з новими підходами та технологіями інтелектуальної підтримки прийняття рішень в галузі освіти, сценарно-прецедентними моделями зокрема.

Директор НПФК Одеської політехніки

Віталій ВІТКОВ

Заступник директора з НР

Лілія ВОЙТЮК

Голова ЦК

Євгенія ТОДОРІКО