

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

КОЛУМБЕТ ВАДИМ ПЕТРОВИЧ

УДК 004.896

ДИСЕРТАЦІЯ

**МЕТОД ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ РОЗРОБЦІ
ІНФОРМАЦІЙНИХ СИСТЕМ НА ОСНОВІ МУЛЬТИАГЕНТНОГО
ПІДХОДУ**

121 – Інженерія програмного забезпечення

12 – Інформаційні технології

Подається на здобуття наукового ступеня
доктора філософії.

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

_____ В.П. Колумбет

Науковий керівник:
БАРАБАШ Олег Володимирович,
доктор технічних наук, професор

КИЇВ – 2023

АНОТАЦІЯ

Колумбет В.П. Метод підтримки прийняття рішень при розробці інформаційних систем на основі мультиагентного підходу. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії з галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення. – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, 2023.

Дисертаційна робота присвячена вирішенню проблеми розробки методу підтримки прийняття рішень у сфері створення інформаційних систем на основі мультиагентного підходу. Об'єктом автоматизації виступає організаційно–технічна система, що є сукупністю організаційної структури і технічних засобів, тобто спільно розглядається людина та інформаційна система.

Успіх розробки інформаційної системи багато в чому визначається розробкою методологічного підходу, використовуваного в процесі проектування. Так, наявні методи і інструменти не забезпечують єдину модель інформаційної системи, як з точки зору розробника, так і користувача – предметного фахівця. Організаційно-технічна система характеризується процесами прийняття рішень, які передбачають роботу зі знаннями, що формалізовані сценаріями, а в деяких випадках пов'язані з координацією рішень, на жаль, існуючі методи не дозволяють комплексно вирішувати питання формалізації та інформатизації процесів прийняття рішень. Для аналізу, вдосконалення та перепроєктування бізнес–процесів в організаційно-технічній системі використовуються симуляційні та мультиагентні засоби моделювання. Проте, використання цих інструментів на етапах автоматизації та інформатизації все ще обмежене з двох причин: з однієї сторони, суттєві витрати на розробку імітаційної моделі, а з іншої – відсутність можливостей використовувати отримані результати і знання на етапах автоматизації. Ефект від інформатизації буде набагато вищим, якщо вирішувати завдання автоматизації разом із завданням удосконалення бізнес–процесів.

Метою дисертації є розробка методу і програмного забезпечення в області створення інформаційних систем для аналізу варіантів реалізації процесів в організаційно-технічних системах за допомогою імітаційного моделювання з метою скорочення часу на впровадження інформаційних систем шляхом автоматизації процесів переходу між етапами розробки, а також зменшення кінцевих витрат при розробці інформаційних систем.

Таким чином, актуальним є аналіз існуючих динамічних моделей процесів організаційно-технічної системи та моделей архітектури інформаційних систем, та на їх основі, створення методу підтримки прийняття рішень, що поєднує у собі ці підходи, а також програмного забезпечення для його реалізації – системи підтримки прийняття рішень.

Великий внесок та розвиток питань підтримки прийняття рішень та мультиагентних систем зробили роботи наступних дослідників: Александрова Д.В., Боема Б., Буча Г, Барабаша О.В., Вендрова А.М., Коршевніюка Л.О., Гейна К., Городецького В.І., Ковалю О.В., Калянова Г.М., Тоценко В.Г., Карсаєва О.В., Коваленко А.Є., Мінського М., Бідюка П.І., Ойхмана Є.Г., Попова Е.В., Тимощука О.Л., Рамбо Дж., Ройса У., Сарсона Т., Пилиповича А.Ю., Хаммера М., Дж. Чампі, Jennings, N.R., Wooldridge M J., Швецова О.М., та ін.

Об'єкт дослідження – процеси розробки інформаційних систем та автоматизації організаційно-технічних систем.

Предмет дослідження – методи підтримки прийняття рішень у сфері розробки інформаційних систем з використанням мультиагентного підходу.

Методи дослідження – для досягнення мети дисертації використовуються наступні методи: методи прийняття рішень(фреймово–семантичний підхід), системний аналіз, теорія проектування інформаційних систем, моделювання та експертне моделювання, методології структурного та об'єктно–орієнтованого аналізу та проектування, мультиагентний підхід

Наукова новизна одержаних результатів полягає в тому, що в дисертаційній роботі:

– вперше розроблено архітектуру програмного забезпечення для

автоматизації процесу проектування діаграм DFD, IDEF0, UML–схем прецедентів, послідовностей і класів, а також програмного інтерфейсу при проектуванні інформаційної системи, що базується на удосконаленому методі підтримки прийняття рішень та розробленому методі трансформації моделі організаційно-технічної системи в модель інформаційної системи, використання якої дозволяє спростити процес розроблення програмного забезпечення інформаційних систем та надає можливість створення прототипу форм інтерфейсу користувача

- вперше розроблено концептуальну модель предметної області організаційно-технічної системи, що заснована на фреймово-семантичному представленні знань та дозволяє більш повно відобразити предметну область організаційно-технічної системи, впорядкувати та вдвічі прискорити процес розробки програмного забезпечення інформаційної системи;

- удосконалено метод підтримки прийняття рішень для завдання розробки інформаційної системи на основі інтеграції структурних, агентних та об'єктно-орієнтованих підходів, який відрізняється від існуючих методів використання інформаційної складової опису предметної області на основі аналізу інформаційних потоків для побудови концептуальної моделі інформаційної системи, що дозволяє значно прискорити та спростити розробку програмних комплексів, що дозволяє прискорити розробку програмних комплексів у середньому в 1,8 разів;

- вперше розроблено метод трансформації моделі організаційно-технічної системи в модель інформаційної системи, що сформована як сукупність діаграм функціонального та об'єктно-орієнтованого підходу на основі використання концептуальних моделей предметної області. Даний метод дозволяє побудувати взаємно-однозначну відповідність між концептуальною моделлю предметної області організаційно-технічної системи та моделлю інформаційної системи для забезпечення ефективної взаємодії між фахівцями-предметниками та ІТ-фахівцями.

Практичне значення роботи полягає в тому, що результати дослідження лягли в основу реальних алгоритмів, методів і систем підтримки прийняття рішень

у сфері створення інформаційних систем, при цьому, зокрема:

- розроблено програмне забезпечення для перекладу моделі організаційно-технічної системи в модель інформаційної системи як сукупність діаграм функціонального та об'єктно–орієнтованого підходу;

- розроблено програмне забезпечення для моделювання користувацького інтерфейсу інформаційної системи.

- розроблено нову методику переходу від об'єктів концептуальної моделі предметної області організаційно-технічної системи до об'єктів концептуальної моделі предметної області інформаційної системи, що вирішує завдання встановлення відповідності концептів предметної області автоматизованих процесів організаційно-технічної системи концептам предметної області інформаційної системи, який ліг в основу розробки та створення проблемно–орієнтованої системи підтримки прийняття рішень, що дозволяє на відміну від існуючих методів формувати узгоджені між собою діаграми структурного та об'єктно–орієнтованого підходу, що описують архітектуру інформаційної системи.

Розроблений метод і система підтримки прийняття рішень в області створення інформаційної системи дозволяють аналізувати варіанти реалізації процесів в організаційно-технічних системах за допомогою імітаційного моделювання, скорочувати час на впровадження інформаційних систем шляхом автоматизації процесів переходу між етапами розробки, а також зменшувати кінцеві витрати при розробці інформаційних систем.

Дисертаційна робота виконана відповідно з поточними та перспективними планами наукової та науково–технічної діяльності Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» і є частиною досліджень в рамках науково–дослідних робіт:

- "Методи забезпечення функціональної стійкості розподілених інформаційних систем підприємств." Державний реєстраційний номер: № 0121U108334, КПП, м. Київ.

- "Управління ризиками сталого розвитку території з використанням методів штучного інтелекту" Державний реєстраційний номер: № 0120U105256, КПП,

м. Київ

– "Дослідження системи оцінки рівня інтернаціоналізації науково–дослідних установ" Державний реєстраційний номер: № 0121U110151, КПІ, м. Київ

У першому розділі розглянуто етапи системного аналізу розробки інформаційних систем. Проведено огляд та порівняльний аналіз найбільш поширених CASE–засобів, визначено вимоги до створення інтелектуальної системи прийняття рішень у області розробки інформаційних систем.

У другому розділі визначено вимоги до методу підтримки прийняття рішень у області розробки інформаційних систем для організаційно–технічних систем, проаналізовано існуючі моделі представлення динамічних моделей процесів, розроблено концептуальну модель предметної області організаційно–технічної системи, що заснована на фреймово–семантичному представленні знань.

У третьому розділі на основі методу підтримки прийняття рішення представлені принципи розробки та технічні рішення розробленої системи прийняття рішень.

У четвертому розділі представлено приклад використання розробленого програмного продукту реальному проєкті та проведено аналіз ефективності запропонованого методу підтримки прийняття рішень у розробленому модулі з сторонньою реалізацією задачі у пакетах аналогів, що показав скорочення часу переходу між етапами розробки інформаційних систем.

Мета досліджень щодо скорочення часу та зниження витрат на розробку і впровадження інформаційних систем досягнута та всі часткові завдання вирішені повністю. Наукові результати досліджень є внеском у розвиток теоретичних і прикладних основ розроблення та дослідження моделей і методів підтримки прийняття рішень при розробці інформаційних систем на основі мультиагентного підходу.

Ключові слова: підтримка прийняття рішень, організаційно–технічна система, імітаційне моделювання, автоматизація, концепти предметної області, інформаційна система, інформаційна технологія, стійкість системи, архітектура програмного забезпечення, семантичні та фреймові мережі, розподілені бази даних.

SUMMARY

Kolumbet V.P. Method of decision support the development of information systems based on a multi-agent approach.

Dissertation for the degree of Doctor of Philosophy in the field of knowledge 12 Information Technology, specialty 121 Software Engineering – National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, 2023.

The dissertation is devoted to solving the problem of developing a decision support method in the field of information systems development based on a multi-agent approach. The object of automation is an organizational and technical system, which is a combination of organizational structure and technical means, i.e., a person and an information system are considered together.

The success of an information system (IS) development is largely determined by the development of the methodological approach used in the design process. Thus, the available methods and tools do not provide a single model of an information system, from the point of view of the developer and the user – a subject matter expert. An organizational technical system (OTS) is characterized decision-making processes that involve working with knowledge that is formalized in scenarios, and in some cases are associated with the coordination of decisions, unfortunately, existing methods do not allow for a comprehensive solution to the issues of formalization and informatization of decision-making processes. Simulation and multi-agent modeling tools are used to analyze, improve, and redesign business processes (BP) in an organizational and technical system (OTS). However, the use of these tools at the stages of automation and informatization is still limited for two reasons: on the one hand, the significant costs of developing a simulation model, and in the other hand, the lack of opportunities to use the results and knowledge obtained at the stages of automation. The effect of informatization will be much higher if the automation task is solved together with the task improving business processes (BP).

The purpose of the dissertation is to develop methods and software in the field of information systems development for analyzing options for implementing processes in

organizational and technical systems using simulation modeling in order to reduce the time for implementing information systems by automating the transition processes between development stages, as well as reducing the final costs of developing information systems.

Thus, it is relevant to analyze the existing dynamic models of the processes of the organizational and technical system of the OTS and models of the architecture of information systems, and on their basis, to create a decision support method (DSM) that combines these approaches, as well as software for its implementation – a decision support system (DSS).

The following researchers have made a great contribution and development of decision support and multiagent systems: Aleksandrov D.V., Boem B., Bucha G., Barabash O.V., Vendrov A.M., Korshevnyuk L.O., Hein K., Gorodetsky V.I., Koval O.V., Kalyanov G.M., Totsenko V.G., Karsayev O.V., Kovalenko A.Ye, Minsky M., Bidyuk P.I., Eichmann E.G., Popov E.V., Tymoshchuk O.L., Rambo J., Royce W., Sarson T., Pylypovych A.Y., Hammer M., J. Ciampi, Jennings, N.R., Wooldridge M.J., Shvetsov O.M., and others.

Object of research – processes of information systems development and automation of organizational and technical systems.

Subject of research – methods of decision support in the field of information systems development using a multi-agent approach.

Research methods – to achieve this goal, the following methods are used: decision-making methods (frame-semantic approach), system analysis, information systems design theory, modeling and expert modeling, methodologies of structural and object-oriented analysis and design, multi-agent approach

The scientific novelty of the results obtained is that in the dissertation

– for the first time developed software architecture for automating the process of designing DFD, IDEF0 diagrams, UML diagrams of precedents, sequences and classes, as well as a software interface in the design of an information system, based on an improved method of decision support and a developed method of transforming a model of an organizational and technical system into a model of an information system, the use

of which simplifies the process of developing information system software and makes it possible to create a prototype of the software

- a conceptual model of the subject area the organizational and technical system based on the frame-semantic representation of knowledge was developed for the first time, which allows to more fully reflect the subject area of the organizational and technical system, to streamline and accelerate the process of developing information system software;

- a method of decision support for the task of developing an information system based on the integration of structural, agent-based and object-oriented approaches, which differs from existing methods by using the information component of the description of the subject area based on the analysis of information flows to build a conceptual model of the information system, which can significantly speed up and simplify the development of software systems;

- for the first time, a method for transforming model of an organizational and technical system into a model of an information system, formed as a set of diagrams of the functional and object-oriented approach based on the use of conceptual models of the subject area, has been developed. This method allows to build a mutually unambiguous correspondence between conceptual model of the subject area of the organizational and technical system and the model of the information system.

The practical significance of the work lies in the fact that the results of the study formed basis of real algorithms, methods and decision support systems in the field of information systems development, in particular

- tools for translating the OTS model into an information system model as a set of diagrams of the functional and object-oriented approach were developed;

- tools for modeling the user interface of an information system were developed.

- a new method of transition from the objects of the conceptual model of the subject area (CMSA) of the OTS to the objects of the conceptual model of the subject area (CMSA) of the IS was developed, which solves the task of establishing the correspondence of the concepts of the subject area of the automated processes of the OTS to the concepts of the subject area of the IS, which was the basis development and creation

of a problem-oriented decision support system (PoDSS), which allows, unlike existing methods, to form mutually agreed-upon diagrams of a structural and object-oriented approach describing the IS architecture.

The developed method and decision support system (DSS) in the field of IS development allow analyzing options for implementing processes in organizational and technical systems using simulation modeling, reducing the time for implementing information systems by automating the transition between development stages, and reducing the final costs of IS development.

The dissertation was performed in accordance with the current and future plans of scientific and scientific and technical activities of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" and is part of the research within the framework of scientific and research works:

- "Methods of ensuring the functional stability of distributed information systems of enterprises." State registration number: № 0121U108334, KPI, Kyiv.
- "Management of risks of sustainable development of the territory using methods of artificial intelligence" State registration number: № 0120U105256, KPI, Kyiv.
- "Research of the system for assessing the level of internationalization of research institutions" State registration number: № 0121U110151, KPI, Kyiv.

The first section discusses the stages of system analysis of information systems development. A review and comparative analysis of the most common CASE tools are carried out, the requirements for creating an intelligent decision-making system in the field of information systems development are determined.

The second section defines the following requirements for a decision support method in the field of information systems development for organizational and technical systems, analyzes existing models for representing dynamic process models, and develops conceptual model of the subject area of an organizational and technical system based on a frame-semantic representation of knowledge.

The third section, based on the above method, presents the principles of development and technical solutions of the developed EABP decision-making system, which is implemented on the basis of the Enterprise Architec package: functional and

object-oriented development of an information system, modeling of the software interface.

The fourth section presents an example of using the developed software product in a real project and analyzes the effectiveness of the proposed method of decision support in the developed module with a third-party implementation of the task in analogous packages, which showed a reduction in the transition time between the stages of information system development.

The goal of the research to reduce the time and costs for the development and implementation of information systems has been achieved and all partial tasks have been fully solved. The scientific results of the research are a contribution to the development of theoretical and applied foundations for the development and research of models and methods of decision support in the development of information systems based on a multi-agent approach.

Keywords: decision support, organizational and technical system, simulation modeling, automation, concepts of the subject area, information system, information technology, system stability, software architecture, semantic and frame networks, distributed databases.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Kolumbet V., Svynchuk O. Multiagent methods of management of distributed computing in hybrid clusters. *Сучасні інформаційні системи*. 2022. Том 6. № 1. С. 32 – 36.

<http://ais.khpi.edu.ua/article/view/254197/251480>

2. Колумбет В.П., Сторчак К.П. Метод трансформації моделі організаційно–технічної системи в модель інформаційної системи. *Науково–практичний журнал «Зв'язок»*. 2022. Вип. № 1 (155). С. 33 – 36.

<http://con.dut.edu.ua/index.php/communication/issue/view/159>

3. Барабаш О.В., Колумбет В.П. Дослідження систем масового обслуговування на основі імітаційного моделювання з урахуванням мультиагентного підходу. *Інфокомунікаційні та комп'ютерні технології*. 2022. Том 2. № 04. С. 115 – 121.

<https://visn-icct.uu.edu.ua/index.php/icct/issue/view/4>

4. Барабаш О.В., Колумбет В.П. Інтелектуальний аналіз та обробка текстових документів на основі мультиагентного підходу. *Інформатика та математичні методи в моделюванні*. 2022. Том 12. № 4. С. 296 – 305.

http://immm.op.edu.ua/files/archive/n4_v12_2022/immm_n4_v12_2022.pdf

Наукові праці, які засвідчують апробацію матеріалів дисертації:

5. Барабаш О.В., Колумбет В.П. Оптимізація обробки запитів кластерами в реальному часі із застосуванням мультиагентного підходу. Матеріали XIX міжнародної науково–практичної конференції молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики», м. Київ, 20 – 23 квітня 2021 р. Київ, КПІ ім. Ігоря Сікорського, 2021. Том 2. С. 121 – 122.

<https://tef.kpi.ua/files/pdf/2021-tom2-tezy.pdf>

6. Колумбет В.П., Барабаш О.В. Використання методу потенціалів для керування МРТС в середовищі з перешкодами. XIV міжнародна науково–

практична конференція «Інтегровані інтелектуальні робототехнічні комплекси (ІІРТК–2021)»: збірка тез, м. Київ, 18 – 19 травня 2021 р. Київ, НАУ, 2021. С. 34 – 36.

<https://drive.google.com/drive/folders/1mVxrvWRYfsG2O8jgbTwtXHhkGn9UobCq>

7. Колумбет В.П., Барабаш О.В. Мультиагентний підхід до оцінки стану безпеки інформаційних систем. XVII Міжнародна науково–практична конференція «Військова освіта і наука: сьогодення та майбутнє»: збірник тез доповідей, м. Київ, 26 листопада 2021 р. Том 1. Київ, ВІКНУ, 2021. С. 47 – 48.

<https://mil.univ.kiev.ua/page/lib/32>

8. Колумбет В.П. Методичні та теоретичні засади підтримки прийняття рішень, моделювання та розвитку інформаційних систем з використанням мультиагентного підходу. Міжнародна науково–практична інтернет–конференція «Розвиток освіти, науки та бізнесу: результати 2020»: тези доповідей, м. Дніпро, 3 – 4 грудня 2020 р. Том 1. Дніпро, 2020. С. 513 – 515.

<http://www.wayscience.com/wp-content/uploads/2020/12/Part-1-Conference-Results-2020..pdf>

9. Колумбет В. П. Мультиагентний підхід до управління ресурсами в розподілених обчислювальних середовищах. VII Міжнародна науково–практична конференція “*Priority Directions of Science and Technology Development*”, м. Київ, 21 – 23 березня 2021 р. Київ, Україна. С. 340 – 344.

<https://sci-conf.com.ua/wp-content/uploads/2021/03/PRIORITY-DIRECTIONS-OF-SCIENCE-AND-TECHNOLOGY-DEVELOPMENT-21-23.03.21.pdf>

10. Колумбет В.П., Барабаш О.В. Мультиагентний підхід для управління розподіленими обчисленнями на рівні GRID–системи. International scientific conference “*Innovation around us ‘2021*” March 9–10, 2021, Published by: SWorld in conjunction with D.A. Tsenov Academy of Economics, Svishtov, Bulgaria. С. 5 – 8. DOI: 10.30889/2709–183X.2021–07–00

<https://www.sworld.com.ua/konferbg7/cp-bg-7.pdf>

11. Barabash O.V., Svynchuk O.V., Kolumbet V.P. Simulation Modeling of

Distributed Random Processes Using a Multiagent Approach. Conference proceedings. International scientific conference “*The current stage of development of scientific and technological progress'2021*”. February 8 – 9, 2021. Karlsruhe, Germany. P. 12 – 16.

<https://www.sworld.education/konferge15/sbor-ge15.pdf>

12. Колумбет В.П., Барабаш О.В. Розв’язання навігаційної задачі для агента в неоднорідному середовищі. Наукові праці Четвертої міжнародної науково–практичної конференції «*Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій*» 1–2 лютого 2022 р., м. Київ, НУХТ, 2022. С. 87 – 88.

http://kist.ntu.edu.ua/konferencii/31_konf_2022.pdf

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	18
ВСТУП.....	19
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ НАУКОВИХ МЕТОДІВ ЩО ДО РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ НА ОСНОВІ МУЛЬТИАГЕНТНОГО ПІДХОДУ	28
1.1. Дослідження системного аналізу в розробці інформаційних систем	29
1.1.1. Організаційно–технічні системи	29
1.1.2. Моделювання автоматизованих процесів.....	31
1.1.3. Реінжиніринг бізнес–процесів	35
1.2. Дослідження ризиків пов'язаних з розробкою інформаційних систем і шляхи їх зниження	36
1.3. Аналіз методологічних та теоретичних основ підтримки прийняття рішень, моделювання та розвитку інформаційних систем.....	37
1.3.1. Застосування імітаційного моделювання в розробці програмного забезпечення.....	38
1.3.2. Використання систем штучного інтелекту при розробці інформаційних систем	39
1.3.3. Застосування мультиагентного підходу	40
1.4.1. Класифікація CASE–інструментів.....	42
1.4.2. Опис CASE–інструментів.....	43
1.4.3. Критерії порівняння функціоналу CASE–інструментів.....	49
1.4.4. Порівняльний аналіз CASE–інструментів	50
1.5. Постановка наукового завдання та часткових завдань досліджень	52
Висновки до розділу 1.	55

РОЗДІЛ 2 РОЗРОБКА НАУКОВО–МЕТОДИЧНОГО АПАРАТУ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ РОЗРОБЦІ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ МУЛЬТИАГЕНТНОГО ПІДХОДУ	56
2.1. Дослідження вимог до моделі та методу підтримки прийняття рішень при розробці інформаційних систем	56
2.2. Вибір моделі подання бізнес–процесів.....	57
2.3. Вибір моделі подання знань	59
2.4. Розробка концептуальної моделі предметної області організаційно-технічної системи на основі фреймово семантичної моделі представлення знань	63
2.4.1. Удосконалення методу підтримки прийняття рішень для завдання розробки інформаційної системи.....	63
2.4.2. Розробка методу трансформації моделі організаційно–технічної системи в модель інформаційної системи	71
2.5. Дослідження методу розробки інформаційних систем	77
2.6. Аналіз затримок в синхронній інформаційній системі	104
2.7. Вибір та застосування методики оцінки ефективності до методу розробки інформаційних систем	108
Висновок до розділу 2.....	110
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ ЕАВР ДЛЯ ВІЗУАЛЬНОГО МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ.....	112
3.1. Функціональність модуля ЕАВР	112
3.2. Опис CASE–засобу ЕАВР	113
3.2.1 Загальна структура інструменту CASE ЕАВР	113
3.2.2 Створення діаграми DFD.....	115
3.2.3. Створення діаграми прецедентів	120

3.2.4. Створення схеми послідовності.....	122
3.2.5. Створення діаграми класів	127
3.2.6. Підсистема моделювання інтерфейсу користувача.....	131
3.3. Опис агента інтеграції ЕАВР	136
3.4. Методика використання модулю ЕАВР	137
Висновки до розділу 3	139
РОЗДІЛ 4 ЗАСТОСУВАННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЕАВР ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ.....	141
4.1. Дослідження можливості застосування модулю ЕАВР для розробки проекту з аналізу бізнес–процесів та розробки технічного завдання на єдину інформаційну систему	141
4.1.1. Процес «Хід сесії».....	142
4.1.2. Процес «Рух контингенту».....	144
4.1.3. Оцінка ефективності впровадження модуля «Рух контингенту»	149
4.2. Розробка додаткових модулів для системи ведення реєстрів документів «Planaria»	152
4.2.1. Реєстрація вхідних документів	152
4.2.2. Модуль "Трансфер–агентський обмін"	156
4.2.3. Система «Електронний документообіг»	159
4.3. Експериментальні оцінки.....	162
Висновки до розділу 4	168
ВИСНОВКИ.....	170
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	174
ДОДАТОК А	187
ДОДАТОК Б	189

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CASE (Computer Aided Software Engineering) – автоматизована розробка ПЗ

UML (Unified Modeling Language) – уніфікована мова моделювання

БД – база даних

БП – бізнес–процес

ЄІС – єдина інформаційна система

ЖЦ ПЗ – життєвий цикл ПЗ

ІА – інтелектуальні агенти

ІІІ – штучний інтелект

ІМ – імітаційне моделювання

ІС – інформаційна система

ІТ – інформаційні технології

ОПР – особа, яка приймає рішення

МАС – мультиагентні системи

МППР – мультиагентні процеси перетворення ресурсів

МЛВ – машина логічного виводу

ООМ – об'єктно–орієнтовані методи

ОТС – організаційно–технічні системи

ПЗ – програмне забезпечення

ППР – підтримка прийняття рішення

СВР – система ведення реєстрів

СМО – системи масового обслуговування

СППР – система підтримки прийняття рішень

СУБД – система управління базами даних

ТА – трансфер–агент

ТЗ – технічне завдання

КМПО – концептуальна модель предметної області

КІ – користувальницький інтерфейс

ПІ – програмний інтерфейс

ВСТУП

Актуальність теми. Дисертаційна робота присвячена вирішенню задачі розробки методу підтримки прийняття рішень в області створення інформаційних систем на основі мультиагентного підходу. Об'єктом автоматизації є організаційно–технічна система, яка є набором організаційної структури і технічних засобів, що знаходяться в її розпорядженні, тобто спільно розглядається людина та інформаційна система. Розробка ІС це задача із високим ступенем ризику.

Згідно з дослідженнями, лише 21 % проектів тривалістю більше двох років виконуються вчасно. Однією з причин такого явища є спотворення і втрата інформації про розроблену інформацію і особливості процесів в ланцюжку її передачі між користувачем, аналітиком, розробником. Технічне завдання – має бути сполучною ланкою, але воно не повною мірою відображає як предметну область ОТС (в плані процесів координації і прийняття рішень), так і не вирішує питання зв'язування очікувань користувача з вимогами до ІС. Так, наприклад, з усього ТЗ користувач розуміє лише розділ «Функції системи» і сприймає візуальний, користувацький інтерфейс як всю інформаційну систему.

Успіх розробки ІС багато в чому визначається розробкою методологічного підходу, використовуваного в процесі проектування. Слід зазначити наступні моменти. По–перше, існуючі методи і інструменти не забезпечують єдину модель інформаційної системи, як з точки зору розробника, так і користувача – предметного фахівця. По–друге, ОТС характеризується процесами прийняття рішень, які передбачають роботу зі знаннями, що формалізовані сценаріями, а в деяких випадках пов'язані з координацією рішень. Існуючі методи не дозволяють комплексно вирішувати питання формалізації та інформатизації процесів прийняття рішень. По–третє, для аналізу, вдосконалення та перепроєктування бізнес–процесів в ОТС використовуються симуляційні та мультиагентні засоби моделювання.

Однак використання цих інструментів на етапах автоматизації та

інформатизації все ще обмежене з двох причин: з однієї сторони, суттєві витрати на розробку імітаційної моделі, а з іншої – відсутність можливостей використовувати отримані результати і знання на етапах автоматизації.

В даний час існують різні підходи для вирішення задачі розробки методу підтримки прийняття рішень. Структурний підхід (IDEF0, DFD) дозволяє описати систему, що розробляється у вигляді ієрархії взаємопов'язаних функцій. Ця ідея зрозуміла аналітику і користувачеві. Для аналізу вузьких місць і динамічних характеристик використовується імітаційне моделювання. Експертні системи закривають питання, пов'язані з описом знань і сценаріїв прийняття рішень. Мультиагентні системи (МАС) дозволяють автоматизувати процеси прийняття рішень і взаємодію осіб, які приймають рішення (ОПР). Програмні агенти виконують функції ОПР. Кожен з них окремо не закриває всі питання, які виникають при автоматизації ОТС–процесів.

Таким чином, актуальним є аналіз існуючих динамічних моделей процесів ОТС та моделей архітектури інформаційних систем, та на їх основі, створення методу ППР, що поєднує у собі ці підходи, а також програмного забезпечення для його реалізації – системи підтримки прийняття рішень (СППР).

Великий внесок та розвиток питань підтримки прийняття рішень та мультиагентних систем зробили роботи наступних дослідників: Александрова Д.В., Боема Б., Буча Г, Вендрова А.М., Коршевніюка Л.О., Гейна К., Городецького В.І., Калянова Г.М., Тоценко В.Г., Карсаєва О.В., Коваленко А.Є., Мінського М., Бідюка П.І., Ойхмана Є.Г., Попова Е.В., Тимошука О.Л., Рамбо Дж., Ройса У., Сарсона Т., Пилиповича А.Ю., Хаммера М., Дж. Чампі, Jennings, N.R., Wooldridge M J., Швецова О.М., та ін.

Існуючі методи і інструменти розглянуті в роботах зазначених дослідників не забезпечують єдину модель інформаційної системи та не дозволяють комплексно вирішувати питання формалізації та інформатизації процесів прийняття рішень. Також недостатньо розроблені питання використання методів та моделей на етапах автоматизації та інформатизації, що обмежені суттєвими витратами на розробку імітаційної моделі і відсутністю можливостей

використовувати отримані результати і знання на етапах автоматизації.

Недосконало досліджені питання прогнозування достовірності отриманої інформації від мультиагентних систем з подальшим використанням цієї інформації при формуванні рішень. Всі ці фактори підтверджують актуальність розробки методів ППР при розробці інформаційних систем на основі мультиагентного підходу.

На підставі проведеного аналізу виявлено існування протиріч між вимогами до розроблюваного методу ППР: між вимогою до підвищення ефективності інформаційної системи з використанням мультиагентного підходу, що потребує додаткових витрат як часу так і коштів, та вимогою на зменшення витрат на розробку ефективного математичного та програмного забезпечення інформаційної системи.

Дана суперечлива ситуація лежить в основі актуального наукового завдання щодо розробки методів і засобів в області створення інформаційних систем для аналізу варіантів реалізації процесів в організаційно-технічних системах за допомогою імітаційного моделювання, скорочення часу на впровадження інформаційних систем шляхом автоматизації процесів переходу між етапами розробки, а також зменшення кінцевих витрат при розробці інформаційних систем.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконана відповідно з поточними та перспективними планами наукової та науково-технічної діяльності Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» і є частиною досліджень в рамках науково-дослідних робіт, які виконувались на кафедрі інженерії програмного забезпечення в енергетиці:

– "Методи забезпечення функціональної стійкості розподілених інформаційних систем підприємств." Державний реєстраційний номер: № 0121U108334, КПП, м. Київ.

– "Управління ризиками сталого розвитку території з використанням методів штучного інтелекту" Державний реєстраційний номер: № 0120U105256, КПП ім. Ігоря Сікорського, м. Київ

– "Дослідження системи оцінки рівня інтернаціоналізації науково–дослідних установ" Державний реєстраційний номер: № 0121U110151, КПІ ім. Ігоря Сікорського, м. Київ

Особисто автором в НДР "Управління ризиками сталого розвитку території з використанням методів штучного інтелекту" запропоновано модель мультиагентного підходу при розробці системи моніторингу індикаторів сталого людського розвитку регіонів України; в НДР "Методи забезпечення функціональної стійкості розподілених інформаційних систем підприємств." запропоновано метод підтримки прийняття рішень на основі інтеграції структурних, агентурних та об'єктно–орієнтованих підходів, імітаційного моделювання.

Мета і завдання дослідження.

Метою дисертації є скорочення часу та зниження витрат на розробку і впровадження інформаційних систем за рахунок реалізації розробленого методу підтримки прийняття рішень для завдання автоматизації процесів організаційно–технічної системи при розробці інформаційних систем.

Для реалізації мети дослідження в роботі необхідно вирішити такі завдання:

1. Проаналізувати існуючі моделі формалізації процесів організаційно–технічних систем з урахуванням динаміки процесів і людського фактору, з метою вибору адекватної моделі її подальшого використання при розробці методу.

2. Проаналізувати існуючі методи і засоби створення інформаційної системи з метою оцінки рівня розвитку цього напрямку.

3. Визначити особливості, які є специфічними для організаційно–технічної системи, та розробити концептуальну модель: визначити і описати основні об'єкти і взаємозв'язки між ними, з метою формалізації предметної області.

4. Удосконалити існуючий метод підтримки прийняття рішень для завдання розробки інформаційної системи з використанням інтеграції структурних, агентних, імітаційних та об'єктно–орієнтованих підходів.

5. Розробити метод трансформації моделі організаційно–технічної системи в модель інформаційної системи, що сформована як сукупність діаграм

функціонального та об'єктно–орієнтованого підходу на основі використання концептуальних моделей предметної області.

6. Провести імітаційне моделювання для аналізу роботи окремих елементів інформаційної системи, оцінити ефективність роботи запропонованого методу проектування програмного забезпечення та провести оцінку продуктивності інформаційної системи.

7. Розробити програмні модулі та провести дослідження нового програмного середовища для аналізу процесів та розробки інформаційної системи, а також провести випробування розроблених положень для підтвердження їх достовірності.

Об'єкт дослідження – процеси розробки інформаційних систем та автоматизації організаційно-технічних систем.

Предмет дослідження – методи підтримки прийняття рішень у сфері розробки інформаційних систем з використанням мультиагентного підходу.

Методи дослідження. Для досягнення цієї мети використовуються наступні методи: методи прийняття рішень(фреймово–семантичний підхід, запропонованим Швецовим А.Н.), системний аналіз(для вдосконалення концептуальних моделей предметної області), теорія проектування інформаційних систем,(для розробки комп'ютерної технології підтримки прийняття рішень у сфері інформаційних систем) моделювання та експертне моделювання, методології структурного та об'єктно–орієнтованого аналізу та проектування, мультиагентний підхід.

Наукова новизна дослідження полягає в такому:

1. Вперше розроблено архітектуру програмного забезпечення для автоматизації процесу проектування діаграм DFD, IDEF0, UML–схем прецедентів, послідовностей і класів, а також програмного інтерфейсу при проектуванні інформаційної системи, що базується на удосконаленому методі підтримки прийняття рішень та розробленому методі трансформації моделі організаційно–технічної системи в модель інформаційної системи, використання якої дозволяє спростити процес розроблення програмного забезпечення інформаційних систем та надає можливість створення прототипу форм інтерфейсу користувача.

2. Вперше розроблено концептуальну модель предметної області організаційно–технічної системи, що заснована на фреймово–семантичному представленні знань, що дозволяє відобразити предметну область організаційно–технічної системи, впорядкувати та вдвічі прискорити процес розробки програмного забезпечення інформаційної системи.

3. Удосконалено метод підтримки прийняття рішень щодо розробки інформаційної системи на основі інтеграції структурних, агентних та об'єктно–орієнтованих підходів, який відрізняється від існуючих методів використання інформаційної складової опису предметної області на основі аналізу інформаційних потоків для побудови концептуальної моделі інформаційної системи, що дозволяє прискорити розробку програмних комплексів у середньому в 1,8 разів.

4. Вперше розроблено метод трансформації моделі організаційно–технічної системи в модель інформаційної системи, що базується на представленні архітектури інформаційної системи у вигляді структурних діаграм та діаграм об'єктно–орієнтованого підходу на основі використання концептуальних моделей предметної області. Даний метод дозволяє побудувати взаємно–однозначну відповідність між концептуальною моделлю предметної області організаційно–технічної системи та моделлю інформаційної системи для забезпечення ефективної взаємодії між фахівцями–предметниками та ІТ–фахівцями.

Наукове значення дослідження полягає в розвитку теорії і методів побудови систем підтримки прийняття рішень, комп'ютерного моделювання мультиагентних процесів перетворення ресурсів та інформаційних систем.

Практичне значення одержаних результатів полягає в тому, що результати дослідження лягли в основу реальних алгоритмів, методів і систем підтримки прийняття рішень у сфері створення інформаційних систем, а саме:

- розроблено програмне забезпечення для перекладу моделі ОТС в модель інформаційної системи як сукупність діаграм функціонального та об'єктно–орієнтованого підходу;
- розроблено програмне забезпечення для моделювання користувацького

інтерфейсу інформаційної системи.

– розроблено нову методику переходу від об'єктів КМПО ОТС до об'єктів КМПО ІС, що вирішує завдання встановлення відповідності концептів предметної області автоматизованих процесів ОТС концептам предметної області ІС, який ліг в основу розробки та створення проблемно–орієнтованої СППР, що дозволяє на відміну від існуючих методів формувати узгоджені між собою діаграми структурного та об'єктно–орієнтованого підходу, що описують архітектуру ІС

Результати досліджень прийняті до впровадження в ДП «Одеський науково–дослідний інститут зв'язку» (два акти від 31.08.2022 р.). З його допомогою було підготовлено технічне завдання на Електронну інформаційну систему закладу вищої освіти, розроблено модулі систем ведення реєстру «Planaria» та «Planaria 2». Економічний ефект заощадження коштів від застосування запропонованих моделей «як буде» і автоматизації процесу «Рух контингенту» становить 26% на рік.

Особистий внесок здобувача. Дисертаційна робота є самостійно виконаною науковою працею. Всі представлені наукові результати, приклади та експериментальні розрахунки, викладені у дисертації, одержані здобувачем одноосібно. У наукових роботах, що опубліковані у співавторстві, в дисертаційній роботі використані лише ті результати, які становлять індивідуальний внесок автора. У статтях із співавторами автору належить: встановлено мультиагентні методи управління розподіленими обчисленнями у гібридних кластерах [1], метод трансформації моделі організаційно–технічної системи в модель інформаційної системи [2], досліджено системи масового обслуговування на основі імітаційного моделювання з урахуванням мультиагентного підходу [3], проведено інтелектуальний аналіз та обробку текстових документів на основі мультиагентного підходу [4], запропоновано мультиагентний підхід для оптимізації обробки запитів кластерами в реальному часі [5], запропоновано метод потенціалів для керування МРТС в середовищі з перешкодами [6], удосконалено мультиагентний підхід до оцінки стану безпеки інформаційних систем [7], розглянуто та проаналізовано методичні та теоретичні засади підтримки прийняття

рішень, моделювання та розвитку інформаційних систем з використанням мультиагентного підходу [8], удосконалено Мультиагентний підхід до управління ресурсами в розподілених обчислювальних середовищах [9] та для управління розподіленими обчисленнями на рівні GRID–системи [10], проведено імітаційне моделювання розподілених випадкових процесів з використанням мультиагентного підходу [11], виконано розв’язання навігаційної задачі для агента в неоднорідному середовищі [12].

Розроблено метод підтримки прийняття рішень при розробці інформаційних систем, предметної області організаційно-технічних систем, який відрізняється від існуючих:

- застосуванням на етапі обстеження предметної області мультиагентного імітаційного моделювання для аналізу процесів організаційно-технічних систем;
- розглядом розподіленої інформаційної системи у вигляді мультиагентної системи;
- застосуванням фреймово–семантичної моделі подання знань на основі фрейм–концептів та концептуальних графів, з метою формалізації знань про предметні області розробки інформаційної системи та організаційно-технічних систем;
- інтеграцією структурного, об'єктно–орієнтованого та мультиагентного підходу для вирішення задач підтримки прийняття рішень під час автоматизації процесів організаційно–технічної системи;
- перетворенням мультиагентної імітаційної моделі організаційно–технічної системи в основу моделі архітектури інформаційної системи та її елементів, представлення архітектури у вигляді структурних діаграм та діаграм об'єктно–орієнтованого підходу для забезпечення ефективної взаємодії між фахівцями–предметниками та ІТ–фахівцями;
- аналізом ефективного розподілу бази даних за наявності ризику тривалих відмов у обслуговуванні розподіленої інформаційної системи.

Апробація роботи. Результати дисертаційної роботи доповідались та обговорювались на науково–практичних конференціях:

– XIX міжнародна науково–практична конференція молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики» (м. Київ, 20 – 23 квітня 2021 р.);

– XIV міжнародна науково–практична конференція «Інтегровані інтелектуальні робототехнічні комплекси (ПРТК–2021)» (м. Київ, 18 – 19 травня 2021 р.);

– XVII Міжнародна науково–практична конференція «Військова освіта і наука: сьогодення та майбутнє» (м. Київ, 26 листопада 2021 р.);

– Міжнародна науково–практична інтернет–конференція «Розвиток освіти, науки та бізнесу: результати 2020» (м. Дніпро, 3 – 4 грудня 2020 р.);

– VII Міжнародна науково–практична конференція “Priority Directions of Science and Technology Development” (м. Київ, 21 – 23 березня 2021 р.);

– International scientific conference "Innovation around us '2021" (Svishtov, Bulgaria, 9 – 10 березня 2021 р.);

– International scientific conference “The current stage of development of scientific and technological progress'2021”. (Karlsruhe, Germany, 8 – 10 лютого 2021 р.).

Четверта міжнародна науково–практична конференція *«Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій»* 1–2 лютого 2022 р., м. Київ, НУХТ, 2022.

Публікації. За результатами досліджень опубліковано 12 наукових праць. Основні наукові положення викладено в 4 наукових статтях [1–4] у спеціалізованих фахових виданнях України. За матеріалами виступів на науково–технічних конференціях опубліковано 8 тез доповідей [5–12].

Структура і обсяг роботи. Дисертація складається зі вступу, 4 розділів, висновків, 2 додатків та списку використаних джерел із 113 найменувань на 12 сторінках. Повний обсяг дисертації складає 192 сторінки, з них 154 сторінок основного тексту.

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ НАУКОВИХ МЕТОДІВ ЩО ДО РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ НА ОСНОВІ МУЛЬТИАГЕНТНОГО ПІДХОДУ

Згідно з дослідженнями, лише 21 % проектів тривалістю більше двох років виконуються вчасно. Однією з причин такого явища є спотворення і втрата інформації про розроблену інформацію і особливості процесів в ланцюжку її передачі між користувачем, аналітиком, розробником. Технічне завдання (ТЗ) – має бути сполучною ланкою, але воно не повною мірою відображає як предметну область ОТС (в плані процесів координації і прийняття рішень), так і не вирішує питання зв'язування очікувань користувача з вимогами до ІС. Так, наприклад, з усього ТЗ користувач розуміє лише розділ «Функції системи» і сприймає візуальний, користувацький інтерфейс (КІ) як всю інформаційну систему. Також згідно статистики, більшість проектів перевищують початкову оцінку запланованої «вартості», на рис. 1.1 відображено процентне відношення перевищених витрат на період проектування.



Рис. 1.1. Відсоток проектів, які перевищили початкову оцінку «вартості»

1.1. Дослідження системного аналізу в розробці інформаційних систем

Основною проблемою, яка виникає при розвитку інформаційної системи (ІС), є складність розуміння відразу всієї системи в цілому. Для вирішення цієї проблеми доцільно використовувати системний аналіз. Такий підхід дозволяє отримати цілісний вигляд об'єкта автоматизації і розробки. Проведення системного аналізу об'єкта автоматизації при розробці інформаційних систем можна розділити на наступні етапи:

1. Визначення та призначення об'єкта автоматизації.
2. Визначення цілей розробленої системи.
3. Аналіз стану внутрішнього і зовнішнього середовища автоматизованого об'єкта і прогнозування їх змін.
4. Побудова та аналіз моделей автоматизованих процесів.
5. Розробка нових моделей автоматизованих процесів з урахуванням проблем, діагностованих на попередньому етапі.
6. Розробка моделі ІС.
7. Розробка та впровадження отриманої системи.

Розробка інформаційної системи йде у тісній співпраці користувачів, аналітиків та розробників інтереси яких потрібно поєднати. Перші – спеціалісти з галузі. Останні отримують знання від аналітиків і формують вимоги до розробленої ІС. Розробники реалізують отримані вимоги у вигляді готового продукту.

1.1.1. Організаційно–технічні системи

Під об'єктом автоматизації в роботі розглядаються організаційно–технічні системи (ОТС), які є набором організаційної структури і технічних засобів, що знаходяться в його розпорядженні, тобто спільно розглядаються людина і інформаційна система, тобто – складається із засобів автоматизації певного виду діяльності людей та персоналу, що здійснює цю діяльність. У роботі [13] виділяються наступні особливості ОТС: багато параметрична, ієрархічна,

імовірнісна поведінка, складність структури і алгоритми поведінки.

Процеси, що відбуваються в ОТС, можна розділити на три групи:

- виробничі та бізнес–процеси (БП) – процеси, пов'язані з основною діяльністю підприємства;
- гармонізаційні процеси;
- процесів прийняття рішень.

При автоматизації виробничих і БП підприємств підприємства аналітики, опитуючи користувачів, отримують інформацію про структуру і функції ОТС, створюють модель БП та при необхідності вдосконалюють її.

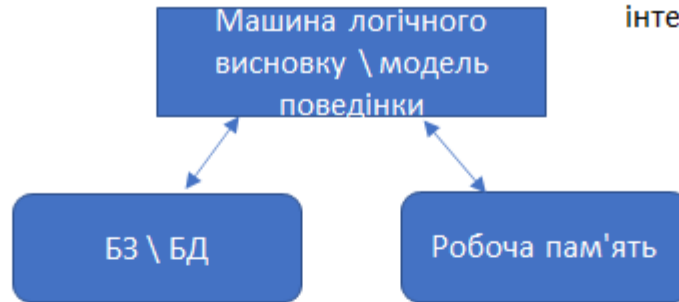
Мультіагентні системи (МАС) найкраще підходять для автоматизації процесів затвердження.

Процеси прийняття рішень мають свої особливості [13]. Перш за все, ці проблеми важко описати алгоритмічно. Рішення приймаються за певними сценаріями, для опису яких доцільно використовувати бази знань і технології експертних систем (ЕС), сценарії частково або повністю будуються з моделі, яка описує деякі аспекти інформаційної системи. При автоматизації діяльності особи, що приймає рішення (ОПР), можна використовувати програмні інтелектуальні агенти. Отже, для таких процесів інформаційна система має включати систему підтримки прийняття рішень (СППР), яка допоможе ОПР на основі наявної інформації правильно визначити проблему та вибрати оптимальне рішення.. Виходячи з наявної інформації, особи, що приймає рішення правильно визначає проблему і вибирає оптимальне рішення. Слід зазначити, що не всі алгоритми і сценарії поведінки піддаються повній формалізації. У деяких випадках потрібна безпосередня участь особи, що приймає рішення.

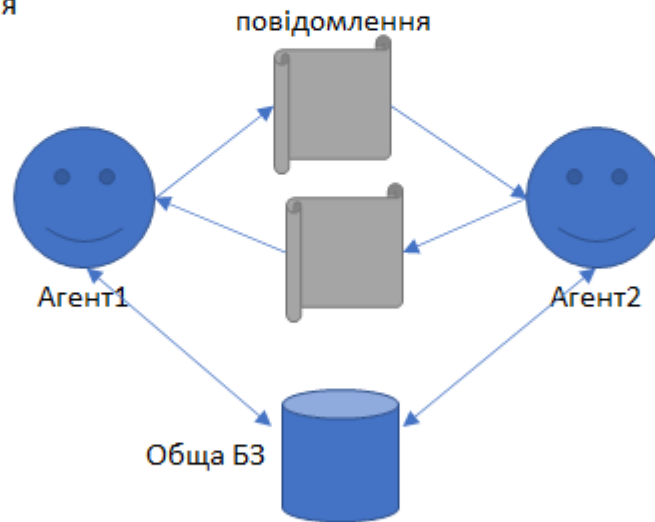
На рис. 1.2 показані особливості автоматизації ОТС.

Автоматизація кожної групи процесів, що відбуваються в ОТС, має свої особливості. Існуючі підходи до розробки, кожен окремо, не охоплюють всі питання, які виникають в даному випадку.

Прийняття рішень

Експертні системи \
інтелектуальні агенти

Процеси узгодження



Бізнес процеси

WorkFlow системи



Рис. 1.2. Особливості автоматизації ОТС

1.1.2. Моделювання автоматизованих процесів

Заміна одного об'єкта на інший з метою отримання інформації про найважливіші властивості вихідного об'єкта за допомогою об'єкта моделі називається моделюванням [14]. Ви можете побудувати кілька моделей для однієї системи, кожна з яких визначає певний аспект системи, використовуючи різні набори діаграм і документів.

Успішне вирішення проблеми створення ІС неможливе без чіткої картини БП, що відбувається на автоматизованому підприємстві. Тому, перш за все,

необхідно побудувати бізнес–модель, яка відображає автоматизовані процеси і необхідні для них ресурси.

У процесі аналізу предметної області при розробці ІС будуються моделі діяльності підприємства. Вони можуть бути двох видів:

- моделі «AS–IS» («як є»), які розсіюють існуючі бізнес–процеси підприємства;
- моделі "AS–TO–BE" ("як буде"), які показують уявлення про нові процеси і технології підприємства.

Перехід від моделі «AS–IS» до моделі «AS–TO–BE» обумовлений реінжинірингом (розділ 1.1.3).

Далі на основі моделей «AS–TO–BE» будуються моделі розробленого ІС. Наявність моделей ІС також позитивно впливає на документацію проекту, так як прийняті проектні рішення стають більш помітними. Склад моделей, що використовуються в кожному конкретному проекті, ступінь їх деталізації залежить від наступних факторів [15]:

- складність розробленої системи;
- необхідна повнота його опису;
- знання та навички учасників проекту;
- час, відведений на розробку.

Оскільки проектування і моделювання тісно пов'язані один з одним, то при моделюванні БП підприємства і структури програмного забезпечення(ПЗ) також використовуються структурні і об'єктно–орієнтовані методи.

Структурні методи аналізу

Структурний аналіз – це метод вивчення системи, починаючи з її загального огляду, який потім деталізується, набуваючи ієрархічну структуру зі збільшенням числа рівнів [16]. Вона характеризується:

- розділення системи на рівні абстракції з обмеженням кількості елементів на кожному рівні;
- обмежений контекст, включаючи лише основні деталі на кожному рівні;

- використання строгих формальних правил запису;
- послідовне наближення до кінцевого результату.

У структурному аналізі і проектуванні використовуються різні моделі, найбільш поширеними є:

- функціональна модель SADT (IDEF0), яка описує функціональну структуру системи [17, 18];
- модель EDEF3, в якій описуються процеси, в яких важливо розуміти послідовність виконання дій і взаємозалежності між ними, моделі IDEF3 часто використовуються для деталізації функціональних блоків IDEF0 [19, 21];
- Діаграми потоку даних DFD, які описують передачу інформації між функціональними процесами [19, 22];
- Специфікація BPMN (англ. Business Process Model and Notation – Модель бізнес-процесів і нотація) дозволяє описати БП у вигляді послідовності керуючих витрат об'єктів з можливістю зазначення циклічних дій [23].

Сімейство IDEF включає в себе інші моделі, які широко не використовуються. Вона дозволяє описати правила і обмеження, які представляють стан системи, використовуючи словник термінів [24].

Можна виділити наступні позитивні особливості структурних методів:

- графічні діаграми дозволяють візуалізувати структуру системи;
- ієрархічна структура діаграм дозволяє описати функції системи з різним ступенем деталізації;

При розробці ІС доцільно використовувати діаграми потоку даних DFD [19], оскільки цей метод добре узгоджується з інструментом моделювання даних (моделлю зв'язку сутностей). Їх комбіноване використання Склади даних, описані на діаграмах DFD, є основою для побудови моделі зв'язку сутностей.

Об'єктно-орієнтовані методи аналізу та проектування програмного забезпечення IDEF4 – це методологія побудови об'єктно-орієнтованих систем. Інструменти IDEF4 дозволяють візуально відображати структуру об'єктів і вбудовані принципи їх взаємодії, тим самим дозволяючи аналізувати і оптимізувати складні об'єктно-орієнтовані системи.

Мова UML [26, 27, 28] є в даний час широко використовуваним методом об'єктно-орієнтованого аналізу. Мова UML – це набір діаграм, які дозволяють описати розвинену інформаційну систему з двох сторін: логічна \ фізична і статична \ динамічна. Такі діаграми: класи, об'єкти, компоненти та розгортання. Для опису динамічного компонента ІП використовуються наступні: прецеденти, узгодженість, співпраця, держави та діяльність.

Широке використання UML багато в чому обумовлено можливістю розширення існуючих мовних елементів, що дозволяє створювати мови для використання в певній предметній області на основі UML. Ця мова включає в себе наступний набір діаграм [28]:

- 1) приклади використання – призначені для визначення набору функцій, які буде виконувати ІС;
- 2) діяльності – дозволяє описати діяльність у вигляді послідовності дій, які виконуються різними елементами, що входять до системи;
- 3) послідовності – ІС-об'єкти взаємодіють один з одним, передаючи повідомлення один одному; ці діаграми дозволяють описати послідовність передачі повідомлень між об'єктами;
- 4) комунікації – призначені для представлення взаємодії в контексті внутрішньої архітектури системи та повідомлень, які повинні бути розглянуті;
- 5) огляду взаємодії – призначені для представлення взаємодії тільки в контексті керуючого потоку в якійсь агрегованій формі;
- 6) тимчасова – являє собою зміни стану окремих життєвих ліній або особливості синхронізації повідомлень у часі;
- 7) скінченний автомат – граф, який представляє якийсь скінченний автомат;
- 8) класів – дозволяють створити логічне представлення системи, тобто описати ієрархію класів, які будуть реалізовувати ІС об'єкти, методи і властивості цих класів;
- 9) композитної структури – дозволяє зобразити внутрішню структуру або склад класів і взаємозв'язки між ними;
- 10) пакетів – призначені для представлення розміщення елементів моделі в

пакетах і специфікації залежностей між пакетами та їх елементами;

11) об'єкти – призначені для визначення об'єктів і зв'язків між ними на фіксований момент часу;

12) компоненти – призначені для представлення фізичних компонентів системи та їх взаємодії;

13) розгортання – дозволяє описати апаратні вимоги розробленої ІС: процесори, пристрої та їх з'єднання; для кожної моделі побудована одна діаграма.

Моделювання БП є компонентом розробки ІС. Існують структурні та об'єктно–орієнтовані методи аналізу та проектування програмного забезпечення. Кожен з них має свої переваги та недоліки. Для отримання найбільшого ефекту в роботі поставлено і вирішено завдання інтеграції цих методів для завдання розробки ІС.

1.1.3. Реінжиніринг бізнес–процесів

Сучасне підприємство має досить складну структуру і алгоритми роботи. На початковому етапі розробки ІС необхідно переконатися, що організація існуючих БП дозволяє підприємству працювати найоптимальніше, і, якщо це не так, то внести відповідні зміни. Для цього використовується реінжиніринг бізнес–процесів (РБП) – фундаментальне переосмислення і радикальний редизайн БП для досягнення фундаментальних поліпшень основних показників діяльності підприємства [29]. Метою РБП є системна реорганізація всіх потоків підприємства (матеріальних, фінансових та інформаційних), що має призвести до спрощення організаційної структури, перерозподілу та мінімізації використання різних ресурсів, скорочення часу на реалізацію потреб клієнтів. Потім будується модель існуючих БП компанії, проводиться її аналіз. В результаті пропонуються зміни, які покращують БП підприємства [30]. На даному етапі доцільно використовувати імітаційні інструменти. Це дозволяє оцінити ефективність роботи на різних етапах БП і виявити проблемні ділянки. Після цього розробляється нова модель БП, відбувається її реалізація.

Найбільш поширеним засобом моделювання динамічних процесів (переходів від одного стану до іншого (від однієї ситуації до іншої) є імітаційне моделювання і, зокрема, дискретно–подійне моделювання [31, 32, 33].

Мультиагентний підхід також використовується в імітаційному моделюванні. При цьому поведінка системи визначається як результат діяльності багатьох агентів. Для побудови мультиагентної моделі необхідно визначити поведінку окремих агентів і взаємозв'язки між ними [34].

Таким чином, для аналізу, вдосконалення та реінжинірингу бізнес–процесів в організаційно-технічних системах використовуються інструменти моделювання та мультиагентного моделювання, що підвищує ефективність автоматизації.

1.2. Дослідження ризиків пов'язаних з розробкою інформаційних систем і шляхи їх зниження

Розробка ІС супроводжується певними ризиками [35]. Перерахуємо деякі з них:

1. Проміжні етапи можуть бути відстають від графіка або бюджет проекту може бути значно перевищений. Це може призвести до закриття проекту до завершення, отже, ІС навіть не буде розроблений.

2. Низька якість реалізованої ІС може призвести до відмови від його використання.

3. Розроблена ІС втрачає свою корисність після декількох років експлуатації, так як кількість недоліків і вартість внесення змін збільшується настільки, що стає простіше і дешевше розробити нову систему, ніж підтримувати існуючу.

4. В процесі експлуатації ІС виявляється, що вона не вирішує необхідних завдань підприємства. Це може бути наслідком спочатку неточного постановки завдання або змін в БП при розробці ІС, які не були враховані.

5. В процесі експлуатації ІС виявляється, що вона має безліч функцій, які не використовуються замовником, і дійсно корисні не реалізовані в системі.

6. Протягом життєвого циклу ІС відбувається практично повне оновлення

команди розробників. Недостатня документація системи не дозволяє новим співробітникам успішно модернізувати ІС.

В даний час створення ІС розглядається як формалізований процес, який повинен відповідати певним стандартам і нормативним документам [19]. Створюються різні методи і технології розробки ІС, що знижують ризики, пов'язані з його розробкою, підвищують продуктивність учасників ІТ-проекту, підвищують якість розробленої системи за рахунок підвищення керованості процесу створення ІС і своєчасно з урахуванням змінених вимог до системи.

Слід зазначити, що ефективність застосування цих методів і технологій безпосередньо пов'язана з наявністю програмного інструменту, що забезпечує автоматизацію етапів – CASE–affinity (Computer Aided Software Engineering – комп'ютерна інженерія програмного забезпечення).

Метод розробки включає в себе:

- певна послідовність етапів розвитку;
- список нотацій (графічних і текстових інструментів), що використовуються для опису створеної системи;
- критерії та правила, що використовуються для оцінки отриманих результатів.

Проведене дослідження організаційно-технічних систем показало, що автоматизація кожної групи процесів, що відбуваються в організаційно-технічних системах, має свої особливості. Існуючі підходи до розробки, кожен окремо, не охоплюють всі питання, які виникають в даному випадку.

1.3. Аналіз методологічних та теоретичних основ підтримки прийняття рішень, моделювання та розвитку інформаційних систем

Проведемо аналіз існуючих наукових методів що до розробки інформаційних систем на основі мультиагентного підходу та аналіз етапів системного аналізу в розробці інформаційних систем.

1.3.1. Застосування імітаційного моделювання в розробці програмного забезпечення

Прийняття рішення про введення єдиної ІС на підприємстві пов'язане з багатьма труднощами. Основні проблеми, які виникають в цьому випадку, можна виділити наступним чином:

- 1) наявність декількох непов'язаних ІС, які вирішують вузькі проблеми;
- 2) неоптимальні БП, які не дозволяють компанії ефективно працювати;
- 3) складність прогнозування характеристик ІС (швидкість обробки запитів в базу даних, швидкість передачі даних по мережі і т.д.).

Для вирішення першої проблеми необхідно розробити нову базу даних для єдиної ІС і перенести на неї накопичені дані.

Найбільш ефективним способом вирішення другої проблеми є побудова моделі існуючої БП і проведення їх РБП [18, 37]. Імітаційне моделювання (ІМ) на даному етапі істотно допомагає в аналізі такої складної системи, як сучасне підприємство.

Імітаційна модель описує закони функціонування кожного елемента системи та їх взаємозв'язок. В результаті імітаційного експерименту на початкових даних можна отримати дані про стани БП в певний момент часу.

Це дозволяє оцінити характеристики системи і краще зрозуміти, як розробити систему, яка відповідає заданим критеріям оцінки ефективності [18]. Для отримання достовірних і точних результатів ІМ з невеликою кількістю комп'ютерного часу необхідно планувати експерименти. Воно буває стратегічним та тактичним [18]. У першому випадку визначають які фактори та які їх рівні використовувати в експерименті, щоб отримати максимум інформації про систему без повного перебору. У другому випадку визначають методи проведення експериментів (тривалість проходження, точність результату).

Для прогнозування характеристик експлуатації ІС можна розширити модель моделювання таким чином, щоб вона включила не тільки описи автоматизованих БП, їх статистичні характеристики, такі як час обробки заявки (документа), час

простою в черзі, кількісні характеристики процесів, а також часові та кількісні характеристики операцій в ІС. Експерименти з такою імітаційною моделлю дозволять отримати уявлення про її виконання до реалізації готової ІС, вимоги до швидкості та інтенсивності обробки заяв (документів) та транзакцій у базі даних.

Використання ІМ при розробці ІС доцільно як на етапі аналізу автоматизованих БП з метою їх оптимізації, так і перед розробкою ІС для прогнозування характеристик його експлуатації.

1.3.2. Використання систем штучного інтелекту при розробці інформаційних систем

Використання штучного інтелекту (ШІ) при розробці ІС виводить цей процес на якісно новий рівень, так як дозволяє автоматизувати не тільки рутинні операції, але і процес прийняття рішень при створенні ІС. Для цього в інструменти, що підтримують процес розробки, слід віднести ЕС. Оскільки повністю виключити людину з процесу розробки ІС доцільно використовувати діалогову ЕС [38].

У БЗ інтелектуальної СППР у галузі розробки ПЗ (CASE–засобі) повинні зберігатися факти та правила, отримані від експертів предметної галузі, а також професійні знання розробників. Використання такої СППР дозволять вирішувати практичні завдання, що виникають у процесі розробки ІС

Розвиток інформаційних технологій визначив підвищену увагу до комплексного використання ІС та ЕС в частині автоматизації діяльності ОПР. Такі системи можуть використовуватися в завданнях, які вимагають взаємодії ряду фахівців з тематики. Використання такої системи підтримки прийняття рішень дозволять вирішувати практичні завдання, що виникають у процесі розробки ІС. Використання БЗ та машини логічного виводу (МЛВ) в інформаційній системі дозволять автоматизувати пошук рішення, якого міг би прийти у аналогічній ситуації експерт [38].

З урахуванням великого обсягу даних, знань фахівців, формування відповіді на основі певного аналога процесу міркування, діалогу з користувачем – все це

робить ЕС інструментом, який зробить процес створення інформаційної системи розумнішим і автоматизує роботу ОПР.

1.3.3. Застосування мультиагентного підходу

Мультиагентні системи (МАС) являють собою новий напрямок у розвитку штучного інтелекту, який сформувався на основі результатів досліджень в області розподілених комп'ютерних систем, мережових технологій для вирішення проблем і паралельних обчислень.

Агентський підхід вже використовується в розподіленому вирішенні складних завдань, реінжинірингу підприємств, електронному бізнесі, логістиці, однаковому моделюванню. При класифікації Д.А. Поспелова виділяє два класи проблем:

1) завдання розподіленого управління і завдання планування досягнення цілей, в яких агенти спільно вирішують поставлені проблеми, і необхідно забезпечити ефективний спосіб співпраці їх діяльності;

2) локальні завдання, в яких агенти використовують загальні, як правило, обмежені ресурси.

Агент – це апаратна або програмна організація, здатна діяти в інтересах досягнення цілей, поставлених його власником або користувачем [41]. Агенти також описуються рядом властивостей, що характеризують поняття агенту. Зазвичай агент має набір наступних властивостей [39]:

- адаптивність: агент має здатність вчитися;
- автономія: агент працює як самостійна програма, ставлячи перед собою цілі і виконуючи дії для досягнення цих цілей;
- співпраця: агент може взаємодіяти з іншими агентами кількома способами, наприклад, граючи роль постачальника / споживача інформації або обох; – здатність міркувати: агенти можуть мати часткові механізми знань або висновків, наприклад, знання про те, як довести дані з різних джерел до однієї форми.
- комунікабельність: агенти можуть спілкуватися з іншими агентами;

- мобільність : можливість передачі коду агенту з одного сервера на інший.

Слід зазначити, що агенти можна класифікувати наступним чином:

- реактивний;
- інтелектуальний;
- гібридні.

Реактивний агент приймає рішення на основі знань про ситуацію–дії. Розумний агент вирішує поставлені перед ним завдання, виходячи зі своїх цілей, використовуючи загальні обмежені ресурси і знання про зовнішній світ. Гібридний агент поєднує в собі можливості перших двох.

Інтелектуальний МАС – це набір інтелектуальних агентів, розподілених по мережі. Вони відстежують необхідні дані і взаємодіють один з одним для досягнення своїх цілей. Можна виділити кілька важливих причин взаємодії агентів: сумісність цілей (спільна мета); ставлення до ресурсів; необхідність залучення відсутнього досвіду; взаємних зобов'язань [39].

З точки зору програмування агентами є програмне забезпечення, яке може діяти незалежно від імені користувача. При створенні МАС може використовуватися клієнт–серверна архітектура. Залежно від завдання можливі дві реалізації. Перший, «товстий клієнт – тонкий сервер» – серверна частина реалізує доступ до ресурсів, а додатки розташовуються на клієнтських комп'ютерах. інтерфейс, а сервер виконує всі інші програмні завдання.

Впровадження програмних агентів в ІС дозволить , в якійсь мірі, спростити роботу користувачів, так як агенти зможуть стежити за станом системи і пропонувати певні рішення.

Важливою проблемою в розвитку ІС є отримання знань фахівців в певній предметній області. Здебільшого ці знання не формалізовані, а тому недоступні іншим людям. Успішним рішенням цієї проблеми є додавання загального БЗ до створеної ІР–адреси та розробка агентів, які, спираючись на ці знання, запропонують рішення певних проблем в автоматизованій області. який виконує певні формалізовані функції користувачів і надає підтримку у вирішенні завдань організаційно–технічного управління.

Також при розробці розподілених ІС може бути використаний мультиагентний підхід.

Отже, мультиагентні системи являють собою новий напрямок у розвитку штучного інтелекту, який сформувався на основі результатів досліджень в області розподілених комп'ютерних систем, мережових технологій для вирішення проблем і паралельних обчислень.

1.4. Огляд та порівняльний аналіз існуючих систем підтримки прийняття рішень у сфері розробки інформаційних систем (CASE–інструменти).

Проведемо огляд існуючих CASE–інструментів приведемо їх класифікацію, опис, визначимо критерії порівняння функціоналу та проведемо порівняльний аналіз інструментів.

1.4.1. Класифікація CASE–інструментів

Сучасні інструменти CASE охоплюють широку сферу підтримки численних технологій проектування ІС, від простих інструментів аналізу та документації до повномасштабних інструментів автоматизації, які охоплюють весь життєвий цикл програмного забезпечення [42].

Етапи аналізу і проектування є найбільш трудомісткими в розробці АН, тому використання CASE–інструментів дозволяє підвищити якість прийнятих технічних рішень і полегшити підготовку проектної документації. CASE–інструменти дозволяють будувати різні типи діаграм, при цьому перевіряючи існуючі для них синтаксичні правила.

Сучасний ринок програмного забезпечення має сотні різних CASE–інструментів [42]. Якщо це досить дешеві системи для персональних комп'ютерів з дуже обмеженими можливостями, і потужні дорогі системи, які підтримують весь життєвий цикл програмного забезпечення.

Як правило, CASE–інструменти включають в себе будь–який програмний

інструмент, який автоматизує певний набір процесів життєвого циклу програмного забезпечення і має наступні основні характерні особливості [42]:

- потужні графічні інструменти для опису та документування ІС, забезпечення зручного інтерфейсу з розробником та розвитку його творчих можливостей;
- інтеграція окремих компонентів CASE-інструментів, забезпечення керованості процесу розробки ІС;
- використання спеціальним чином організованого сховища проектних метаданих.

Найбільш використовуваними CASE-інструментами є Modeling Suite, Rational Suite, ELMA BPM, ARIS ToolSet, Power Designer и Borland Together Designer, BizAgi BPM Suit, CA ERwin у наступному розділі дано їх описання.

1.4.2. Опис CASE-інструментів

CA ERwin Modeling Suite

CA ERwin Modeling Suite – це набір інструментів CA Technologies, які повністю забезпечують вирішення всіх проблем аналізу, проектування, генерації, тестування та обслуговування інформаційних систем [43, 44]. Стисло опишемо продукти, що входять в даний пакет.

Process Modeler (раніше BPwin) [45, 46] – це інструмент візуального моделювання бізнес-процесів. Це дає можливість представити будь-яку діяльність або структуру у вигляді моделі. Візуальне моделювання дозволяє оптимізувати роботу організації, перевірити її на відповідність стандартам ISO 9000, спроектувати організаційну структуру, усунути непотрібні операції. Модельєр процесу підтримує три модельні нотації: IDEFO, IDEF3 та DFD.

ERwin Data Modeler (ERwin) дозволяє проектувати, документувати і обслуговувати бази даних і сховища даних. Модель візуальної бази даних полегшує оптимізацію структури бази даних, що дозволяє домогтися її повної відповідності вимогам і завданням підприємства. Erwin підтримує такі СУБД: DB2 для LUW, SQL Server і Teradata, Oracle, InterBase, Ingres.

Component Modeler (Paradigm Plus) [47] – це інструмент для моделювання

програмних компонентів і генерації коду об'єкта додатків на основі створених моделей. Продукт може використовуватися як при створенні нових додатків, так і при модифікації або об'єднанні існуючих. Інтегрований з Process Modeler, що дає додаткові можливості при роботі з функціональними моделями. Component Modeler забезпечує повну підтримку UML, підтримує синхронізацію проектування та реалізації додатків. Component Modeler підтримує генерацію коду та зворотну розробку Microsoft .NET (C# і Visual Basic), Microsoft Visual J++, і Microsoft Visual C++.

Так, CA ERwin Modeling Suite дозволяє описати модель БП в нотаціях IDEF0, IDEF3, DFD, розробити модель даних і архітектуру програмного забезпечення з підтримкою різних СКБД. Для аналізу і оптимізації БП пропонується використовувати функціональний і витратний аналіз.

Продукти IBM Rational

Продукти IBM Rational [48, 49, 50, 51] – це набір інструментів, які підтримують весь життєвий цикл розробки програмного забезпечення і інтегровані з IBM Rational Unified Process (RUP). Технічно RUP розроблений як веб-база знань, яка оснащена пошуковою системою. Вона включає в себе: великі посібники для розробників для кожного часового інтервалу життєвого циклу програмного забезпечення; ручне використання інструментів; приклади та шаблони.

CASE інструмент IBM Rational Software Architect надає розробникам наступні можливості: аналізувати та проектувати програмне забезпечення за допомогою UML, аналізувати та контролювати структуру java-додатків, а також підготувати проектну документацію, а також базуватися на технології Eclipse. Розширення Simulation дозволяє моделювати поведінку програмного забезпечення на основі UML-діаграм діяльності, послідовності, зв'язку або діаграм стану. Рациональний архітектор програмного забезпечення підтримує генерацію коду та зворотну розробку: Microsoft Visual Studio та .NET Framework для C# і VB., Java і C++.

Технологія розробки програмного забезпечення, реалізована в IBM Rational, також може бути використана для проектування користувацьких інтерфейсів (UI).

Розробка пі заснована на концепції розкадрування прикладів використання, і є моделлю взаємодії користувача з інтерфейсом користувача. Перероблені рішення до того, як буде розроблений і реалізований фактичний інтерфейс. Розглянемо послідовність проектування UI розробленої системи:

1. Діаграми інцидентів використання призначені для відображення взаємодії користувача з UI.
2. Для кожного випадку використання будується діаграма послідовності, яка показує всі можливі взаємодії користувача з елементами UI.
3. Кожна форма заявки представлена на діаграмі класу як окремий клас, елементи форми – атрибути класу, дії користувача і системи, пов'язані з вікном, є методами класу. Успадкування класу використовується для опису складних форм.

Прототипи будуються на основі отриманих моделей UI.

Відзначимо ще кілька продуктів: IBM Rational Requirements Composer – займається визначенням і управлінням вимогами на всіх етапах життєвого циклу програмного забезпечення; IBM Rational Team Concert – інструмент для керування конфігураціями програмного забезпечення, збірками та звітуваннями; Rational Quality Manager надає можливість планувати, виконувати та контролювати стан ручного та автоматизованого тестування програмного забезпечення.

IBM Rational Rose Data Modeler – це інструмент моделювання даних у нотації ER

Продукти IBM Rational орієнтовані на розробку архітектури і програмного забезпечення організації, управління життєвим циклом створеної інформаційної системи, не торкаючись питань, пов'язаних з моделюванням даних.

Набір інструментів ARIS ToolSet

Інтегроване середовище ARIS ToolSet [52] – це набір інструментів для:

- проектування та управління підприємством;
- моделювання, аналіз та оцінка бізнес-процесів;
- документація бізнес-процесів відповідно до вимог міжнародних стандартів;
- розробка, впровадження та обслуговування ІС.

Методологія моделювання та проектування ІС, реалізована в цій системі,

базується на наборі різних методів моделювання, які відображають різні погляди на розроблену систему (організація, функції та цілі, дані, продукти та послуги, процеси). При побудові моделей в ARIS ToolSet можна використовувати як власні методи моделювання ARIS, так і відомі мови моделювання, наприклад, UML, BPMN).

Набір інструментів ARIS включає в себе наступні додаткові модулі:

- ARIS ABC – для функціонального та вартісного аналізу бізнес–процесів;
- ARIS Simulation – для моделювання різних БП;
- ARIS Weblink – для організації віддаленої роботи;
- ARIS Connectivity for Lotus Notes – робота з документами на основі Lotus Notes;
- ARIS Connectivity for SAP/R3 HR – для організації взаємодії з відповідними модулями системи R/3;
- різні інтерфейси з інструментами розробки (Designer, ERWin, PowerDesigner, Rational Software Architect тощо) – реалізувати наступні етапи проектування, аналізу та впровадження ІС.
- ARIS є основою пакету аналізу бізнес–процесів Business Process Analysis Suite Oracle, що дозволяє автоматично конвертувати модулі процесів в компоненти SOA (англ. service–oriented architecture, сервісно–орієнтована архітектура) [58].

Інтегроване середовище ARIS ToolSet займається імітаційним моделюванням БП, їх аналізом, вдосконаленням та оптимізацією, можливе використання нотації ER для розробки моделі даних. Питання, пов'язані з розробкою архітектури програмного забезпечення, вирішуються за допомогою інших додатків, з якими ARIS має інтеграцію.

Power Designer

Power Designer є інструментом CASE від Sybase і має модульну архітектуру [59]. Power Designer має такі функції:

- структурне моделювання бізнес–процесів;
- концептуальне та фізичне проектування та генерація баз даних (підтримує понад 60 СКБД);

- об'єктно–орієнтований аналіз і моделювання даних за допомогою UML;
- Інтеграція з провідними середовищами розробки (Eclipse, Microsoft Visual Studio®, PowerBuilder).

Power Designer підтримує моделювання даних, статичне моделювання БП та моделювання додатків. Він безпосередньо не вирішує наступних питань: вимоги до програмного забезпечення та управління конфігурацією; моделювання моделювання.

Borland Together

Borland Together [28, 55] – це програмний комплекс Micro Focus/Borland. Він призначений для створення моделей, які представляють схему бізнес–процесів, структуру даних, архітектуру додатків і підприємства. Borland Together складається з наступних програм: , Borland Together Designer Community Edition, Borland Together Designer и Borland Together Developer..

Borland Together надає наступні функції:

- візуальне моделювання метамodelів для конкретної предметної області;
- моделювання бізнес–процесів з використанням моделювання бізнес–процесів, нотацій Business Process Modeling, BPMN;
- розробка проектів графічних моделей програмних додатків в нотації мови UML;
- розробка логічних діаграм даних «сутності–зв'язку» в нотації ER та IDEF1x;
- передова і зворотна інженерія для провідних СУБД (Oracle, DB, Sybase, MS SQLServer);
- створення документації;
- генерація програмного коду для Java, C++ і C#;
- розпізнавання шаблонів проектів вихідного коду;
- Створюйте діаграми тестових випадків з вимог та елементи моделі трасування від вимог та вимог за допомогою CaliberRM та RequisitePro.

Borland Together – це CASE–інструмент, який автоматизує процес розробки програмного забезпечення на етапах моделювання БП, створення схем "сутності–зв'язку" з можливістю створення структури БП, моделювання програмного

забезпечення, що закінчується генерацією програмного коду.

BizAgi BPM Suit

Система BizAgi BPM Suit [56, 57] є платформою для автоматизації БП. Для моделювання БП використовується додаток BizAgi Process Modeler, який використовує нотацію BPMN. Bizagi Studio дозволяє конвертувати описаний БП в запусканий додаток без участі програміста. Модель знаходиться в серверному сховищі, інтерпретується і виконується через веб-додаток на Bizagi BPM Server. Даний сервер дозволяє аналізувати різні показники продуктивності БП з метою виявлення проблемних подій і їх поліпшення. Є доступ до зовнішніх джерел даних (Oracle, MS SQL Server).

Важливо робити зауваження щодо функціональності додатків, розроблених з BizAgi і змодельованих процесів: 1) функціональність додатків відноситься до класу Work Flow; 2) Моделювання стосується тільки організаційних процесів.

Система BizAgi BPM Suit відноситься до класу систем управління блок-блоком і дозволяє розробляти проблемно-орієнтовані додатки класу Work Flow, з аналітикою і можливостями управління. Ця система не підтримує проектування архітектури більш широкого класу програмного забезпечення.

ELMA BPM Suite

ELMA BPM Suite [58] включає в себе додаток для управління індикаторами БП і управління БП. У програмі «ELMA Designer» моделювання БП здійснюється за допомогою BPMN-нотації. Розроблена модель зберігається на сервері додатків. Далі виконується БП, і можна запустити кілька екземплярів одного і того ж БП для виконання. Користувач отримує інформацію про завдання, які йому потрібно виконати у вигляді сторінок в браузері. Система автоматично генерує завдання для кожного співробітника при виконанні БП. За допомогою спеціальних додатків БП контролюється і контролюється.

Система управління ELMA Designer BPM Suite дозволяє розробити внутрішній корпоративний портал для управління його роботою. ELMA не

підтримує UML і проектування широкого класу програмного забезпечення, імітаційне моделювання процесів.

1.4.3. Критерії порівняння функціоналу CASE–інструментів

Для порівняльної оцінки функціональності CASE–інструментів пропонується наступний набір критеріїв:

- а) Нотації, що використовуються в описі бізнес–процесів і архітектури ІС (IDEF0, DFD, UML, BPMN).
- б) Використання імітаційного моделювання для аналізу, вдосконалення та оптимізації БП.
- в) Розробка програмного забезпечення на основі моделі моделювання БП. Ця функція дозволяє використовувати інформацію з моделі при модифікації програмного забезпечення, що знижує складність роботи.
- г) Розробка моделей осіб, які приймають рішення (ОПР).
- д) Опис сценаріїв прийняття рішень ОПР.
- е) Дизайн інтерфейсу користувача. Розробка макета (попереднього прототипу) інтерфейсу користувача.
- ж) Перетворення деяких діаграм на інші. Для створення нових діаграм, а не для побудови діаграм, а не для побудови діаграм з самого початку, використовується інформація, яка вже відображена в перших діаграмах.
- з) Проектування структури бази даних.
- и) Можливість створення виконуваного коду.
- к) Генерація технічної документації для прогнозованого ІС. Оскільки інформація про проект зберігається в єдиній базі даних і автоматично оновлюється при внесенні до нього змін, в будь–який момент можна автоматично створювати актуальну технічну документацію.

1.4.4. Порівняльний аналіз CASE–інструментів

Нижче наведено порівняння ARIS ToolSet, Borland Разом Designer, Power Designer, IBM Rational, CA ERwinModeling Suite, програмних продуктів Biz Agi та Elma (табл. 1.1). Це найпоширеніші серед користувачів – розробників програмного забезпечення.

Таблиця 1.1

Порівняння CASE–інструментів

Критерії порівняння	IBM Rational	CA ERwin Modeling Suite	ARIS ToolSet	Power Designer	Borland Together Designer	BizAgi BPM Suit	ELMA BPM BPM Suit
Підтримка IDEFO, DFD	Ні	Так	Так	Ні	Ні	Ні	Ні
Підтримка UML	Так	Так	Так	Так	Так	Ні	Ні
Підтримка BPMN	Ні	Ні	Так	Ні	Так	Так	Так
ІМ БП	Ні	Ні	Так	Ні	Ні	Так	Ні
Проектування ПЗ на основі ІМ БП	Так	Так	Так	Так	Так	Ні	Ні
Розробка моделі ОПР	Ні	Ні	Ні	Ні	Ні	Так	Так
Опис сценарію прийняття рішень ОПР	Ні	Ні	Ні	Ні	Ні	Так	Ні
Проектування UI	Ні	Ні	Ні	Ні	Ні	Так	Ні
Конвертація діаграм	Ні	Ні	Ні	Ні	Ні	Ні	Ні
Проектування структури БД	Так	Так	Так	Так	Так	Ні	Ні
Генерація тех. документ до створюваної ІС	Так	Так	Так	Так	Так	Так	Ні

У пакеті CA ERwin Modeling Suite немає засобів ІМ БП, але можна експортувати модель в систему моделювання Arena.

Aris Toolset, крім цих нотацій, використовує велику кількість інших нотацій (діаграми Чена, Object Modeling Technique). Слід зазначити, що велика кількість нотацій вимагає додаткових знань від аналітика для їх використання. Пакет Aris Toolset має адаптований модельний движок, до недоліків якого можна віднести відсутність підтримки моделювання систем масового обслуговування [59].

Power Designer і Borland Together Designer – це в першу чергу інструменти для розробки UML–діаграм і генерації програмного коду, вони не надають можливостей для аналізу і моделювання моделювання процесів.

Розглянуті системи класу BPM підходять для розробки вузького класу систем.

Всі розглянуті продукти можна розділити на чотири групи:

а) Системи, які охоплюють весь життєвий цикл програмного забезпечення (продукти IBM Rational, набір моделювання CA ERwin).

б) Системи, що автоматизують розробку програмного забезпечення (Power Designer і Borland Together Designer).

в) Системи аналізу та моделювання БП (Aris Toolset).

г) Системи управління БП (BizAgi і Elma).

Оскільки робота вирішує проблеми автоматизації ОТС–процесів, кращими варіантами є системи групи 3 (Aris Toolset і Enterprise Architect).

До недоліків розглянутих кейсів можна віднести наступне:

- відсутня інтеграція структурного та об'єктно–орієнтованого підходу;
- відсутність інтелекту процесу проектування – проблема автоматичного переходу до проектування одних діаграм на основі інших не вирішена;
- управління блоками тільки певного класу систем (Work Flow);
- відсутність пакетів для ІМ БП, а отже, і можливість використання інформації з моделі при проектуванні ІС.

Пропонується розробити метод підтримки прийняття рішень при розробці ІС і CASE–інструмент, який усунув би ці недоліки.

1.5. Постановка наукового завдання та часткових завдань досліджень

Успіх розробки ІС багато в чому визначається розробкою методологічного підходу, використовуваного в процесі проектування. Слід зазначити наступні моменти. По–перше, існуючі методи і інструменти не забезпечують єдину модель інформаційної системи, як з точки зору розробника, так і користувача – предметного фахівця. По–друге, ОТС характеризується процесами прийняття рішень, які передбачають роботу зі знаннями, що формалізовані сценаріями, а в деяких випадках пов'язані з координацією рішень. Існуючі методи не дозволяють комплексно вирішувати питання формалізації та інформатизації процесів прийняття рішень. По–третє, для аналізу, вдосконалення та перепроєктування бізнес–процесів (БП) в ОТС використовуються симуляційні та мультиагентні засоби моделювання.

Однак використання цих інструментів на етапах автоматизації та інформатизації все ще обмежене з двох причин: з однієї сторони, суттєві витрати на розробку імітаційної моделі, а з іншої – відсутність можливостей використовувати отримані результати і знання на етапах автоматизації.

В даний час існують різні підходи для вирішення задачі розробки методу підтримки прийняття рішень. Структурний підхід (IDEF0, DFD) дозволяє описати систему, що розробляється у вигляді ієрархії взаємопов'язаних функцій. Ця ідея зрозуміла аналітику і користувачеві. Для аналізу вузьких місць і динамічних характеристик використовується імітаційне моделювання. Експертні системи закривають питання, пов'язані з описом знань і сценаріїв прийняття рішень. Мультиагентні системи (МАС) дозволяють автоматизувати процеси прийняття рішень і взаємодію осіб, які приймають рішення (ОПР). Програмні агенти виконують функції ОПР. Кожен з них окремо не закриває всі питання, які виникають при автоматизації ОТС–процесів.

Таким чином, актуальним є аналіз існуючих динамічних моделей процесів ОТС та моделей архітектури інформаційних систем, та на їх основі, створення методу ППР, що поєднує у собі ці підходи, а також програмного забезпечення для

його реалізації – системи підтримки прийняття рішень (СППР).

Великий внесок та розвиток питань підтримки прийняття рішень та мультиагентних систем зробили роботи наступних дослідників: Александрова Д.В., Боема Б., Буча Г, Вендрова А.М., Коршевніюка Л.О., Гейна К., Городецького В.І., Калянова Г.М., Тоценко В.Г., Карсаєва О.В., Коваленко А.Є., Мінського М., Бідюка П.І., Ойхмана Є.Г., Попова Е.В., Тимошука О.Л., Рамбо Дж., Ройса У., Сарсона Т., Пилиповича А.Ю., Хаммера М., Дж. Чампі, Jennings, N.R., Wooldridge M J., Швецова О.М., та ін.

Існуючі методи і інструменти розглянуті в роботах зазначених дослідників не забезпечують єдину модель інформаційної системи та не дозволяють комплексно вирішувати питання формалізації та інформатизації процесів прийняття рішень. Також недостатньо розроблені питання використання методів та моделей на етапах автоматизації та інформатизації, що обмежені суттєвими витратами на розробку імітаційної моделі і відсутністю можливостей використовувати отримані результати і знання на етапах автоматизації.

Недосконало досліджені питання прогнозування достовірності отриманої інформації від мультиагентних систем з подальшим використанням цієї інформації при формуванні рішень. Всі ці фактори підтверджують актуальність розробки методів ППР при розробці інформаційних систем на основі мультиагентного підходу.

На підставі проведеного аналізу виявлено існування протиріч між вимогами до розроблюваного методу підтримки прийняття рішень: між вимогою до підвищення ефективності інформаційної системи з використанням мультиагентного підходу, що потребує додаткових витрат як часу так і коштів, та вимогою на зменшення витрат на розробку ефективного математичного та програмного забезпечення інформаційної системи. Ці обставини не дають можливості до збільшення ефективності процесу прийняття рішення при розробці інформаційних систем. Дане протиріччя підтверджує актуальність поставленого наукового завдання.

Метою дисертації є скорочення часу та зниження витрат на розробку і

впровадження інформаційних систем за рахунок реалізації розробленого методу підтримки прийняття рішень для завдання автоматизації процесів організаційно-технічної системи при розробці інформаційних систем.

Для реалізації мети дослідження в роботі необхідно вирішити такі завдання:

1. Проаналізувати існуючі моделі формалізації процесів організаційно-технічних систем з урахуванням динаміки процесів і людського фактору, з метою вибору адекватної моделі її подальшого використання при розробці методу.

2. Проаналізувати існуючі методи і засоби створення інформаційної системи з метою оцінки рівня розвитку цього напрямку.

3. Визначити особливості, які є специфічними для організаційно-технічної системи, та розробити концептуальну модель: визначити і описати основні об'єкти і взаємозв'язки між ними, з метою формалізації предметної області.

4. Удосконалити існуючий метод підтримки прийняття рішень для завдання розробки інформаційної системи з використанням інтеграції структурних, агентних, імітаційних та об'єктно-орієнтованих підходів.

5. Розробити метод трансформації моделі організаційно-технічної системи в модель інформаційної системи, що сформована як сукупність діаграм функціонального та об'єктно-орієнтованого підходу на основі використання концептуальних моделей предметної області.

6. Провести імітаційне моделювання для аналізу роботи окремих елементів інформаційної системи, оцінити ефективність роботи запропонованого методу проектування програмного забезпечення та провести оцінку продуктивності інформаційної системи.

7. Розробити програмні модулі та провести дослідження нового програмного середовища для аналізу процесів та розробки інформаційної системи, а також провести випробування розроблених положень для підтвердження їх достовірності.

Розв'язання даних наукових завдань можливо шляхом застосування розроблених у дисертації підходів і методів систем підтримки прийняття рішень у сфері створення інформаційних систем.

Висновки до розділу 1.

Розділ містить аналіз існуючих наукових методів що до розробки інформаційних систем на основі мультиагентного підходу та аналіз етапів системного аналізу в розробці інформаційних систем. Основною проблемою, яка виникає при розвитку інформаційної системи, є складність розуміння відразу всієї системи в цілому. Проведене дослідження організаційно-технічних систем показало, що автоматизація кожної групи процесів, що відбуваються в організаційно-технічних системах, має свої особливості. Існуючі підходи до розробки, кожен окремо, не охоплюють всі питання, які виникають в даному випадку. Успішне вирішення проблеми створення інформаційних систем неможливе без чіткої картини бізнес-процесів, що відбувається на автоматизованому підприємстві. Тому, перш за все, необхідно побудувати бізнес-модель, яка відображає автоматизовані процеси і необхідні для них ресурси. Також розглядаються етапи розвитку інформаційної системи, що передують написанню технічного завдання.

Мультиагентні системи являють собою новий напрямок у розвитку штучного інтелекту, який сформувався на основі результатів досліджень в області розподілених комп'ютерних систем, мережевих технологій для вирішення проблем і паралельних обчислень.

Важливою проблемою в розвитку інформаційних систем є отримання знань фахівців в певній предметній області. Здебільшого ці знання не формалізовані, а тому недоступні іншим людям. Розглянуті системи класу BPM підходять для розробки вузького класу систем, та кожні з них, з огляду на поставлену задачу мають ряд недоліків то й пропонується розробити метод підтримки прийняття рішень при розробці інформаційних систем і CASE-інструмент, який усунув би ці недоліки.

Розв'язання даних наукових завдань можливо шляхом застосування розроблених у дисертації підходів і методів систем підтримки прийняття рішень у сфері створення інформаційних систем.

РОЗДІЛ 2 РОЗРОБКА НАУКОВО–МЕТОДИЧНОГО АПАРАТУ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ РОЗРОБЦІ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ МУЛЬТИАГЕНТНОГО ПІДХОДУ

У другому розділі буде проведено дослідження та обрано моделі, представлення знань, досліджено області МППР, описано КМППР, розроблено модель та метод у області створення ІС та розроблено метод трансформації моделі організаційно–технічної системи в модель інформаційної системи.

2.1. Дослідження вимог до моделі та методу підтримки прийняття рішень при розробці інформаційних систем

Розглянемо розробку ІС предметної області мультиагентних процесів перетворення ресурсів (МППР). На основі аналізу методологічних та теоретичних основ підтримки прийняття рішень, моделювання та розвитку інструментів ІС та інструментів CASE ми сформулюємо вимоги до методу підтримки прийняття рішень (МППР) у сфері розробки ІС.

Метод ППР дослідження при розробці ІС повинен передбачати автоматизацію процесу створення ІС для предметної області організаційно–технічних систем. Таким чином, можна виділити наступні вимоги:

- 1) вибір методології системного аналізу та моделі формалізації ОТС–процесів. При цьому ми врахуємо наявність ОПР, які можуть бути представлені у вигляді ІА.
- 2) ІМ для перевірки моделі «як вона буде» на етапі реінжинірингу БП, а також для оцінки продуктивності ІС.
- 3) інтелектуальний розвиток ІС, включаючи функціональний і об'єктно–орієнтований аналіз, ІС–моделювання, формування виконуваного ІС–коду.

Розглянемо принципи побудови інтелектуального СППР в області розробки програмного забезпечення (CASE–інструменти). Структура такої системи включає в себе діалог ЕС, що складається з бази знань (БЗ), бази даних, механізму логічного висновку (МЛВ), блоку пояснення отриманих рішень, навчального блоку

(адаптація ЕС до мінливої реальності), блоку розуміння, блоку обслуговування, поповнення і коригування БЗ.

У розділі 2.2 ми порівняємо різні моделі БП і виберемо одну для презентації ППР.

2.2. Вибір моделі подання бізнес–процесів

У роботі [59] проведено аналіз можливостей використання найбільш поширених математичних моделей дискретних процесів для представлення ППР. Показано, що моделі: мережі Петрі; розширені мережі Петрі; системи масового обслуговування (СМО); моделі системної динаміки не забезпечують всіх вимог до моделювання ППР. Математична модель ППР, представлена в роботі [59], була додатково розширена апаратом мультиагентних систем [61].

Динамічну модель МППР отримали Гончарова Н.В та Аксьонов К.А. Він складається з: процесів (*PR*), операцій (*Op*), ресурсів (*RES*), управлінських команд (*U*), інструментів (*MECH*), відправників (*Sender*) і приймачів (*Reciver*), перетинів (*Junction*), параметрів (*P*), агентів (*Agent*). Окремо виділяються інформаційні типи ресурсів: повідомлення (*Message*) і запити на роботу (*Order*). Опис причинно–наслідкових зв'язків між елементами перетворення і ресурсами задано об'єктом (*Relation*).

i–а операція (Op_i) представлена наступною структурою:

$$Op_i = \langle f, RESin_i, RESout_i, MECH_i \rangle \quad (2.1)$$

де f – функція, яку реалізує *i*–а операція;

$RESin_i$ – вхідні ресурси для виконання *i*–ої операції;

$RESout_i$ – вихідні ресурси для виконання *i*–ої операції, $RESout_i = f(RESin_i)$;

$MECH_i$ – механізми для виконання *i*–ої операції;

$RESin_i = \{RESin_{i1}, RESin_{i2}, \dots, RESin_{ik}\}$ – множина вхідних ресурсів

$RESout_i = \{RESout_{i1}, RESout_{i2}, \dots, RESout_{im}\}$ – множина вихідних ресурсів

$Sender_{ik}$ – джерело ik –ресурсу;

$Receiver_{ik}$ – приймачі ik –ресурсу.

Семантика процесу перетворення ресурсу показано рис. 2.1.

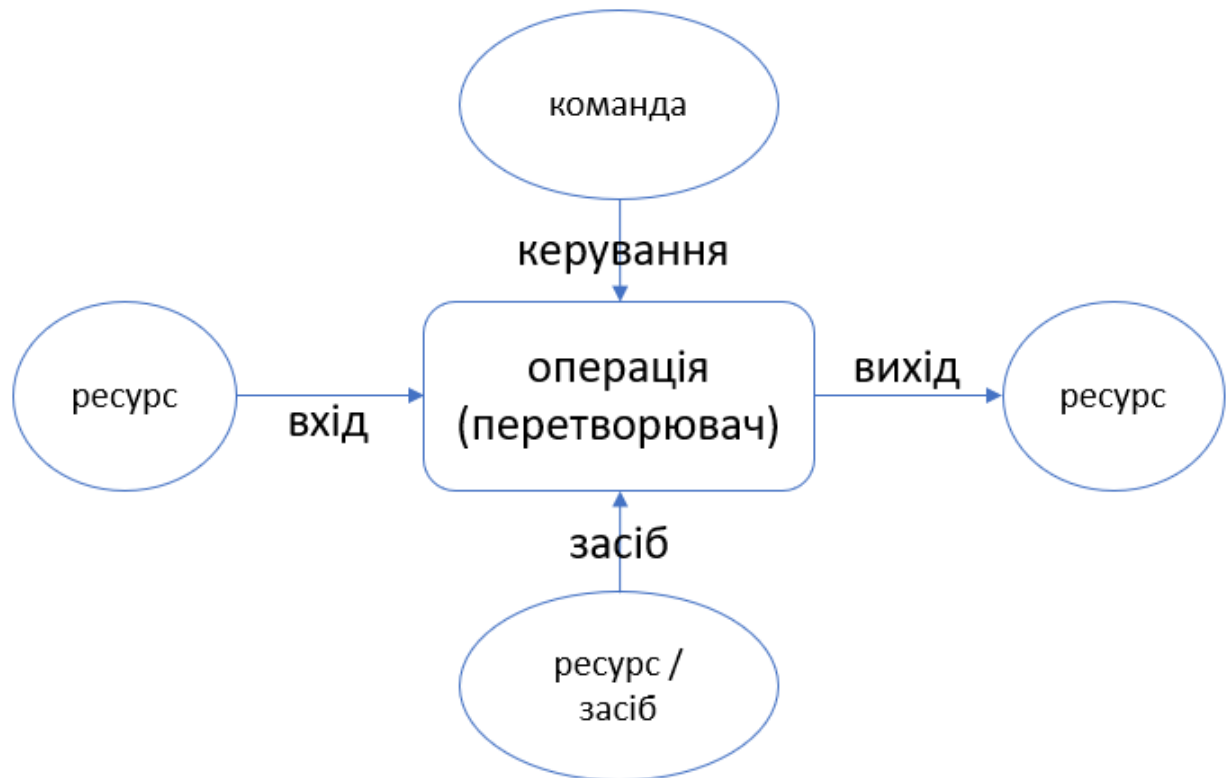


Рис. 2.1. Семантика процесу перетворення ресурсів

Результати порівняння представлені у вигляді табл. 2.1.

Таблиця 2.1

Результати порівняння дискретних математичних моделей

Характеристики	Мережі Петри	розширені мережі Петрі	СМО	моделі системної динаміки	модель МППР
Врахування тимчасових характеристик	НІ	ТАК	ТАК	ТАК	ТАК
Можливість обліку різних типів ресурсів	НІ	ТАК	ТАК	НІ	ТАК
Моделювання конфліктів на загальних засобах	НІ	НІ	ТАК	НІ	ТАК
Модель ОПР (ІА)	НІ	НІ	НІ	НІ	ТАК

Отриману математичну модель МППР пропонується використовувати в якості моделі опису БП. Перевагою моделі МППР є можливість використання моделей ОПР.

Одним із завдань при розробці CASE–інструментів є проблема вибору моделі відображення знань (для реалізації семантики переходу від об'єктів автоматизованого процесу до об'єктів предметної області ІС). Це завдання вирішується в розділі 2.4.

2.3. Вибір моделі подання знань

Розглянемо існуючі способи представлення знань [63, 64]. При аналізі будуть враховані наступні вимоги.

1. Найпростіший і найприродніший перехід від неформальних знань і уявлень до формальних моделей опису БД і ІС, видимості подачі інформації користувачеві.
2. Можна представити ієрархічні дані, так як предметні області МППР і ІС утворюють ієрархію.
3. Легко додавати нові знання.
4. Технічна реалізація обраної моделі повинна бути досить простою і відповідати об'єктно–орієнтованому підходу (ООП) розробки програмного забезпечення.
5. Можливість використання мови UML як мови представлення візуальних знань (об'єктно–орієнтований аналіз).

1 Метод – представлення знань у вигляді продуктів або правил і їх перекладача, який визначає, коли і яке правило застосовується.

Модель виробництва визначається наступним чином:

$$P_i, \dots, P_m \rightarrow Q_i, \dots, Q_n \quad (2.2)$$

де P_i, \dots, P_m , – приміщення, Q_i, \dots, Q_n , – дії, виконані у випадку істинності приміщення.

Принцип роботи виробничої системи наступний [38]: продукт (правило) виконується, умова якого буде вірна для поточного стану БЗ і БД. При цьому це правило активує дані, розташовані в зазначеній структурі бази даних. Правила виконуються до тих пір, поки всі вони не будуть виконані або правило зупинки не вступить в силу.

Додати нові знання до серійної моделі досить легко, так як будь-який продукт можна розмістити в будь-якому місці моделі. Механізм висновку в продуктах добре поєднується з процедурним підходом в програмуванні.

До недоліків можна віднести наступне: незручно реалізовувати ієрархічну структуру, непомітне представлення знань і неефективний процес висновку, так як в цілому необхідно перевірити застосовність всіх правил.

2 Метод – представлення знань у вигляді семантичної мережі. Семантична мережа – це орієнтована графова структура, кожна вершина якої являє собою певне поняття (об'єкт, процес, ситуація), а краї графа відповідають відношенням типу "це", "належати", "бути причиною", "увійти", "складатися", "бути як" і подібними між парами понять [64].

Семантична модель визначається наступним чином

$$S = (O, R_1, R_2, \dots, R_n) \quad (2.3)$$

де O – множина ІС-об'єктів;

$R_i | i = 1, n$ – множина зв'язків між об'єктами;

l – тип відносин.

Можна вказати наступні переваги цього методу: видимість представлення знань користувачам, семантична модель добре поєднується з ієрархічними знаннями. До недоліків можна віднести складність реалізації механізму виведення.

3 Метод – Представлення знань у вигляді фреймів. Мінський у своїй роботі визначив фрейм як «структуру даних для представлення стереотипних ситуацій

[64].

Фрейм складається з слотів, які є різними характеристиками об'єкта (атрибуції), і наповнювачів (значення цих атрибутів і процедури, які виконуються при зміні даних фрейму). Кожен фрейм має спеціальний слот, який зберігає назву сутності, яку він представляє. Модель фрейму може бути представлена наступним зображенням:

$$\Phi = \langle I\Phi, T\Phi, CСЛ \rangle \quad (2.4)$$

$$CСЛ = \{CAтр\} \quad (2.5)$$

$$CAтр = \{(IAтр_1, MB_1, ZA_1), (IAтр_2, MB_2, ZA_2), \dots (IAтр_m, MB_m, ZA_m)\} \quad (2.6)$$

де $I\Phi$ – назва фрейму, $T\Phi$ – тип фрейму, $CСЛ$ – структура слотів, $CAтр$ – структура атрибута, $IAтр_m$, – це ім'я атрибута, MB_m – множина визначень, ZA_m , – це значення атрибута.

Крім того, фрейм можна розглядати як вузол у певній мережі. Зв'язки можуть бути таких типів: приклад–клас і клас–суперклас.

Фреймове представлення знань досить візуальне, підходить для представлення ієрархічних знань, добре поєднується з ООП, тому перехід від представлення знань до програмної реалізації ІС здається досить простим. До недоліків фреймової моделі можна віднести складність внесення змін до ієрархічної структури даних.

Поєднання семантичних мереж і фреймів мінімізує недоліки цих двох методів при збереженні достоїнств.

4 Метод – Застосування фреймово–семантичної моделі представлення знань. Швецов А.Н [66]. пропонується об'єднати фреймово–подібні структури зі структурами концептуальних графів J.F. Sowa [67–69]. Основна конструкція концепції фрейму (ФК) представлена формулою 2.7. Назва фрейму – унікальний ідентифікатор, що дозволяє однозначно ідентифікувати фрейм. опис можливих ситуацій використання ФК, сценаріїв поведінки, функцій вибору і т.д., в довільній формі. Динамічна поведінка компонентів області або агентів описує структуру

сценарію поведінки (ССП), яка включає в себе блок вибору сценарію (БВСЦ), що дозволяє формувати альтернативні шляхи поведінки для даного фрейму.

ССЛ складається з двох структур: структури концептів (СК) і СА. СК містить список концепцій фрейму, в деяких аспектах вкладених або породжених всеосяжним ФК, верхівка цього відношення вказується в полі "тип концептуального відношення", тобто відношення цього ІК; до ФК, де ІК є назвою і-го поняття. Для встановлення логічної організації предметної області ФК об'єднуються в структури концептуальних графів. Концептуальний граф (КГ) – це дводольний граф, який має два типи вершин: концептуальні вершини, і вершини концептуальних відносин (КВ). Таким чином, Швецов пропонує використовувати фреймово–семантичне представлення знань, яке виглядає так :

$$\text{ФК} = \langle \text{ІФ}, \text{ТФ}, \text{ІВ}, \text{ССП}, \text{ССЛ} \rangle \quad (2.7)$$

$$\text{ССЛ} = \{\text{СК}, \text{САтр}\} \quad (2.8)$$

$$\text{СК} = \{(\text{ІК}_1, \text{КВ}_1), (\text{ІК}_2, \text{КВ}_2), \dots (\text{ІК}_m, \text{КВ}_m)\} \quad (2.9)$$

$$\text{САтр} = \{(\text{ІАтр}_1, \text{МВ}_1, \text{ЗА}_1), (\text{ІАтр}_2, \text{МВ}_2, \text{ЗА}_2), \dots (\text{ІАтр}_m, \text{МВ}_m, \text{ЗА}_m)\} \quad (2.10)$$

де ІФ ім'я фрейму, ТФ – тип фрейму, ІВ – інформація про використання, ССП – структура сценарію поведінки, ССЛ – структура слотів, СК – структура концептів, САтр – це структура атрибутів, ІК₁ – назва поняття, КВ₁, – концептуальне відношення, ІАтр₁ – ім'я атрибута, МВ₁ – множина визначень, ЗА₁ – значення атрибута.

Переваги фреймово–семантичного підходу: ефективно реалізує ієрархічне представлення даних, добре поєднується з ООП, в роботі [61, 62] вирішується технічна реалізація цієї моделі представлення знань на рівні референтної бази даних.

Порівняльний аналіз розглянутих моделей представлення знань представлений в табл. 2.2.

Порівняння різних моделей представлення знань.

Вимоги до моделі	Продукції	Сем.мережі	Фрейми	ФК та КГ Швецова
Наочність	НІ	ТАК	ТАК	ТАК
Подання ієрархічних даних	НІ	ТАК	ТАК	ТАК
Простота додавання нових знань	ТАК	НІ	НІ	ТАК
Узгодженість із ООП	НІ	НІ	ТАК	ТАК
Використання UML	НІ	НІ	ТАК	ТАК
Опис ІА	НІ	НІ	НІ	ТАК

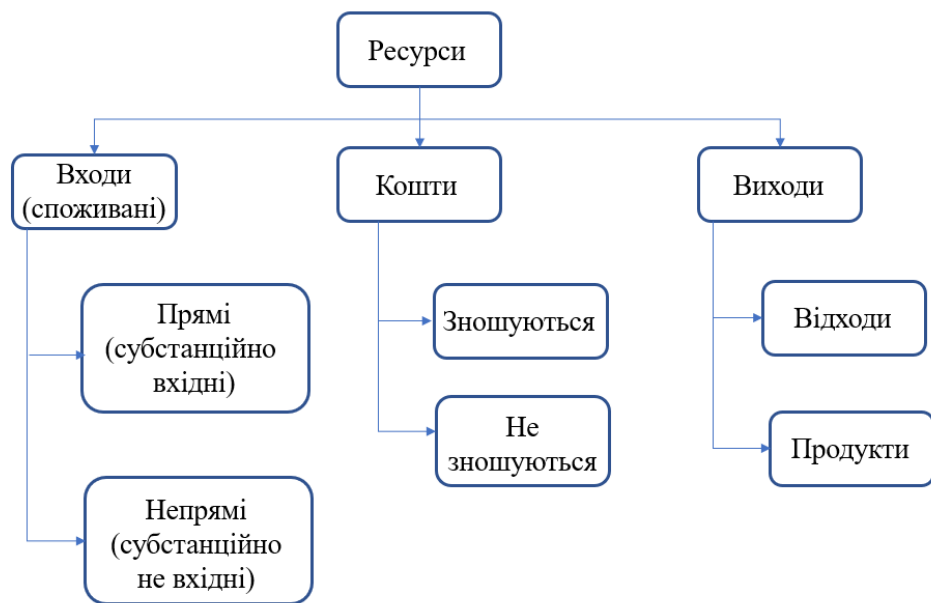
2.4. Розробка концептуальної моделі предметної області організаційно-технічної системи на основі фреймово семантичної моделі представлення знань

2.4.1. Удосконалення методу підтримки прийняття рішень для завдання розробки інформаційної системи

На основі моделі БП, обраної в розділі 2.3, може бути побудована наступна фреймово–семантична модель МППР (рис. 2.2) [70, 72].

При автоматизації процесів роботи будь–який перетворювач відповідає функції обробки даних, однонаправленої або двонаправленої (генерація, прийом, передача, модифікація, видалення).

У роботі Піщулова [74] запропоновано класифікацію ресурсів з точки зору використання. На рис. 2.3 представлено цю класифікацію [81].



1) аналізує зовнішні параметри (поточна ситуація);

- 2) діагностує ситуацію, відноситься до бази знань. При виявленні відповідної ситуації агент намагається знайти рішення (сценарій дії) в базі знань або розробити його самостійно;
- 3) розробляє (приймає) рішення;
- 4) визначає (перевизначає) цілі;
- 5) контролює досягнення цілей;
- 6) делегувати цілі своїм та чужим об'єктам процесу перетворення ресурсів, а також іншим агентам;
- 7) Обмін повідомленнями.

Архітектура агента МППР заснована на архітектурі Hybrid InteRRaP і показана на рис. 2.4.

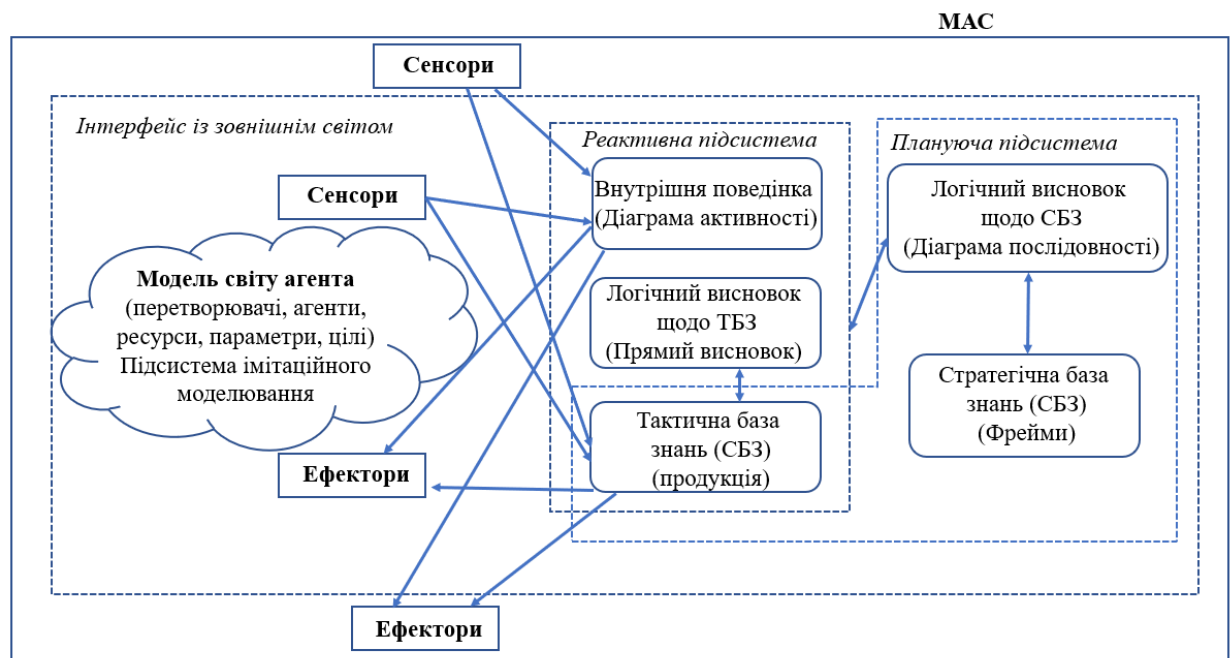


Рис. 2.4. Архітектура агента МППР

Модель ОТС, представлена як моделі МППР, показано рис. 2.5

Визначимо властивості та методи фрейм–концептів (ФК) предметної області МППР. ФК перераховуються в міру їх ускладнення і за принципом «від загального до приватного». У дужках у ФК вказується назва батьківського ФК, якщо така є.

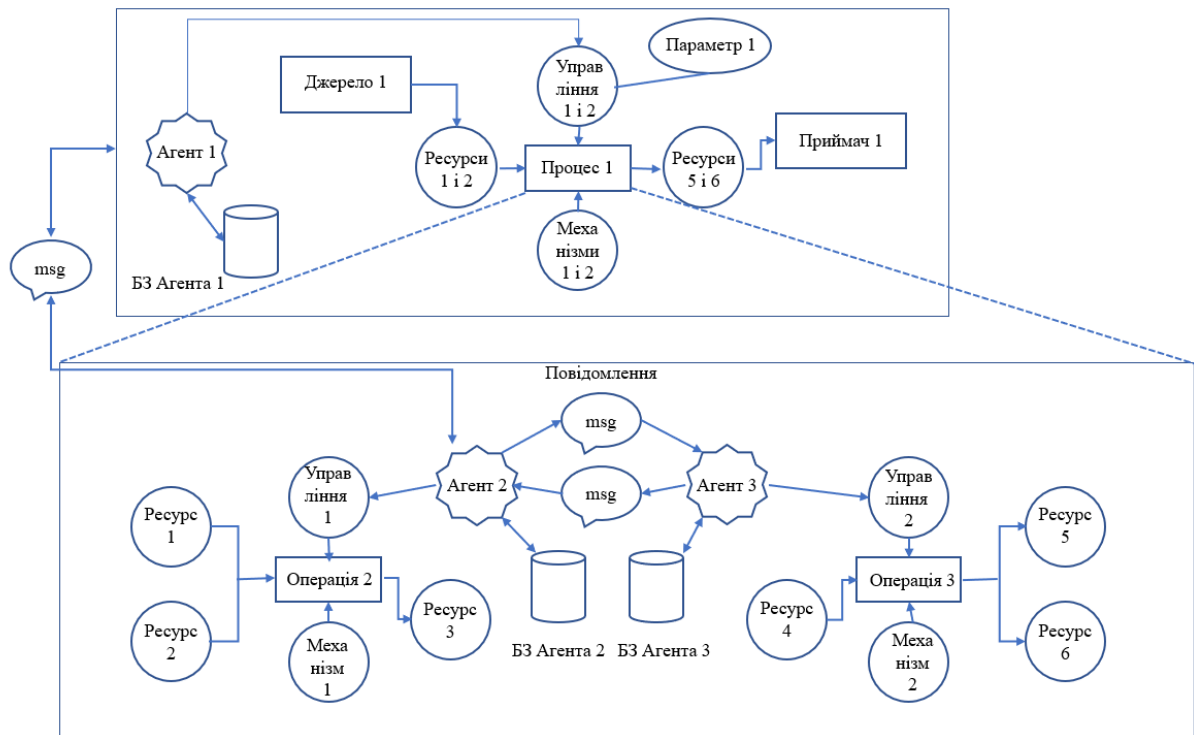


Рис. 2.5. Модель ОТС

ФК “Ресурс”

Властивості:

- 1) ідентифікатор (ID_RES),
- 2) ім'я ресурсу (RES_Name),
- 3) тип ресурсу ($kind$),
- 4) поточне значення ресурсу (RES_i),
- 5) максимально можливе значення ресурсу (RES_{max}),
- 6) початкове значення ресурсу (RES_0),
- 7) Цільове значення ресурсу (RES_T),
- 8) ціна за одиницю ($Cost$),
- 9) час початку моделювання (t_0),
- 10) Ресурсна одиниця ($Metric_R$).

Методи:

- 1) загальний приріст ресурсів за часовий інтервал ($Hsum_{inc_t}$),
- 2) загальне зменшення ресурсу за часовий інтервал ($Hsum_{dec_t}$),

- 3) вартість ресурсу ($Cost\ sum$),
- 4) збільшення ресурсу в поточний час (H_{inc}),
- 5) Зменшення ресурсу в поточному часі (H_{dec}).

ФК “Команда” (ресурс)

Властивості:

- 1) ідентифікатор (SD_Cmd),
- 2) ім'я ресурсу (Cmd_Name),
- 3) тип команди ($kind$),
- 4) відправник ($Msender$),
- 5) одержувач ($Mreceiver$),
- 6) текст повідомлення ($text$),
- 7) пріоритет ($prior$),
- 8) функція зчитування ($read$),
- 9) час створення (T_{create}),
- 10) Час очікування черги (T_{wait}).

ФК "Заявка" (Ресурс)

Властивості:

- 1) ідентифікатор (ID_Order),
- 2) ім'я ресурсу ($Order_Name$),
- 3) замовлений обсяг робіт ($count$),
- 4) обсяг виконаних робіт ($real$),
- 5) ознака блокування програми ($lock$),
- 6) назва предмета, що обробляє претензію ($Owner$),
- 7) ім'я блоку, який створив запит ($Parent$),
- 8) пріоритет заявки ($prior$),
- 9) час створення заявки (T_{create}),
- 10) Час очікування квитка в черзі (T_{wait}).

ФК "Засоби" (Ресурс, Операція)

Властивості:

- 1) ідентифікатор (ID_Mech),
- 2) ім'я ресурсу ($Mech_Name$),
- 3) тип команди ($kind$),
- 4) поточна сума вільних коштів ($Mech_i$),
- 5) загальна сума коштів ($Mech_{all}$),
- 6) час створення (T_{create}),
- 7) стан інструменту ($Status$),
- 8) одноразові накладні витрати ресурсу на початку перетворення (RES_{in}),
- 9) одноразові накладні витрати ресурсу в кінці перетворення (RES_{out}),
- 10) витрата ресурсів за одиницю часу (RES_{use}),
- 11) одноразові накладні витрати ресурсів, коли інструмент захоплюється іншою операцією (RES_{lock}),
- 12) одноразові накладні витрати на ресурс при випуску (RES_{unlock}),
- 13) витрати на ресурси в разі і усунення поломки (RES_{other}),
- 14) частота виникнення поломки (T_{other}),
- 15) початкова ціна одиниці товару ($Cost$),
- 16) загальний час використання інструменту (T_{mech_usse}),
- 17) загальний час простою виробу (T_{mech_stand}).

Методи:

- 1) дія для запуску засобу, коли почнеться перетворення (Am_{in}),
- 2) дії, щоб зупинити інструмент після завершення перетворення (Am_{out}),
- 3) дія перетворення (Am_{use}),
- 4) дії, щоб зупинити інструмент під час переривання перетворення (Am_{lock}),
- 5) дії, щоб запустити інструмент, коли перетворення триває (Am_{unlock}),
- 6) дії для усунення поломки (Am_{other}),
- 7) продуктивність інструменту за одиницю часу ($product$).

Операція є найпоширенішим елементом МППР, решта (джерела, приймачі, перетини) містять усічений набір властивостей і методів, тому ми наведемо тільки

описи ФК «Операція».

ФК "Операція" (Перетворювач)

Властивості:

- 1) множина входів (IN),
- 2) множина виходів (OUT),
- 3) множина ресурсів, необхідних для переривання операції (RES_{lock}),
- 4) множина ресурсів, необхідних для продовження виконання операції, що залишилася в результаті переривання (RES_{unlock}),
- 5) операційні цілі (GOp),
- 6) засоби трансформації ($MECH$),
- 7) стан операції ($Status$),
- 8) тривалість перетворення ($time$),
- 9) пріоритет операції ($prior$),
- 10) тип пріоритету ($kind_prior$),
- 11) ознака заборони переривання ($break_off$).

Методи:

- 1) функція, що реалізується операцією (f);
- 2) $C_a^{message}$ – умова наявності необхідних вхідних повідомлень;
- 3) C_a^{order} – умова наявності необхідних вхідних додатків;
- 4) C_a^m – умова наявності необхідних вхідних ресурсів;
- 5) C_a^{out} – умова врахування меж виходу;
- 6) C_a^{mech} – умова готовності необхідних засобів;
- 7) C_a^{status} – стан готовності до виконання;
- 8) C_a^{time} – це умова запуску на основі часу.
- 9) $Action_{in}^{message}$ – захоплення вхідного повідомлення;
- 10) $Action_{in}^{order}$ – захоплення вхідних запитів;
- 11) $Action_{in}^{MECH}$ – захоплення вхідних ресурсів;

- 12) Акція – за захоплення коштів;
- 13) $Action_{out}^{Message}$ – формування вихідних повідомлень;
- 14) $Action_{out}^{Order}$ – формування вихідних запитів;
- 15) $Action_{out}^{RES}$ – формування вихідних ресурсів;
- 16) $Action_{out}^{MECH}$ – вивільнення вилучених коштів.

ФК “Агент”

Властивості:

- 17) ідентичність (ID_Agent),
- 18) ім'я агента ($Agent_Name$),
- 19) агентські цілі ($GAgent$),
- 20) пріоритет агента ($prior$),
- 21) База знань агента (KB_{Ag}),
- 22) кількість вхідних повідомлень ($Messin_count$),
- 23) кількість вихідних повідомлень ($Messout_count$),
- 24) сценарій поведінки (SPA),
- 25) множина керованих об'єктів МППР ($Control_objects$)
- 26) множина агентів "начальства" (AU),
- 27) множина підпорядкованих агентів (AD).

Методи:

- 1) аналіз світу ($Analyze$),
- 2) діагностика ситуацій ($Diagnost$),
- 3) пошук рішення ($Search$),
- 4) обробка цілей,
- 5) контроль за досягненням цілей,
- 6) делегування цілей,
- 7) обмін повідомленнями,
- 8) взаємодія з БЗ і БД.

ФК “Параметр”

Властивості:

- 1) ідентифікатор (ID_Param),
- 2) ім'я параметра ($Param_Name$),
- 3) поточне значення параметра (P_t),
- 4) початкове значення параметра (P_0),
- 5) опис параметра ($desc$), запланованого значення ($plan$).

Методи:

- 1) функція розрахунку параметра (f).

Таким чином, цей підрозділ описує головний ФК КМПО предметної області МППР. Наступний розділ вирішує проблему побудови КМПО проектування ІС.

2.4.2. Розробка методу трансформації моделі організаційно–технічної системи в модель інформаційної системи

Запропонована КМПО інформаційної системи дає можливість показати структуру інформаційної системи і взаємозв'язки між усіма його компонентами (див. рис. 2.6) [75, 76]. На першому рівні фреймово–семантичної мережі існують вузли, що відповідають програмному забезпеченню і БД інформаційної системи. Оскільки практичне моделювання та розробка інформаційної системи використовують функціональні та об'єктно–орієнтовані підходи, він також містить елементи ФК моделювання архітектури програмного забезпечення, включаючи ФК наступних діаграм: функціональні (IDEF0), потоки даних (DFD) (рис. 2.7) та діаграми UML (прецеденти, послідовності та класи, рисунки 2.8–2.10) [72, 77]. Ці стандарти не містять опису конкретних реалізацій операцій, однак відповідні ФК містять методи, подібні до методів ФК «Операція». Це дозволяє зберігати цю інформацію при переході від моделі МППР до моделі інформаційної системи, а потім використовувати її при складанні програмних модулів.

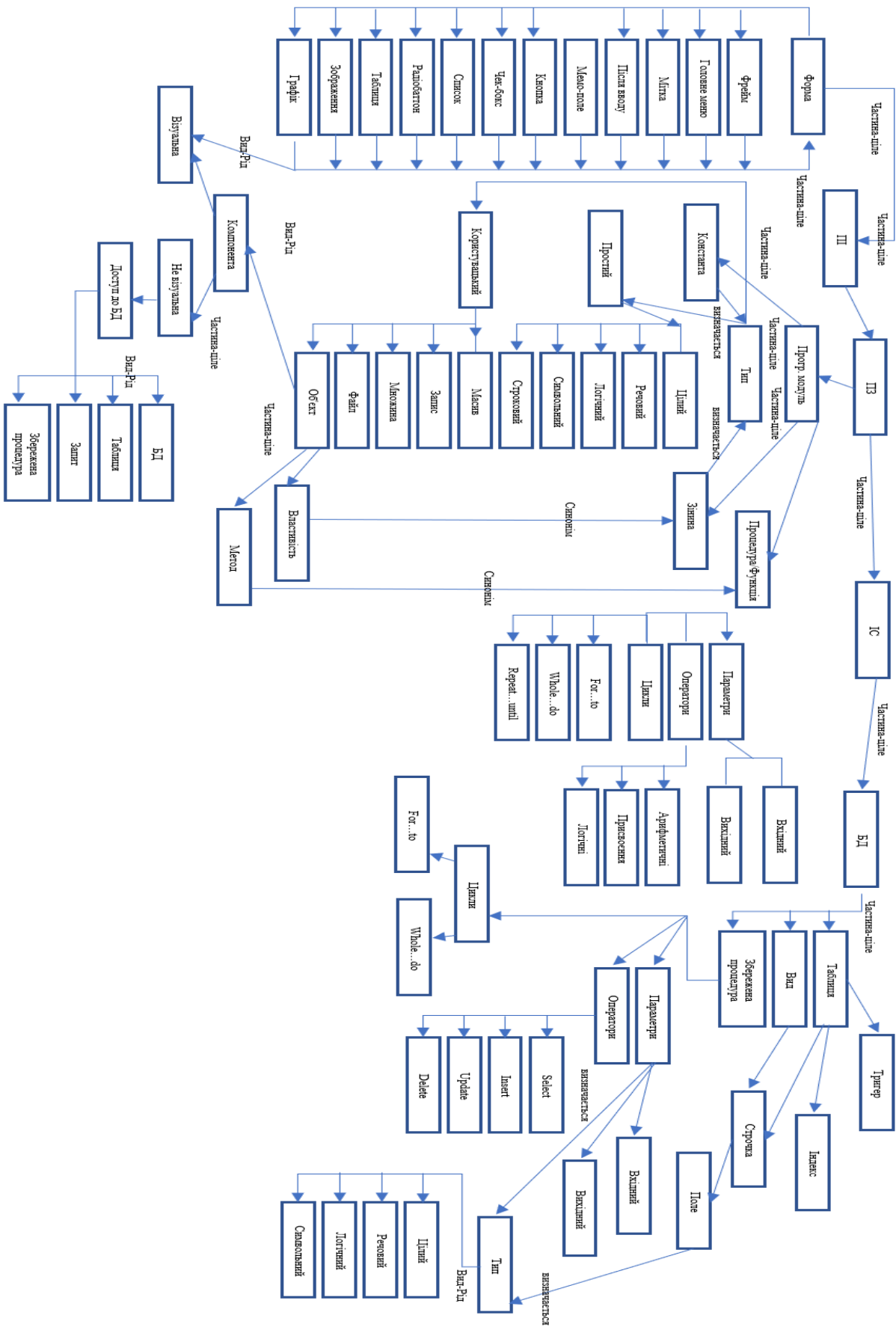


Рис. 2.6. Концептуальна модель предметної області інформаційної системи

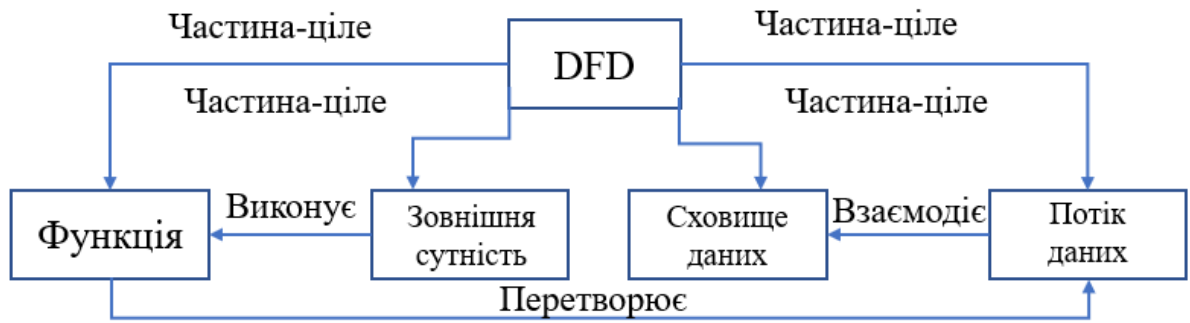


Рис. 2.7. Семантика діаграми DFD

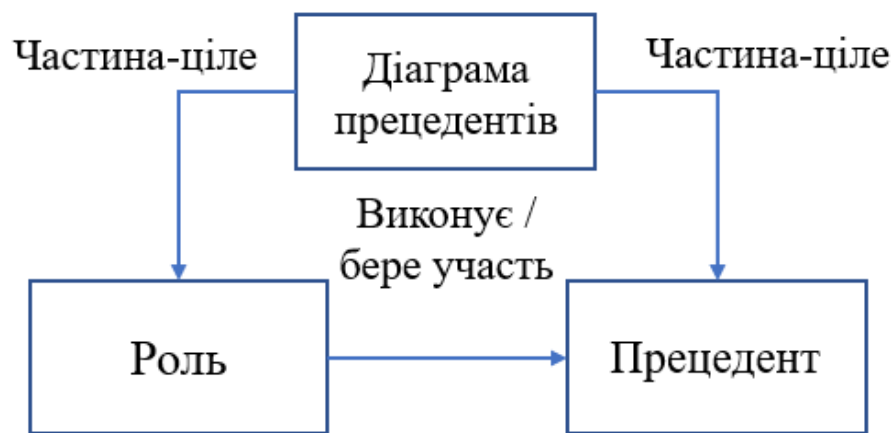


Рис. 2.8. Семантика діаграм прецедентів

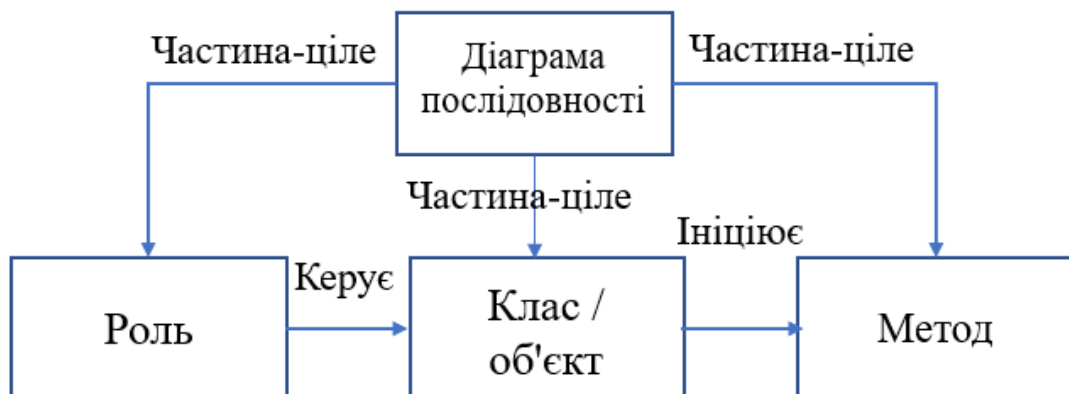


Рис. 2.9. Семантика діаграм послідовностей

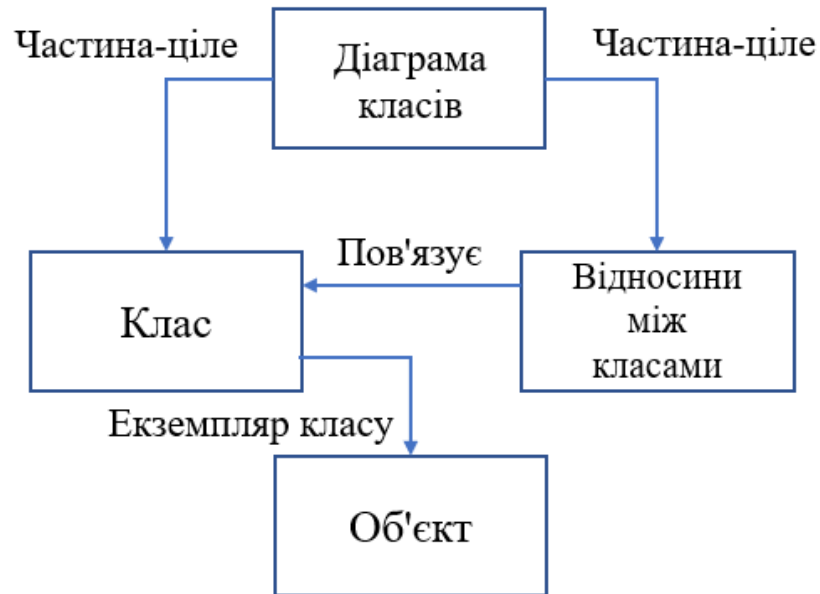


Рис. 2.10. Семантика діаграм класів

Визначимо структуру каркасних понять.

1. Концепції фрейму діаграми DFD

1.1 ФК "Функція"

Властивості:

- 1) ідентифікатор (ID_F);
- 2) Назва (F_Name)
- 3) множина входів (IN);
- 4) множина виходів (OUT).

Методи:

- 1) Функція перетворення (f).

1.2 ФК "Зовнішня сутність"

Властивості:

- 1) ідентифікатор (ID_Ex),
- 2) ім'я зовнішньої сутності (Ex_Name),
- 3) цілі зовнішньої сутності (ExG)
- 4) пріоритет зовнішньої сутності ($Exprior$).

1.3 ФК "Сховище даних"

Властивості:

- 1) ідентифікатор (ID_DS),
- 2) назва сховища даних (DS_Name),
- 3) Структура сховища даних ($DS_Structure$).

1.4 ФК "Потік даних"

Властивості:

- 1) ідентифікатор (ID_DF),
- 2) назва потоку даних (DF_Name),
- 3) тип потоку даних $= \{Select, Insert, Delete, Update, Unknown\}$ (DF_kind),
- 4) поточне значення потоку даних (DFt),
- 5) максимально можливе значення потоку даних ($DFmax$),
- 6) початкове значення потоку даних ($DF0$),
- 7) остаточне значення потоку даних (DFT),
- 8) ціна за одиницю потоку даних (DF_Cost),
- 9) Одиниця вимірювання потоку даних (DF_Metric_R).

2. Рамкові концепції діаграми прецедентів

2.1 ФК "Роль"

Властивості:

- 1) ідентифікатор (ID_R),
- 2) ім'я ролі (R_Name),
- 3) цілі ролі (RG),
- 4) пріоритет ролі ($Rprior$).

2.2 ФК "Прецедент"

Властивості:

- 1) Ідентифікатор (ID_UC),
- 2) ім'я (UC_Name).

Методи:

- 1) функція перетворення (UCJ),
- 2) Умова запуску (UC_Ca).
3. *Фрейм–концепти діаграми класів*

3.1 ФК “Клас”

Властивості:

- 1) Ідентифікатор (*ID_class*),
- 2) ім'я класу (*Class_Name*),
- 3) інші властивості визначаються предметною областю

Методи:

Методи класу визначаються предметною областю.

3.2 ФК “Об'єкт”

Представляє екземпляр відповідного класу, його властивості і методи визначаються властивостями і методами відповідного класу.

3.3 ФК "Зв'язок між класами"

Можливі типи відносин [28]:

- асоціації – довільний взаємозв'язок;
- узагальнення – зв'язок між батьківським класом і класом нащадкомм;
- агрегація – зв'язок між класом–контейнером і класом–частиною;
- композиція – агрегація, при якій у випадку знищення класу–контейнера знищуються всі його класи–частини;
- залежність – взаємозв'язок між класами, у якому одному або кільком класам необхідні, інші класи для їх специфікації або реалізації.
- реалізація – це взаємозв'язок між класами, при котрій один клас представляє якусь специфікацію, а інший представляє її реалізацію.

Властивості:

- 1) ідентифікатор (*ID_rl*),

- 2) ім'я (*Rl_Name*),
- 3) тип зв'язку (*Rl_type*),
- 4) множина початкових класів (*Beg_classes*),
- 5) множина скінченних класів (*End_classes*).

Запропонований метод ППР в області моделювання та розробки ІС повинен вирішити проблему переходу від КМПО МППР до КМПО ІС. Цей метод описаний в наступному розділі.

2.5. Дослідження методу розробки інформаційних систем

Розглянемо, як КМПО МППР трансформується в КМПО ІС (діаграми ФК DFD і діаграми ФК UML). Для опису семантики переходів між різними моделями БП і ІС використовується діаграма стану об'єкта (стандарт IDEF5) [70].

Ресурс (*RES*).

Ресурс в ІС може бути змінною, тобто.

$$RESin_i \rightarrow VAR_i., RESout_k \rightarrow VAR_k \quad (2.11)$$

де $RESin_i$ є i -им вхідним ресурсом,

VAR_i – i -ю змінною,

$RESout_k$ – вихідний ресурс – k -й вихідний ресурс.

Далі, рухаючись по семантичній мережі, виберіть вершини зі зв'язком Вид–рід і предком Простий тип або Користувацький тип. Тому можна визначити необхідний тип для зберігання ресурсу. У більш складних випадках ресурсом може бути складна структура: таблиця (на рівні бази даних) і/або масив (на рівні змінних коду). Ця семантика переходу ресурсів показана на рис. 2.11. Семантика переходу ресурсу в об'єкти діаграм показана на рис. 2.12

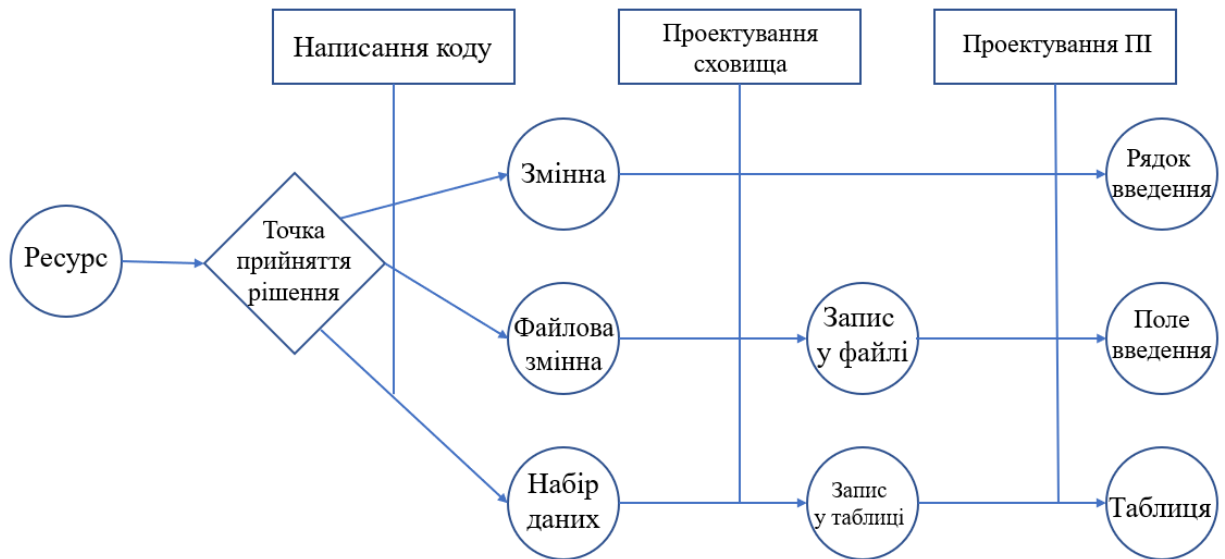


Рис. 2.11. Семантика переходу ресурсів в елементи ІС

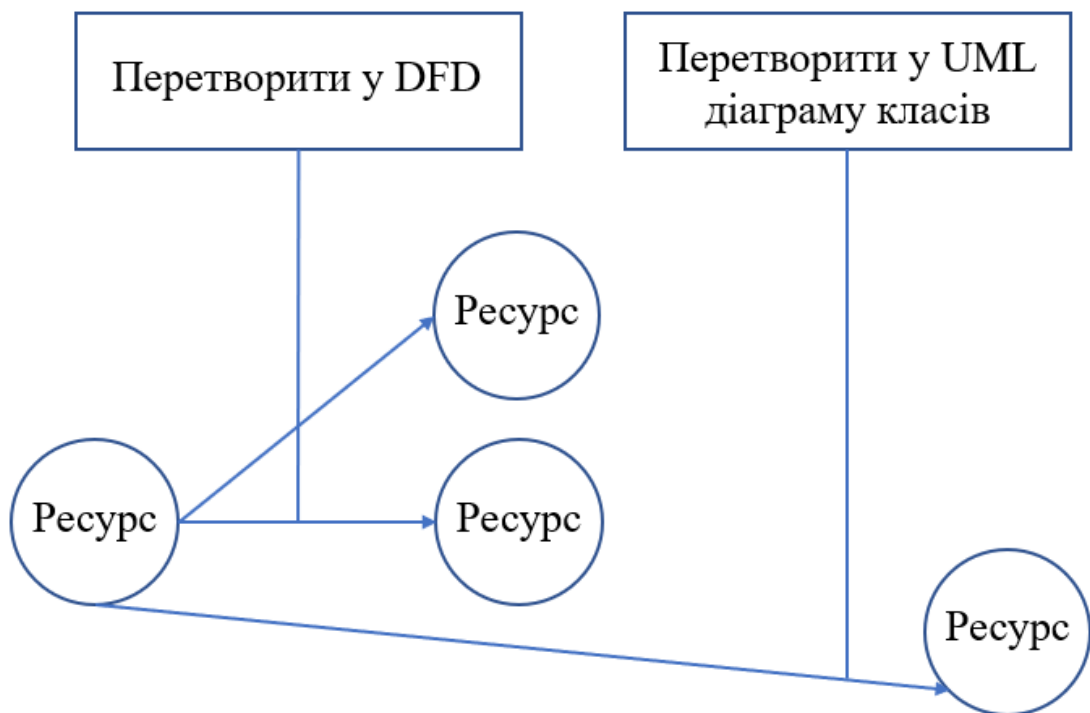


Рис. 2.12. Семантика переходу ресурсів до об'єктів діаграми

Сценарій переходу ресурсів можна представити на наступній схемі (рис. 2.13):

Перетворювач

Перетворювач включає в себе процеси, джерела, приймачі та перехрестя. Перетворювач як функція перетворення вхідних ресурсів на вивід може бути

реалізований як функція (ПЗ), яка обробляє дані в пам'яті або на рівні файлу, або збережена процедура (ЗП) бази даних (БД).

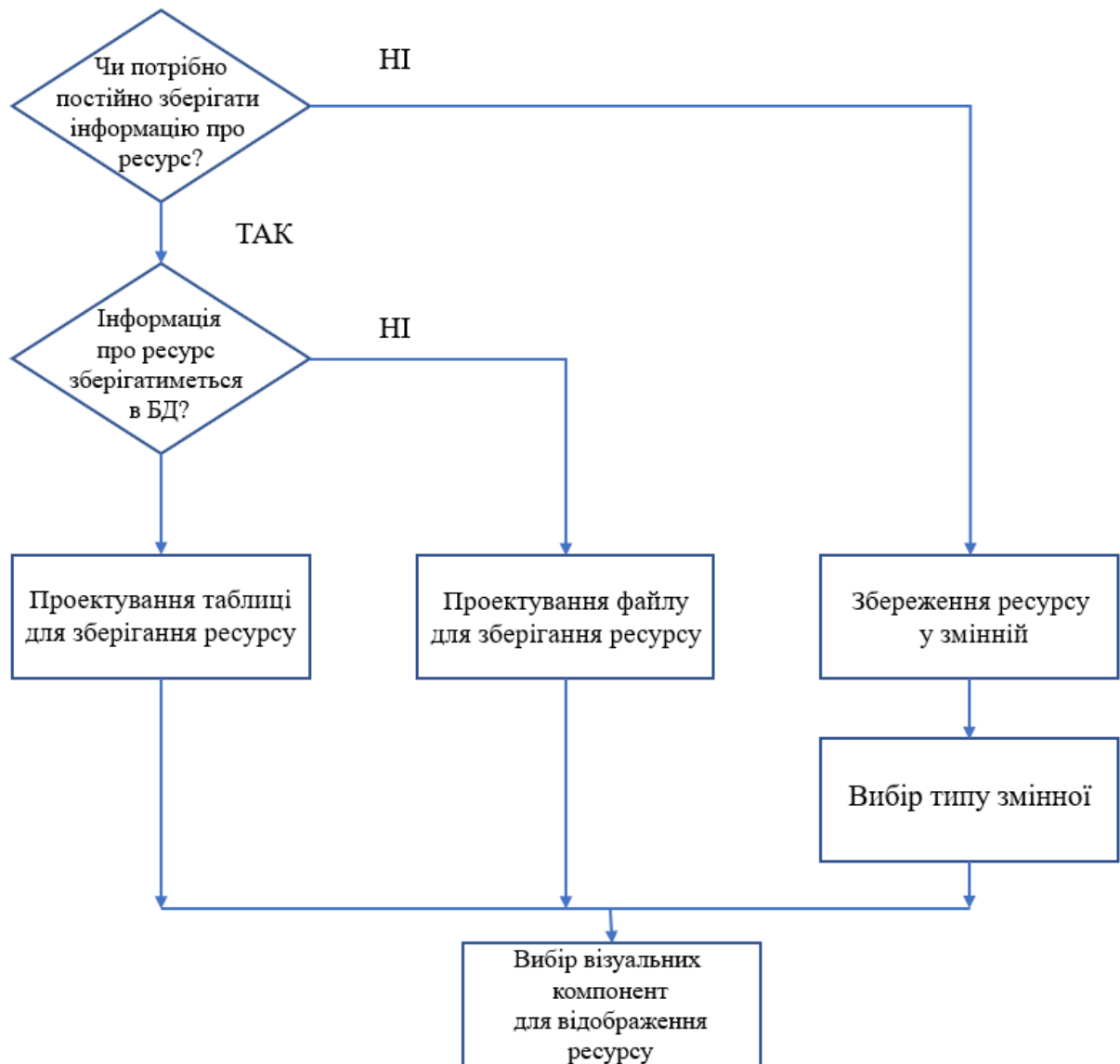


Рис. 2.13. Блок-схема передачі ресурсів

$$Tr_i \rightarrow Func_i, Tr_i \rightarrow StoredProc_i \quad (2.12)$$

де Tr_i – перетворювач;

$Func_i$ – функція ПЗ;

$StoredProc_i$ – збережена процедура.

При необхідності необхідно забезпечити існування елементів КІ для

визначення умов ініціювання конверсії. Вхідними параметрами для функції або ЗП будуть вхідні ресурси, а вихідними параметрами будуть вихідні ресурси. Таким чином, визначив ресурси на першому етапі, тим самим вказуються типи параметрів функцій або ЗП. В процесі інтелектуального моделювання і розробки програмного забезпечення можна надати можливість описати алгоритм виконання операції алгоритмічною мовою або T-SQL.

Семантика переходу перетворювача показана на рис. 2.14.

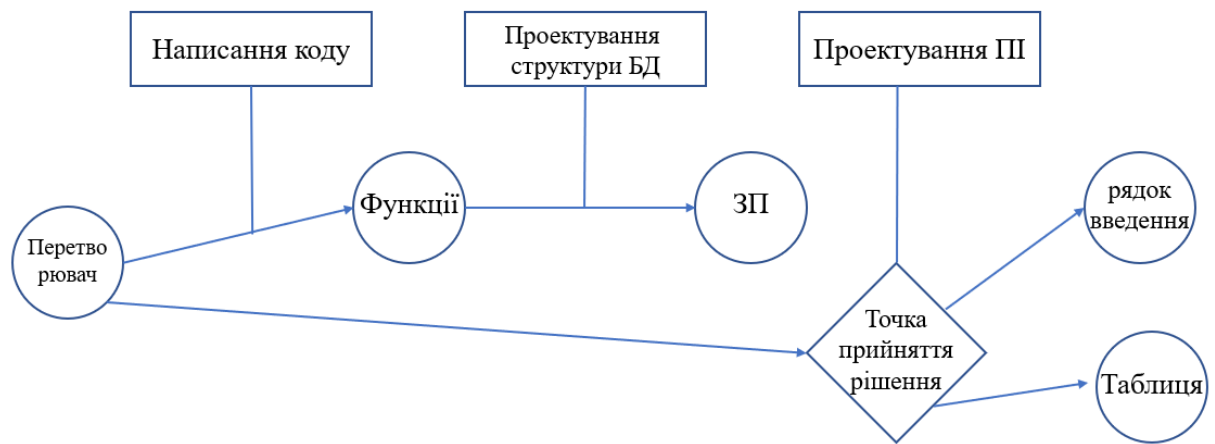


Рис. 2.14. Семантика переходу перетворювача

Семантика переходу ресурсу на діаграмні об'єкти показана на рис. 2.15.

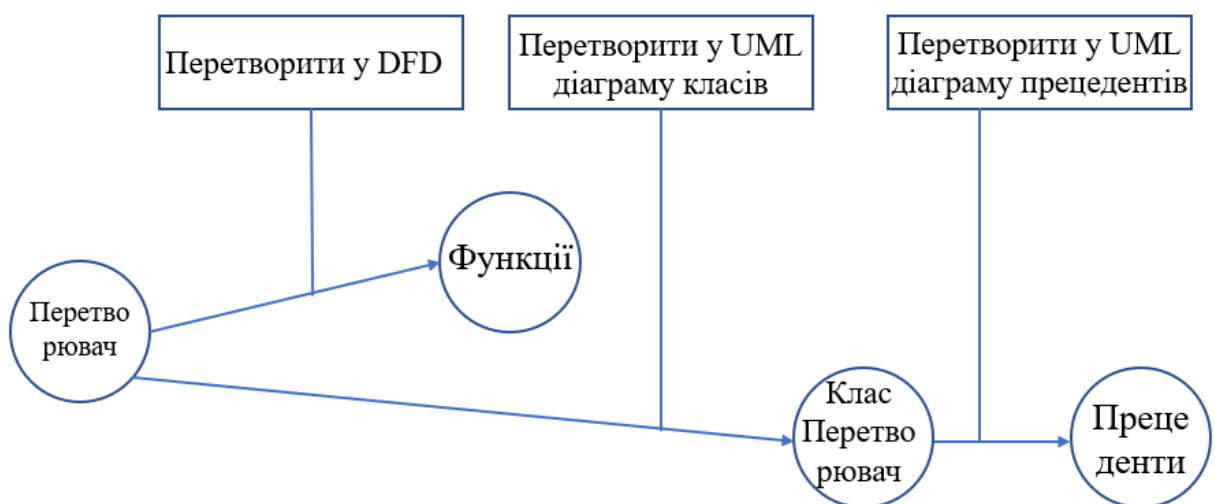


Рис. 2.15. Семантика переходу перетворювача у об'єкти діаграми

Перехід від КМПО МППР до КМПО ІС при роботі з перетворювачем

визначається результатом переходу відповідних ресурсів.

Сценарій переходу перетворювача на елементи ІС показаний на рис. 2.16.



Рис. 2.16. Блок–схема переходів перетворювача

Засоби (MECH)

Засобами виконання операцій можуть бути різні технічні пристрої, які оснащені робочим місцем людини, наприклад, контролери, датчики, комп'ютер, принтер, сканер. Інформація про їх характеристики, отримані в процесі

проектування, йде на технічні вимоги до розділу програмного забезпечення. У деяких ситуаціях оператора цих пристроїв також можна розглядати як засіб.

При розробці та моделюванні ІС необхідно враховувати особливості процесів обробки і передачі даних, оскільки в підсумку вони впливають на характеристики основного процесу.

При розробці людино–машинних програмних комплексів при вирішенні питань захисту інформації засобам встановлюються певні ролі та права доступу до певних об'єктів і функцій ІС.

Параметри (P)

Деякі демонстрації користувачеві характеристик процесу або операції. У найпростішому випадку це компонент «індикатор прогресу», що демонструє, як довго процес все ще буде працювати. це також може бути якесь значення, яке розраховується за формулою, наприклад, якась характеристика ресурсу. тому користувач повинен визначити формулу і спосіб відображення параметра (графік, текстове поле), тобто,

$$P_k \rightarrow Component_k, F_k \quad (2.13)$$

де P_k – k -ий параметр,

$Component_k$ – k -й візуальний компонент ПІ

F_k k -а функція програмного забезпечення.

Семантика переходу параметра показана на рис. 2.17.

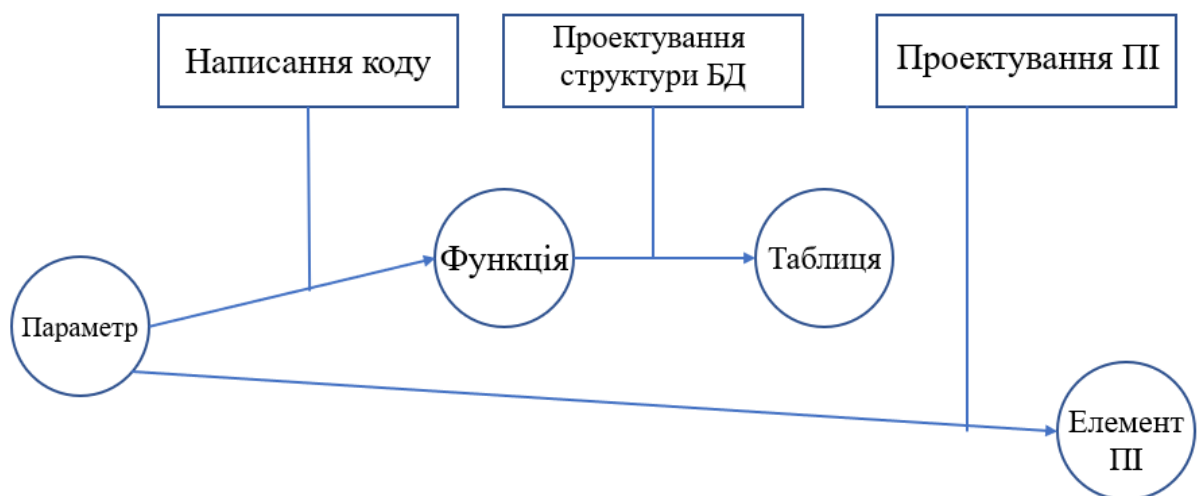


Рис. 2.17. Семантика переходу параметрів

Розглянемо сценарій переходу параметрів (рис. 2.18).

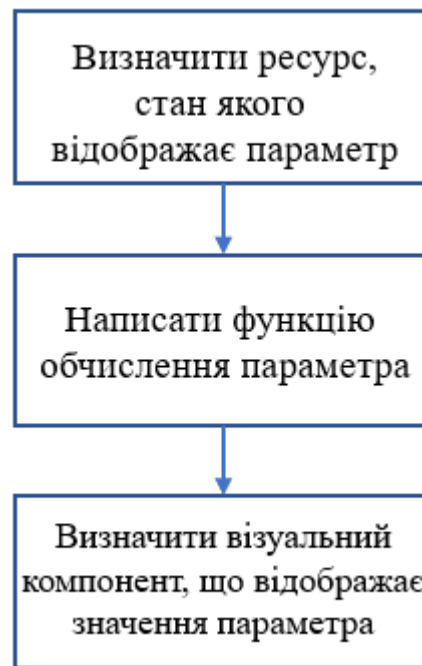


Рис. 2.18. Блок–схема переходу параметрів

Агенти (*Agent*)

Агент відповідає моделі ОПР, що має складну структуру. При його побудові використовуються підходи штучного інтелекту (наприклад, модель може базуватися на алгоритмі або сценарії поведінки, реалізованому у вигляді експертної системи; вона включає в себе: машину логічного висновку, базу знань, правила і робочу пам'ять).

З точки зору інформаційної системи, агент являє собою програмну сутність, у якої, при необхідності, є програмний інтерфейс, функції, що описують сценарій роботи агента, та таблиця для зберігання бази знань агента.

Можна реалізувати сценарій на рівні бази даних у вигляді збереженої процедури. Семантика переходу агента в елементи інформаційної системи показана на рис. 2.19.

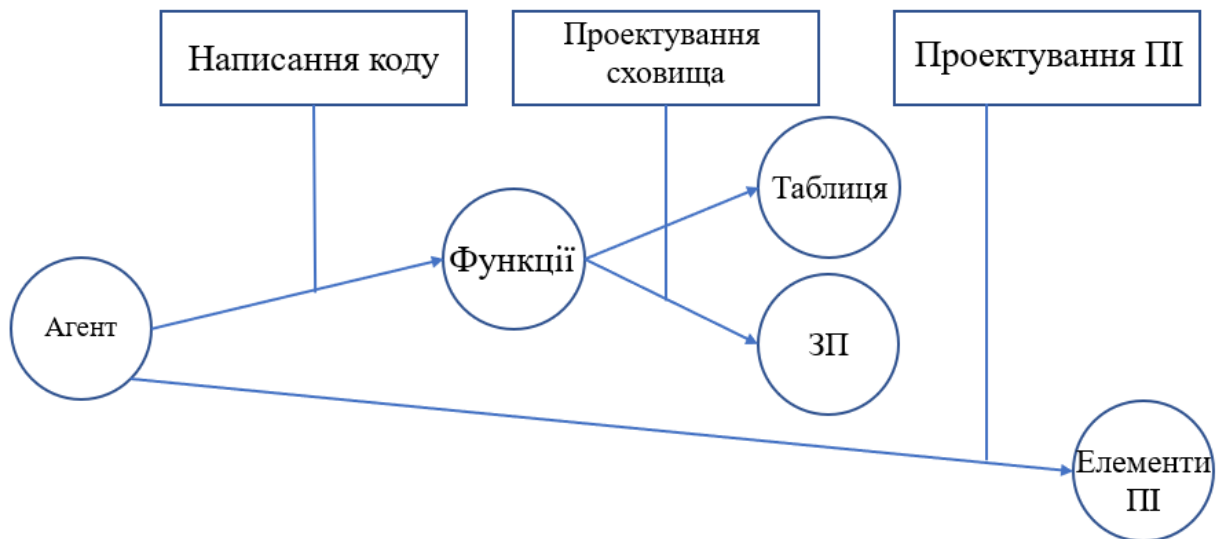


Рис. 2.19. Семантика переходу агента в елементи ІС

Семантика переходу агента у об'єкти діаграм показана на рис. 2.20. Схема послідовності імітує сценарій поведінки агента.

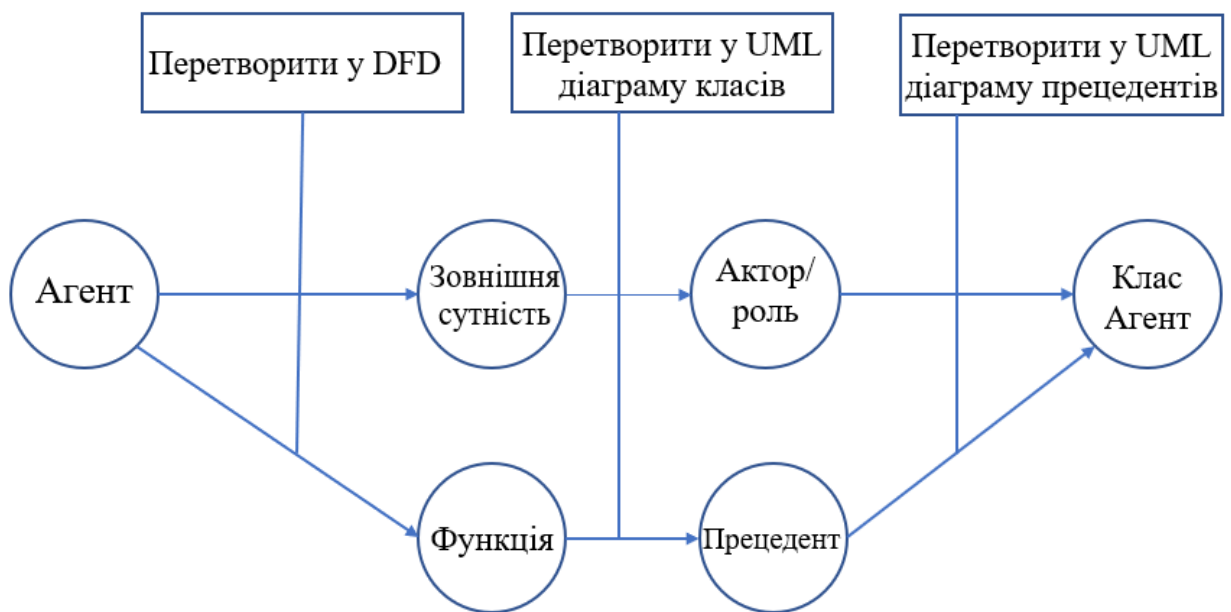


Рис. 2.20. Семантика переходу агента у об'єкти діаграм

Розглянемо сценарій переходу агента в об'єкти ІС (рис. 2.21).

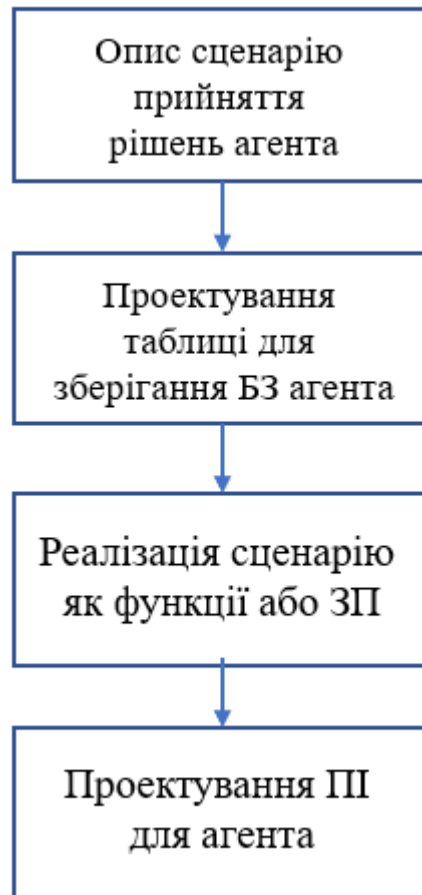


Рис. 2.21. Блок-схема переходу агента

Нижче наведено метод ППР розробки ІС предметної області МППР, що складається з наступних етапів [72]:

1 Розробка ІС починається з обстеження предметної області і побудови імітаційної моделі МППР «як є».

2 На другому етапі проводяться імітаційні експерименти з моделлю «як є» з метою виявлення «вузьких місць» в організації процесів.

3. На третьому етапі побудова моделі ІС здійснюється на основі даних моделі МППР.

3.1. Кожна операція моделі МППР, яка повинна бути автоматизована в ІС, перетворюється в функцію діаграми DFD

$$\forall Op_i \quad Op_i \rightarrow F_{DFD} .$$

На діаграмі класу створюється базовий клас операції, для кожної операції – екземпляр базового класу.

3.2. Всі ресурси, що використовуються в автоматизованих операціях, перетворюються в потоки даних діаграми DFD, а вхідні ресурси i -ї операції стають вхідними потоками i -ї функції DFD-діаграми

$$\forall Res_i: Res_i \in InOp_i \rightarrow DataFlowIn(Op_i),$$

а вихідні ресурси є вихідними потоками діаграми

$$DFD \forall Res_i: Res_i \in OutOp_i \rightarrow DataFlowOut(Op_i).$$

На діаграмі класів створюється базовий клас ресурсу. Для кожного ресурсу – екземпляр базового класу.

3.3. Для всіх ресурсів моделі необхідно створити сховище даних $Create(DataStore_i)$ на діаграмі DFD. На UML-діаграмі класу створюється базовий клас для сховища даних.

3.4. Всі агенти з моделі МППР, які будуть реалізовані програмним способом, перетворюються на зовнішні сутності діаграми DFD

$$\forall Agent_i Agent_i \rightarrow ExtrEss_i.$$

На діаграмі класу створюється базовий клас агента. Для кожного агента – екземпляр базового класу.

$$\forall Agent_i Agent_i \rightarrow Class(Agent_i).$$

На основі даних з DFD-діаграми створюються діаграми прецедентів. Кожна зовнішня сутність перетворюється на актора у відповідній діаграмі прецедентів

$$\forall ExtrEss_i ExtrEss_i \rightarrow Actor_i,$$

а пов'язані з нею функції – у прецеденти

$$\forall F_i F_i \rightarrow Case(Actor_i).$$

і3.4.1.Перетворення агента

Розглянемо приклад агента з одним правилом ("якщо" $a > b$, "то" $a = a - b$). Елементи пам'яті, необхідні для зберігання змінних, перетворюються в сховища даних DFD-діаграми. Правила "якщо" і "то" – в операції. Результатом є діаграма DFD, показана на рис. 2.22.

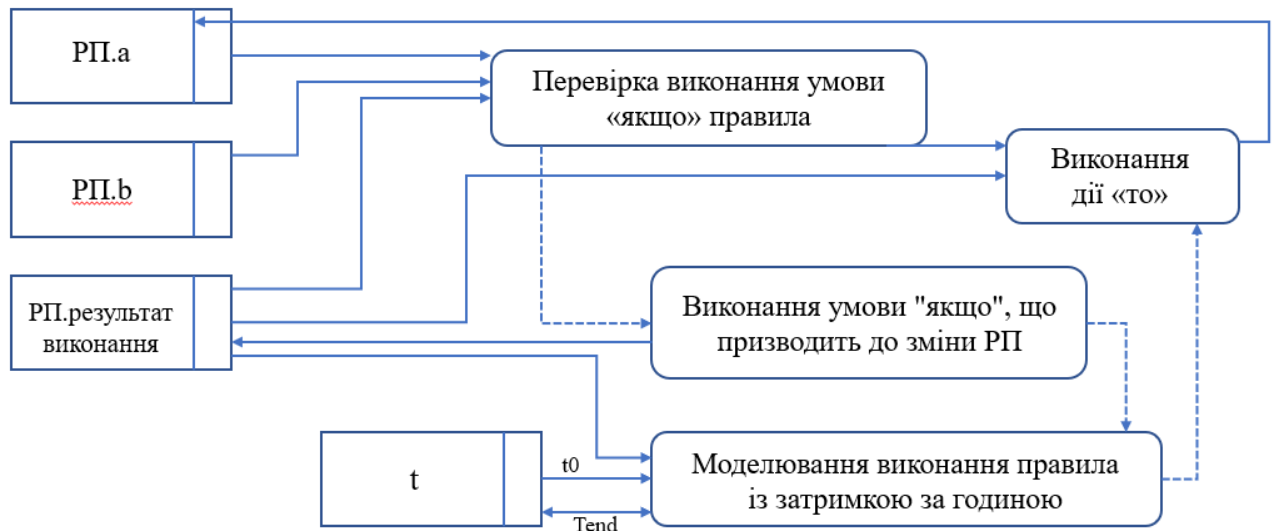


Рис. 2.22. Приклад DFD–діаграми для реактивного агента з одним правилом

3.4.2. Опис правил агента використовується для побудови діаграми прецедентів, тобто кожне правило переходить в прецедент.

3.4.3. Формули, що містяться в умовах «якщо» і «тоді» правил агента, переводяться в опис методу відповідного класу. На рис. 2.23 показана схема пошуку рішення ІА ІС при перетворенні агента на продукцію. Слід зазначити, що:

- ресурси, інструменти та квитки являють собою робочу пам'ять.
- на другому етапі, якщо жодна ситуація не діагностовано, то відбувається перехід на етап 11 (Завершення роботи).

База знань про дизайн ІА проектування ІС – являє собою опис об'єктів МППР та ІС.

3.5. На основі даних з DFD–діаграми створюються діаграми прецедентів. Кожна зовнішня сутність перетворюється на актора відповідній діаграмі прецедентів

$$\forall Ex_{DFD}' Ex_{DFD}' \rightarrow Actor_{UML}',$$

а пов'язані з нею функції в прецеденти

$$\forall F_{DFD}': F_{DFD}'(Ex_{DFD}')F_{DFD}' \rightarrow Case_{UML}'(Actor_{UML}').$$

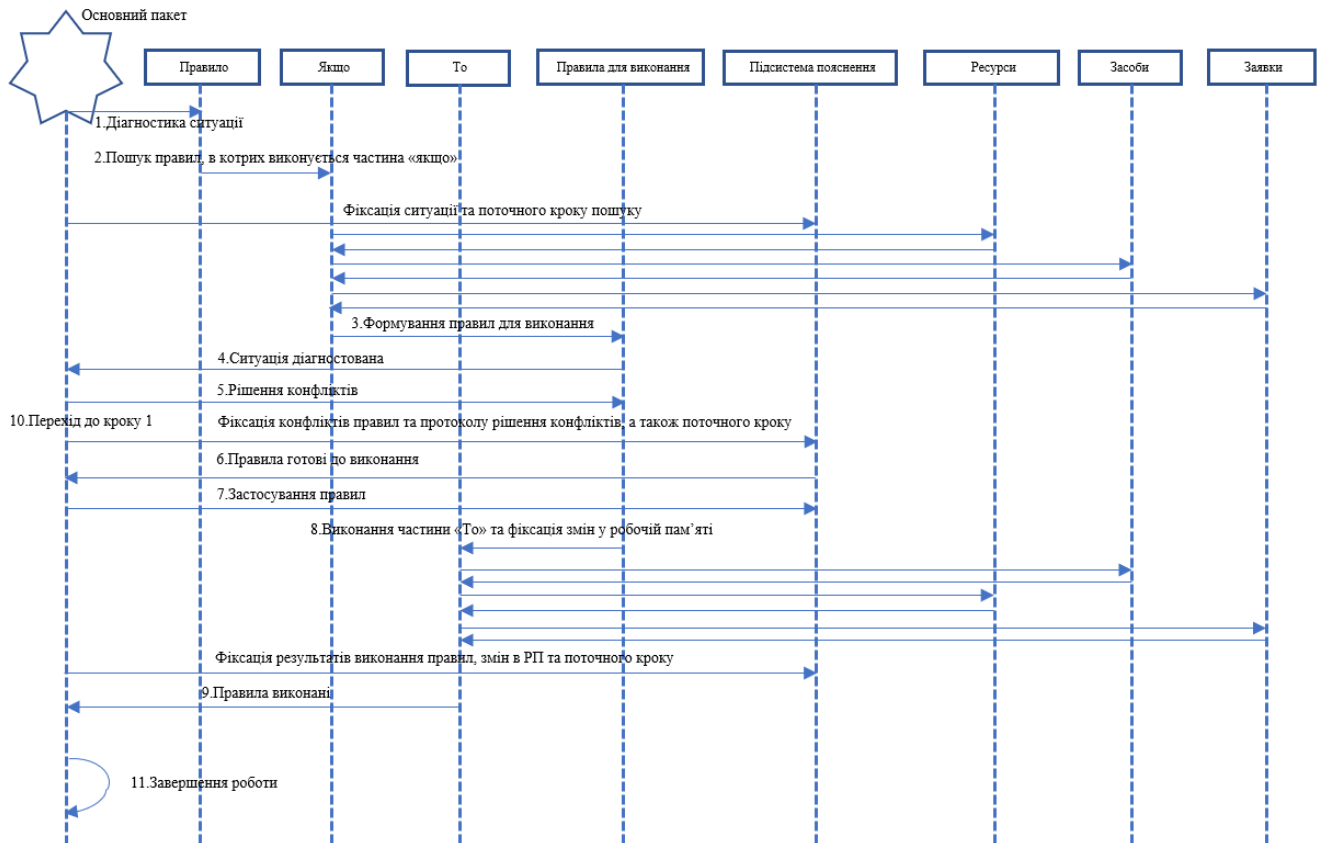


Рис. 2.23. Діаграма пошуку рішення ІА проектування ІС при перетворенні агента на продукцію.

3.6. Атрибути класів, які відповідають зовнішнім сутностям, дозволяють визначити структуру таблиць ER–діаграм.

4. Доробка системи розробниками, побудова діаграм послідовності та моделювання інтерфейсу користувача.

5. Вирішення питання розміщення екземплярів концептів предметної області у базах знань (*KBAg*) агентів. Для вирішення цієї задачі розглянемо проблему розміщення з дискретним простором рішень [79]. Його формулювання представлено нижче.

Знайти

$$\min z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.14)$$

з обмеженнями

$$\sum_{j=1}^n a_{ij} x_{ij} \geq 1, i = 1, \dots, m \quad (2.15)$$

$$x_{ij} = (0,1), i = 1, \dots, m; j = 1, \dots, n \quad (2.16)$$

де n – кількість агентів, m – кількість екземплярів концептів предметної області, c_{ij} – коефіцієнт, що показує величину витрат на розміщення i -го екземпляра концепту j -го агента,

$$x_{ij} = \begin{cases} 1, \text{ якщо } i - \text{й екземпляр концепту} \\ \text{розміщен у } j - \text{го агента} \\ 0, \text{ інакше} \end{cases}$$

a_{ij} – коефіцієнт, що визначає потребу j -го агента в i -му концепті,

$$a_{ij} = \begin{cases} 1, \text{ якщо } i - \text{й екземпляр концепту} \\ \text{розміщен у } j - \text{го агента} \\ 0, \text{ інакше} \end{cases}$$

Таким чином, необхідно розташувати екземпляри концептів в базах знань (*KBAg*) агентів з мінімальними витратами. Коефіцієнти c_{ij} визначаються експертами і дозволяють лінійно упорядкувати агентів за вартістю розміщення екземплярів концептів з урахуванням вартості розробки розподіленої ІС та її експлуатації. Для цього можуть бути використані методи експертної оцінки [80]. Це завдання не враховується в роботі.

Постановка, наведена для розміщення концептів з баз знань (*KBAg*) агентів, має свою особливість. з яких потрібно вибрати мінімальну вартість. В імітаційній моделі БП є інформація про те, якому агенту які екземпляри концептів необхідні, і ця інформація використовується в обмеженні завдання. По суті, відомі припустимі варіанти розміщення концептів, серед яких потрібно обрати мінімальний за витратами. Якщо потреби агентів у концептах досить відокремлені друг від друга, кількість можливих варіантів рішення істотно скорочується проти повним перебором можливих варіантів..

Оскільки завдання, що розглядається, вирішується в рамках автоматизації бізнес-процесів, то в порівнянні з класичним завданням необхідно враховувати, що дії агентів є складовою частиною автоматизованого бізнес-процесу і повинні

вкладатися в певні тимчасові рамки $T_{\text{БП}}^{\text{max}}$ – максимальний час виконання бізнес–процесу. Отже, до обмежень необхідно додати обмеження на час виконання бізнес–процесу ($T_{\text{БП}}$), яке у загальному випадку матиме вигляд:

$$T_{\text{БП}} \leq T_{\text{БП}}^{\text{max}} \quad (2.17)$$

Розглянемо вирішення цієї проблеми для розробки системи обслуговування реєстру. Необхідно розташувати 5 (п'ять) реєстрів між 3 (трьома) агентами. Юридично існує обмеження на виконання операції – 3 дня. У разі порушення термінів можуть бути накладені значні штрафи. Особливість даної системи полягає в тому, що не всі агенти рівноправні. Один з них, центральний офіс, повинен зберігати у себе всі 5 реєстрів, а решті двох агентів потрібен доступ тільки до тих реєстрів, з якими вони працюють. Експертні оцінки витрат на розміщення концептів предметної області за базою знань наведено у табл. 2.3.

Таблиця 2.3

Експертні оцінки витрат на розміщення

концепти\агенти	1	2	3
1	2	1	1
2	2	1	1
3	2	1	1
4	2	1	1
5	2	1	1

Постановка задачі:

$$\min z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

у разі обмеження:

$$a_{11} + a_{21} + a_{31} + a_{41} + a_{51} \geq 1$$

$$a_{12} + a_{22} \geq 1$$

$$a_{23} + a_{33} + a_{43} \geq 1$$

$$x_j = (0,1), j = 1, \dots, 5$$

$$T_{\text{БП}} \leq 3$$

Для наочності ми представляємо обмеження в табличному вигляді (табл. 2.4): на перетині i -го строки і j -го стовбця стоїть 1, якщо $a_{ij} = 1$.

Таблиця 2.4

Подання значень a_{ij} у вигляді таблиці

агенти\концепти	1	2	3	4	5
1	1	1	1	1	1
2	1	1	0	0	0
3	0	1	1	1	0

Розглянемо можливі рішення.

Варіант 1 – розміщення всіх екземплярів концептів у центральному офісі. У табл. 2.5 представлені значення x_{ij} та цільової функції z для 1 варіанта.

Таблиця 2.5

Рішення для варіанту 1

агенти\концепти	1	2	3	4	5	z
1	1	1	1	1	1	10
2	0	1	0	0	0	
3	0	0	1	0	0	

Всі інші варіанти розміщення призводять до дублювання концептів предметної області, оскільки за умовою центрального офісу обов'язково мають бути всі реєстри. Наприклад розглянемо кілька варіантів розміщення з дублюванням.

Варіант 2. У табл. 2.6 представлені значення x_{ij} та цільової функції z для 2-го варіанта.

Таблиця 2.6

Рішення для варіанту 2

агенти\концепти	1	2	3	4	5	z
1	0	1	1	1	1	14
2	1	0	1	0	0	
3	0	0	1	1	0	

Варіант 3. т. У табл. 2.7 представлені значення x_{ij} та цільової функції z для 3–го варіанта.

Таблиця 2.7

Рішення для варіанту 3

агенти\концепти	1	2	3	4	5	z
1	0	1	1	1	1	14
2	0	1	1	0	0	
3	1	0	1	1	0	

Варіант 4. У табл. 2.8 представлені значення x_{ij} та цільової функції z для 4–го варіанта.

Для четвертого варіанту.

Таблиця 2.8

Рішення для варіанту 4

агенти\концепти	1	2	3	4	5	z
1	1	0	1	1	1	12
2	1	0	1	0	0	
3	0	1	0	0	0	

Сучасний рівень розвитку інформаційних технологій дозволяє реалізувати таку інформаційну систему із дотриманням тимчасових обмежень на БП.

Оскільки специфіка БП така, що тривалі простої у роботі ІС можуть призвести до суттєвих штрафів, необхідно оточити рівень надійності системи. Безпосередній розрахунок ймовірностей відмов і збоїв у складній і

багатокомпонентній розподіленій системі не можливий, хоча б у зв'язку з відсутністю необхідних даних про надійність компонентів.

Подібні розрахунки проводяться хіба що в авіабудуванні, але й визначальним фактором в оцінці надійності є досвід експлуатації. Зауважимо, що всі ці оцінки носять імовірнісний характер, задається лише деяка допустима межа ймовірності відмови. Вони не можуть гарантувати відсутності збоїв.

Експлуатаційна надійність серверів та апаратури ІС оцінюється зазвичай сумарними часами простою протягом календарного року. Ці дані фіксуються у логах і можуть бути легко проаналізовані. В даний час апаратна частина ІС, а також операційна система та спеціалізоване ПЗ досить надійні. Для продукції різних фірм фіксуються перебої у функціонуванні з тривалістю від десятків хвилин до кількох годин протягом року через внутрішні апаратні та системні проблеми [81]. Проте досвід роботи реальних розподілених ІС показує, що перебої, пов'язані з обміном даними між агентами, можуть становити кілька робочих днів.

Нештатні та незаплановані простої у роботі ІС, природно, призводять до збитків. Тому інтерес представляє не просто імовірнісні характеристики, а прив'язка їх до можливих фінансових та інших втрат. У нашому завданні, виявилось, досить важко побудувати цільову функцію, яка адекватно враховувала б юридичний та фінансовий ризик, пов'язаний із простоєм у роботі, що перевищує нормативно допустимий.

Для аналізу зв'язку між величиною надійності та тривалістю простоїв у роботі [82] наводиться така таблиця (див. табл. 2.9).

Таблиця 2.9

Класифікація систем за рівнем надійності

Коефіцієнт готовності, K_r	Максимальний час простою на рік	Тип системи
0,99	3,5 діб.	Звичайний
0,999	8,5 год.	Висока надійність
0,9999	1 год.	Відмовостійкий
0,99999	5 хв.	Безвідмовний

Фактично ці результати можна отримати у простій дискретній моделі – схемі Бернуллі. Відмови (у схемі Бернуллі називаються успіхами) характеризуються величиною Kr і приймаються за параметр p . "Випробуванням" буде тимчасова одиниця (добу або годину), а довжиною послідовності випробувань – кількість тимчасових одиниць у календарному році. У цьому тривалість простою оцінюється як математичне очікування ($N \cdot p$) числа «успіхів» у схемі Бернуллі.

Справді, $N = 365 \cdot 24 = 8760$ годин, тоді $N \cdot p = \frac{8760}{100} = 87,5$ год. Або 3,65 діб., як і стоїть у рядку таблиці для звичайної системи. Для системи високої надійності $N \cdot p = \frac{8760}{100} = 87,5$ год.

Як відомо, розподіл Бернуллі такий, що "успіхи" в серії можуть бути не лише ізольовані "невдачами", а й утворювати кластери, тобто. ланцюжки, що йдуть поспіль, "успіхів". У багатьох прикладних завданнях, зокрема і в аналізованих, інтерес представляє ймовірність того, що кластери будуть мати певну довжину. Якщо періоди добового простою розділені проміжками нормального функціонування, такий рівень надійності представляється прийнятним. Проте кластер довжиною понад 3 доби – неприйнятний.

Розглянемо схему Бернуллі, у якій кожен робочий день розглядається як випробування, а подією є можлива перерва у роботі протягом цього робочого дня. Параметр p – можливість перерви в роботі в цей день. Звичайна схема Бернуллі дає можливість обчислити ймовірність до "успіхів" у серії N випробувань

$$P_N(k) = C_N^k * p^k * (1 - p)^{N-k} \quad (2.18)$$

Проте більш цікавою є завдання про ймовірність кількох "успіхів", що йдуть поспіль. Таким чином моделюється тривалий простій системи протягом декількох робочих днів поспіль, що дозволяє точніше оцінити ризик тривалого переривання обслуговування клієнтів.

У літературі з теорії ймовірностей дане завдання багаторазово розглядалося

у різних варіантах постановки, наприклад, [83]. Слід зазначити, що вона припускає точне аналітичне рішення [84]. У той самий час це завдання має досить широку застосовність, наприклад, щодо надійності комп'ютерних мереж [85].

У розрахунках була використана формула для підрахунку ймовірності того, що довжина кластера 1_N у послідовності довжиною N перевершить деяке критичне значення m [86]:

$$P_N(1_N \geq m) = \sum_{j=1}^{\lfloor \frac{N}{m} \rfloor} (-1)^{j+1} \left(p + \left(\frac{N - jm + 1}{j} \right) (1 - p) \right) C_{N-jm}^{j-1} p^{jm} (1 - p)^{j-1} \quad (2.19)$$

Використання цієї формули при малих значеннях ймовірностей може призводити до істотних похибок. Тому обчислення проводилися в цілісному режимі пакета Maple. Завдяки цьому, похибка округлення проявилася тільки при переведенні результуючих значень із раціонального дробу до десяткового.

Результати наведені в табл. 2.10.

Таблиця 2.10

Розв'язання задачі про ймовірність послідовності серії "успіхів", що йдуть поспіль
(2 дні та 3 дні)

p	$\frac{1}{30}$	$\frac{1}{50}$	$\frac{1}{100}$	$\frac{1}{500}$	$\frac{1}{1000}$
$P(_{365} \geq 2)$	0,32	0,133	0,035	0,015	0
$P(_{365} \geq 3)$	0,013	0,0022	0,0004	0,000	0

Таким чином, в рамках розглянутої моделі значення для ймовірності відмови порядку 0.01 ($r = 0.99$) слід вважати як мають високий ризик, зменшити який прийняттого значення можна лише системи високого рівня надійності, тобто $Kr = 0.999$. Отже, зниження сумарного простою з кількох діб до кількох годин відповідає підвищенню надійності на порядок.

Найпростішим вирішенням цієї проблеми є резервування (резервування

заміщенням з ненавантаженим резервом). Нехай у разі відмови основної системи підключається резервна, значно дешевша в експлуатації та, відповідно, менш надійна.

Розрахунок надійності в даному випадку можна проводити у схемі гіпотез, де гіпотезами будуть відповідно безвідмовна робота основної системи та її відмова:

$$\sum_{k=1}^l P(A|H_k) * P(H_k) = P(A) \quad (2.20)$$

де

l – кількість гіпотез,

$P(A|H_k)$ – умовна ймовірність події A під справедливністю H_k

$P(H_k)$ є апіорною ймовірністю гіпотез

У прикладі візьмемо за ймовірності відмов відповідно 0.1 (резервна система) і 0.01 (основна система). Тоді за формулою повної ймовірності отримаємо:

$$0.99 + 0.01 * 0.9 = 0.999$$

Очевидно, що система з ненавантаженим резервом та надійним перемиканням дає підвищення загальної надійності на порядок проти відсутності резервування.

На рис. 2.24 представлена схема БП у тому випадку, коли БЗ розташована у центрі. В цьому випадку:

$$T_{\text{БП}} = t_{\text{обробки}_{\text{А1}}} + t_{\text{обробки}_{\text{Центр}}} + t_{\text{обробки}_{\text{Центр}-\text{А1}}} \quad (2.21)$$

де

$t_{\text{обробки}_{\text{А1}}}$ – час роботи агента1, включаючи людино–машинну участь;

$t_{\text{обробки}_{\text{Центр}}}$ – час роботи Центру, включаючи людино–машинну участь;

$t_{\text{обробки}_{\text{Центр}-\text{А1}}}$ – час передачі результатів роботи з Центру агенту 1.

Розглянемо тепер можливі варіанти резервування системи.

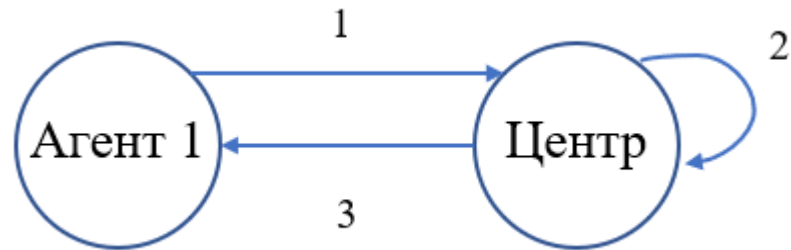


Рис. 2.24. Схема БП у разі розташування БЗ в Центрі

Варіант 1. Розміщення концептів, необхідних роботи агента в нього. Цей варіант представлений рис. 2.25. В цьому випадку:

$$T_{\text{БП}} = t_{\text{обробки}_{A1}} + t_{\text{передачі}_{A1-\text{Центр}}} + t_{\text{обробки}_{\text{Центр}-A1}} + t_{\text{передачі}_{\text{Центр}-A1}} \quad (2.22)$$

Де

$t_{\text{обробки}_{A1}}$ – час роботи агента1, включаючи людино–машинну участь;

$t_{\text{передачі}_{A1-\text{Центр}}}$ – час передачі результатів роботи з агента1 до Центру;

$t_{\text{обробки}_{\text{Центр}-A1}}$ – час роботи Центру, включаючи людино–машинну участь;

$t_{\text{передачі}_{\text{Центр}-A1}}$ – час передачі результатів роботи з Центру агенту1.

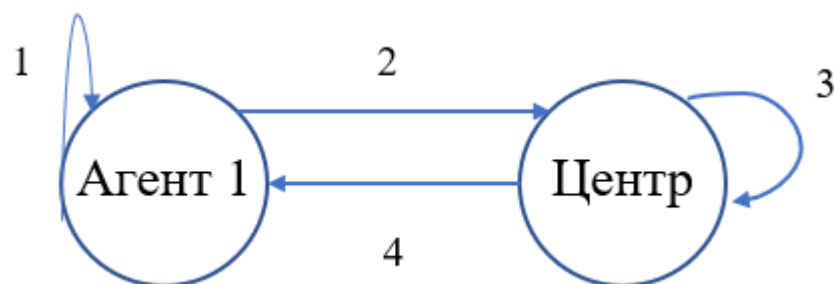


Рис. 2.25. Схема БП у разі розміщення БЗ у Центрі, дублювання концептів у агентів

У разі відсутності 2–3 дні можливості обміну інформацією між агентом та Центром можливе $T_{\text{БП}} > T_{\text{БП}}^{\text{max}}$ – Отже, резервування БЗ у агента може не вирішити проблеми.

Варіант 2. Розміщення концептів, необхідних роботи агента в нього та резервний спосіб передачі повідомлень до Центру. Цей варіант представлений рис. 2.26.

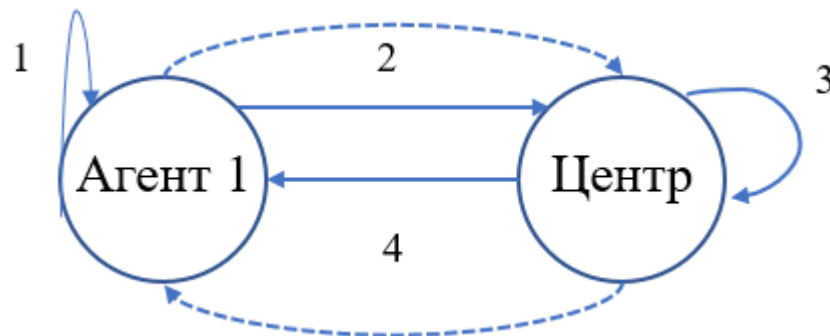


Рис. 2.26. Схема БП у разі розподіленої ІС, дублювання концептів у агентів та резервний спосіб передачі повідомлень до Центру

В цьому випадку при помилках в обміні інформацією між Центром та Агентом 1 використовується резервний спосіб доставки інформації:

$$T_{\text{БП}} = t_{\text{обробки}_{\text{А1}}} + t_{\text{передачі_рез}_{\text{А1}-\text{Центр}}} + t_{\text{обробки}_{\text{Центр}-\text{А1}}} + t_{\text{передачі_рез}_{\text{Центр}-\text{А1}}} \quad (2.23)$$

Де

$t_{\text{обробки}_{\text{А1}}}$ – час роботи агента1, включаючи людино–машинну участь;

$t_{\text{передачі_рез}_{\text{А1}-\text{Центр}}}$ – час передачі результатів роботи з агента1 до Центру резервним способом;

$t_{\text{обробки}_{\text{Центр}-\text{А1}}}$ – час роботи Центру, включаючи людино–машинну участь;

$t_{\text{передачі_рез}_{\text{Центр}-\text{А1}}}$ – час передачі результатів роботи з Центру агенту1 резервним способом.

Отже, для завдання необхідно резервувати базу знань агента і розробляти резервний спосіб передачі даних.

Аналіз архітектурних рішень людино–машинної розподіленої системи на початок процесу кодування дозволяє знизити ризик відмови такої системи у процесі експлуатації.

У табл. 2.11 наведені відповідності елементів МППР елементам ІС.

Таблиця 2.11

Відповідність елементів МППР елементам ІС

Елементи МППР	Елементи DFD	Елементи UML (діаграми)			Елементи ІС		
		преце- дентів	класів	послідовн ості	КІ	Код	Рівень БД
Ресурс	Потік даних, сховище даних	відсутні	Клас	Клас, параметр метода	Поле введення , таблиця	Змінна, файлова змінна, таблиця	Таблиця
Перетвор ювач	Функція	Прецеден т	Метод класу	Метод	Рядок введення , таблиця	Функція	ЗП
Агент	Зовнішня сутність	Актор	Клас	Актор, клас	Інтерфейс с програ много модуля	Код програ много модуля	Таблиці, ЗП

Таким чином, запропонований метод дозволяє перетворити КМПО МППР на КМПО ІС.

Слід зазначити, що на сьогоднішній момент пропонуються методи розробки ІС, що так чи інакше зачіпають аналіз процесів ОТС.

Метод Скобелєва П.О.

Метод Скобелєва призначений для створення МАС з оперативної обробки інформації для підтримки процесів прийняття рішень. В якості моделі ОТС пропонується модель мережі потреб і можливостей (ПМ–мережі) підприємства

[87]. При цьому кожне підприємство представлено у вигляді мережі агентів потреб і можливостей.

Метод розробки Скобелева П.О. включає в себе наступні етапи: опис предметної області МАС; опис класів агентів і правила прийняття рішень; опис протоколів взаємодії агентів; типів і структури повідомлень; програмної реалізації агентів.

Даний метод реалізується у складі набору компонентів для розробки мультиагентних систем – Magenta Toolkit [88]. Налаштування системи для конкретної предметної області передбачено за допомогою інструменту для створення онтологій. Інструментарій призначений для розробки МАС, пов'язаних з плануванням і виділенням ресурсів. Він не займається аналізом і реінжинірингом БП.

Метод Карсаєва О.В. та Городецького В.І.

Метод Карсаєва О.В. і Городецького В.І. заснований на методології Gaia [89] і середовищі MASDK [90], що підтримує його використання. Він призначений для розробки прикладних мультиагентних систем.

Метод складається з десяти етапів, виконаних в певній послідовності. Результати кожного етапу визначають послідовність виконання наступних етапів. Рішення, описані одним етапом, є вихідними даними для виконання наступних етапів. У сукупності метод можна описати в п'ять етапів.

Перший етап «Проектування прикладного МАС» включає в себе два етапи:

- аналіз області;
- опис онтології предметної області.

Результатом етапу аналізу області є завдання виявлення та зіставлення класів агентів з ними. Розподіл ролей між класами агентів визначає, які класи агентів забезпечать рішення для визначених завдань на більш пізніх стадіях розвитку.

Другий етап «Проектування класів агентів» присвячений опису трьох компонентів, що утворюють структуру агента:

- схема поведінки агента;
- Сервісні моделі

– ментальної моделі.

На даному етапі існує тільки опис агентів з використанням діаграм з використанням об'єктно–орієнтованого підходу.

Написання програмного коду відбувається на третьому етапі. У цьому випадку описані функції сервісу. Після цього автоматично генерується програмний код класів агентів.

На четвертому етапі є опис агента, що складається з даних, необхідних для їх установки, і знань і правил його поведінки.

На останньому етапі агенти розгортаються в мережі.

Таким чином, метод Карсаєва О.В. і Городецького В.І. не дозволяє описати статичний і динамічний БП, а, отже, не займається питаннями їх аналізу і реінжинірингу.

Метод Швецова А.Н.

Запропонований метод стосується розробки корпоративних інтелектуальних систем підтримки прийняття рішень [66]. На початковому етапі велика увага приділяється опису структурних, логічних і поведінкових аспектів функціонування ОТС. На етапі формалізації будується структурно–логічна модель, база знань, топологічна і об'єктна моделі.

Цей метод реалізований у якості програмного комплексу DISIT (англ. Distributed Intellectual System Integrated Toolkit [92] – Інтегрований інструментарій розподіленої інтелектуальної системи). Він призначений для розвитку МАС, виходячи з наступних принципів:

- опис моделі предметної області, з використанням концепцій фрейму;
- опис поведінки агентів у вигляді продуктів.

У цьому пакеті інструментів послідовно виконуються такі кроки методу:

- презентація моделі предметної області;
- заповнення моделі логікою взаємозв'язку між поняттями фрейму та їх атрибутами;
- виявити інтелектуальних агентів і визначити їх поведінку з урахуванням

системних обмежень;

- переклад отриманої концептуальної моделі предметної області в структурно–логічну модель МАС;
- розміщення інтелектуальних компонентів і агентів в корпоративній мережі.

Таким чином, метод Швецова А.М. не дозволяє описати статичний і динамічний БП, а, отже, не займається питаннями їх аналізу і реінжинірингу.

Метод Олександрова Д.В.

Олександров запропонував метод моделювання розподілених систем управління БП підприємства [93]. При аналізі предметної області пропонується використовувати структурно функціональний підхід. Моделювання моделей на апараті пофарбованих мереж Петрі. За результатами імітаційного моделювання пропонуються рекомендації щодо вдосконалення БП, а при необхідності і тактична реінжиніринг, що полягає в додаванні / видаленні І функції, співробітники, перерозподіл функцій між співробітниками, переведення співробітників з одного структурного підрозділу в інший і т.д. Реалізація агентської програми для автоматизації реалізації ВР. Також метод передбачає використання ІМ для контролю БП підприємства.

Для порівняльної оцінки методів розробки інформаційних систем пропонується наступний набір критеріїв:

- модель ОТС–процесів, яка повинна описувати статичний і динамічний БП, а також модель ОПР;
- інструменти аналізу процесів, включаючи реінжиніринг організаційних процесів, аналіз вузьких місць;
- можливість використання даних з моделі ОТС при розробці ІС;
- використання структурних та об'єктно–орієнтованих підходів;
- результати автоматизації (бізнес–процеси, координація прийняття рішень, ППР – використання МЛВ).

У табл. 2.12 наведені результати порівняння методів розробки ІС.

Порівняння методів розробки ІС

Критерії порівняння	Метод Скобелєва	Метод Городецького, Карсєва	Метод Швецова	Метод Олександрова	Новий метод
1. Модель процесу ОТС	ПМ–мережі	НІ	НІ	Розфарбовані мережі Петрі	Модель МППР
1.1. Статичний бізнес–процес	ТАК	НІ	НІ	ТАК	ТАК
1.2. Динамічний бізнес–процес	НІ	НІ	НІ	НІ	ТАК
1.3. Модель ОПР	ТАК	ТАК	ТАК	НІ	ТАК
2. Засоби аналізу процесів					
2.1. Реінжиніринг організаційний	НІ	НІ	НІ	ТАК	НІ
2.2. Аналіз «вузьких місць»	НІ	НІ	НІ	НІ	ТАК
3. Розробка ІС					
3.1. На основі моделі ОТС у частині динамічних БП	НІ	НІ	НІ	НІ	ТАК
3.2. На основі моделі ОТС у частині ОПР	ТАК	ТАК	ТАК	НІ	ТАК
3.3. Структурний підхід	НІ	НІ	НІ	ТАК	ТАК

Продовження таблиці 2.12.

3.4. ОО підхід	ТАК	ТАК	ТАК	НІ	ТАК
4. Результати автоматизації					
4.1. Бізнес процеси	ТАК	ТАК	ТАК	ТАК	ТАК
4.2. Погодження рішень	ТАК	НІ	ТАК	НІ	ТАК
4.2. ППР – використання МЛВ	НІ	НІ	ТАК	НІ	ТАК

Існуючі методи не повною мірою вирішують проблему розробки ІС, що впливає на аналіз ОТС–процесів. Вони не враховують динаміку БП, не приділяють достатньої уваги аналізу «вузьких місць», не використовують інформацію з моделі ОТС–процесів в умовах динамічного БП для розвитку ІС. Запропонований метод вирішує ці проблеми.

2.6. Аналіз затримок в синхронній інформаційній системі

При проектуванні ІС необхідно вирішити питання про те, наскільки децентралізованою повинна бути його структура, чи повинні окремі компоненти реалізовуватися у вигляді інтелектуальних агентів [94]. При цьому виникають питання про ступінь структурування і розподілу окремих агентів, синхронізації роботи і обміну між окремими модулями ІС.

На початковому етапі проектування моделі МППР визначається якась початкова структура системи, визначені модулі, які необхідно реалізувати у вигляді програмних агентів. Однак на етапі опису БП не завжди очевидно більш детальне структурування. використовувати імітаційне моделювання.

Як правило, відомі або задані обсяги, необхідні для нормального функціонування готового ІС і характер обміну інформацією між окремими компонентами. Практично обмін інформацією між віддаленими компонентами системи може здійснюватися в різних протоколах і мережевих рішеннях. програмних модулів.

Як приклад такого моделювання розглянемо обмін, пов'язаний з входом клієнта з віддаленого комп'ютера на сервер і роботою з графічним інтерфейсом користувача. Оцінки необхідного ресурсу пропускної здатності за середніми характеристиками можуть бути значно занижені.

Нехай «додатком» стане звернення клієнта (оператора) до системи через графічний інтерфейс через інтернет-канал. «Обробка» запиту ускладнюється заповненням оператором тієї чи іншої форми.

Інтенсивність запитів в реальних ІС віддаленого доступу по суті є стохастичним і нестаціонарним процесом. Наводимо на рис. 2.27 залежність частоти запитів від часу в реальному ІС.

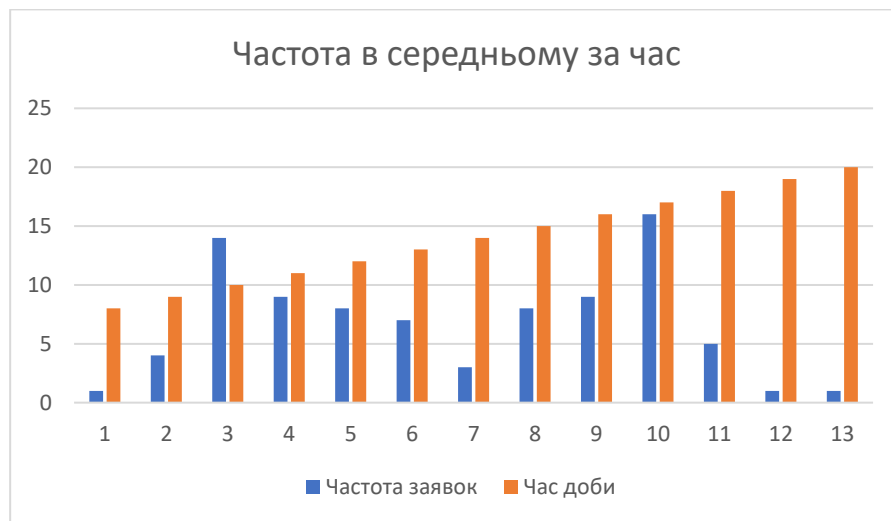


Рис. 2.27. Розподіл заявок протягом доби

Як бачите, з одного боку, в деякі періоди часу частота запитів значно збільшується, в порівнянні з середньодобова. Однак навряд чи економічно доцільно мати надлишок ресурсів, орієнтованих на максимальні показники.

Імітаційне моделювання здійснювалася в найпростішій моделі системи

масового обслуговування (СМО) $m/m/1$, оскільки отриманих результатів достатньо для принципового рішення про поділ на підсистеми. Навіть у роботі А. Дорогува (CERN Bulletin dec, 1999) було показано, що результати чисельного моделювання $m/m/1$ (та інших) прекрасно узгоджуються зі значеннями тих числових характеристик СМО, які можуть бути отримані в рамках простої теорії.

Потоки запитів і послуг розглядаються як з інтенсивністю λ і μ , відповідно. При цьому часові інтервали між послідовними запитами T_a і часом обслуговування T_s мають індикативний розподіл з відповідними параметрами. властивість цього розподілу, а саме очікуване значення і стандартне відхилення дорівнюють $1/\lambda$ ($1/\mu$). Приблизно $2/3$ значень випадкової змінної, що має показовий розподіл, менше математичного очікування і $1/3$ більше (рис. 2.28).

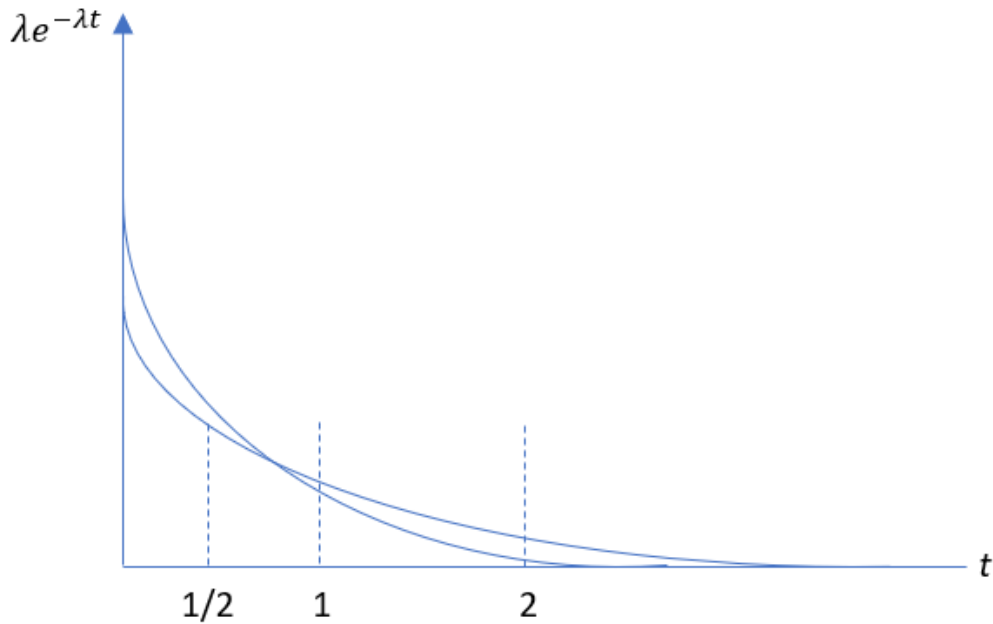


Рис. 2.28. Щільності ймовірності показового розподілу з $\lambda = 1$ і $\lambda = 2$

Це призводить до цікавого ефекту. Нехай $T_a = 2 T_s$, іншими словами, не може бути черги з регулярним періодичним отриманням заявок.

«Хвости» на кривих розподілу призводять до того, що значна кількість додатків надходять частіше, ніж через T_a , а час обслуговування перевищує T_s . Наскільки значним буде збільшення середнього часу обслуговування, можна судити по значенню миттєвої черги q_c .

Алгоритм розрахунку наступний. Послідовно за допомогою вбудованого генератора випадкових чисел отримуємо значення часу приходу та часу обслуговування (з відповідними параметрами). Порівнюючи ці часи та склад черги, або збільшується на одиницю, або зменшується. Зазначений алгоритм був реалізований у мові, що вільно розповсюджується програми *R*, що використовує алгоритм *Mersenne–Twister*. Крім того, було додано блок візуалізації отриманих результатів у вигляді скрипту для програми *TikZ*, що дозволяє отримати графіки у вигляді *pdf* файлів.

Отримані значення довжини миттєвої черги (рис. 2.29) досягає 4 та дають конструктору основу для оптимального структурування системи і впровадження програмних агентів. Аналогічне моделювання здійснюється для всіх «вузьких місць» БП. Остаточне рішення про структурування модуля приймається за результатами моделювання МППР.

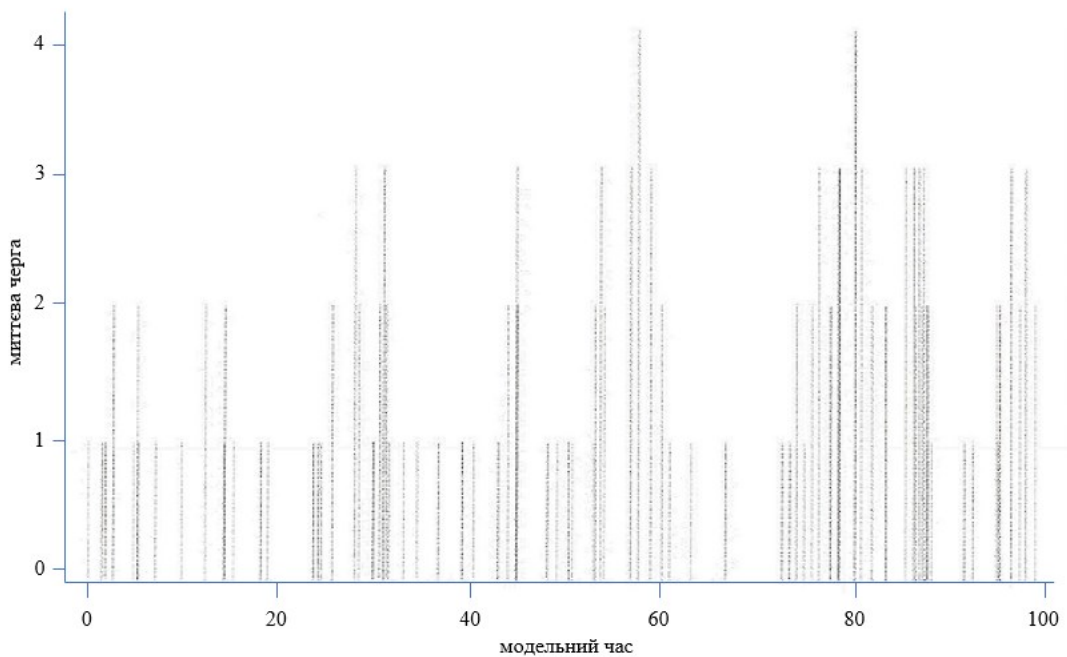


Рис. 2.29. Довжина миттєвої черги при $T_s/T_a=4/3$

Використання імітаційного моделювання для аналізу роботи окремих елементів ІС дозволяє оцінити запропоновані рішення для архітектури програмного забезпечення до його роботи, а, отже, можна регулювати їх в процесі розробки.

2.7. Вибір та застосування методики оцінки ефективності до методу розробки інформаційних систем

Оцінимо ефективність роботи запропонованого методу проектування ПЗ. Одним з головних критеріїв оцінки є час виконання проекту зі створення ПЗ – $T_{\text{ПЗ}}$ – воно складається з часу виконання кожного етапу проекту та часу переходу між етапами. На практиці на різних етапах розробки беруть участь різні фахівці, тому потрібна адаптація результатів попереднього етапу для фахівців наступного, тому виникає час переходу між етапами.

$$T_{\text{ПЗ}} = T_{\text{обст}} + T_{\text{форм}} + T_{\text{модел}} + T_{\text{проект}} + T_{\text{розр}} + T_{\text{перехід}} \quad (2.24)$$

де $T_{\text{обст}}$ – час проведення обстеження (1 етап);

$T_{\text{форм}}$ – Час формалізації БП, тобто. час побудови моделі (2 етап);

$T_{\text{модел}}$ – час проведення моделювання та реінжинірингу (3 етап);

$T_{\text{проект}}$ – час проектування ПЗ (4 етап);

$T_{\text{розр}}$ – час розробки ПЗ (5 етап);

$T_{\text{перехід}}$ – сумарний час, що витрачається під час переходу від одного етапу проекту до іншого.

$$T_{\text{перехід}} = T_{\text{обст} \rightarrow \text{форм}}_{\text{перехід}} + T_{\text{форм} \rightarrow \text{модел}}_{\text{перехід}} + T_{\text{модел} \rightarrow \text{проект}}_{\text{перехід}} + T_{\text{проект} \rightarrow \text{розр}}_{\text{перехід}} \quad (2.25)$$

де $T_{\text{перехід}}^{i \rightarrow j}$ – це час переходу між i та j етапами.

Час проведення обстеження $T_{\text{обст}}$ і час переходу між 1 і 2 етапами $T_{\text{обст} \rightarrow \text{форм}}_{\text{перехід}}$ перехід, не залежить від того, використовувався чи ні запропонований метод.

У процесі розробки ІС будуються DFD–діаграми та діаграми мови UML. Відтак час проектування складається з двох складових:

$$T_{\text{проект}} = T_{DFD} + T_{UML} + T_{DFD \rightarrow UML}^{\text{перехід}} \quad (2.26)$$

де T_{DFD} – час побудови всіх діаграм DFD;

T_{UML} – час побудови моделі архітектури програмного забезпечення у вигляді UML–діаграм;

$T_{DFD \rightarrow UML}^{\text{перехід}}$ – час переходу від побудови DFD–діаграм до UML–діаграм. (На практиці пов'язано з передачею знань від аналітика БП до архітектора ПЗ.)

Час побудови однієї DFD–діаграми визначається кількістю функціональних блоків та потоків даних. Отже,

$$T_{DFD} = \sum (n * t_{\text{бл}} + m * t_{\text{потік}}) \quad (2.27)$$

де n – кількість функціональних блоків;

$t_{\text{бл}}$ – час малювання одного блоку;

m – кількість потоків даних;

$t_{\text{потік}}$ – це час малювання одного потоку.

Запропонований метод використовує наступні три види UML–діаграм: прецедентів, послідовності та класів. Отже,

$$T_{UML} = T_{UC} + T_{Int} + T_{Class} \quad (2.28)$$

де T_{UC} – час побудови всіх діаграм прецедентів;

T_{Int} – час побудови всіх діаграм послідовності;

T_{Class} – час побудови всіх діаграм класів

Час побудови однієї діаграми прецедентів, у випадку, залежить кількості варіантів використання ПЗ. За її автоматичної генерації на основі даних з DFD–діаграми ця залежність незначна порівняно з ручним моделюванням. Отже, при використанні запропонованого методу час побудови однієї діаграми прецедентів можна вважати величиною постійної (t_{UC}), тоді

$$T_{UC} = \sum t_{uc} \quad (2.29)$$

При використанні запропонованого методу $T^{DFD \rightarrow UML}_{\text{перехід}}$ буде значно менше з допомогою процесу автоматизації моделювання. У разі використання методу знижується фактор втрати частини інформації при переході від однієї моделі до іншої.

У запропонованому методі пропонується автоматично генерувати заготовлі програмних модулів, що описують класи та форми ПІ. Отже, час, витрачений перехід від етапу проектування до етапу розробки ПЗ $T^{\text{проект} \rightarrow \text{розр}}_{\text{перехід}}$ скорочується в порівнянні з іншими методами.

Аналіз запропонованого методу показує, що він дозволяє зменшити такі часи:

$$T^{DFD \rightarrow UML}_{\text{перехід}}, T_{UC}, T^{\text{проект} \rightarrow \text{розр}}_{\text{перехід}}$$

Висновок до розділу 2

У другому розділі було проведено дослідження та обрано фреймово–семантично–репрезентативної моделі знань, що дає наступні переваги:

- узгодженість фреймово–семантичної моделі з концепцією об’єктно орієнтованого програмування виправдовує використання об’єктного програмування при розробці програмного забезпечення і мінімізує витрати на створення програмного забезпечення. [61, 62];

- єдина модель представлення знань, яка є відповідною, як показано в наступних розділах, для предметної області мультиагентних процесів перетворення ресурсів та інформаційної системи.

На основі обраної моделі представлення знань в розділі описано концептуальну модель області мультиагентних процесів перетворення ресурсів. Також досліджено предметні області мультиагентних процесів перетворення ресурсів та описано основні об’єкти концептуальної моделі предметної області

мультиагентних процесів перетворення ресурсів, а також предметної області інформаційної системи та побудовано концептуальну модель предметної області інформаційної системи.

Розроблено модель та метод у області створення інформаційної системи, які відрізняються від існуючих:

1) методикою системного аналізу та моделлю для формалізації мультиагентних процесів перетворення ресурсів.

2) імітаційна модель для перевірки моделі «як буде» на етапі реінжинірингу БП, та оцінки продуктивності інформаційної системи;

3) інтелектуальністю розробки інформаційної системи, що включає функціональний та об'єктно-орієнтований аналіз, моделювання програмних інтерфейсів, формування виконуваного коду інформаційної системи;

Розроблено концептуальну модель предметної області організаційно-технічної системи, що заснована на фреймово-семантичному представленні знань

Удосконалено метод підтримки прийняття рішень для завдання розробки інформаційної системи на основі інтеграції структурних, агентних та об'єктно-орієнтованих підходів

Розроблено метод трансформації моделі організаційно-технічної системи в модель інформаційної системи, що сформована як сукупність діаграм функціонального та об'єктно-орієнтованого підходу на основі використання концептуальних моделей предметної області.

Аналіз надійності архітектурних рішень під час автоматизації процесів організаційно-технічних системах за умов обмежень за термінами виконання бізнес-процесів особливо ефективний у разі географічно розподіленої системи.

Запропонований метод підтримки прийняття рішень при розробці інформаційних систем мультиагентних процесів перетворення ресурсів дозволяє зменшити час, що витрачається на перехід від функціонального моделювання до об'єктно-орієнтованого, а також проектування для розробки програмного забезпечення.

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ ЕАВР ДЛЯ ВІЗУАЛЬНОГО МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

На базі функціоналу програмного комплексу Sparx Systems Enterprise Architect було розроблено модуль ЕАВР в якому використано розроблений метод СППР. Sparx Enterprise Architect – це інструмент моделювання повного життєвого циклу на основі UML, який використовується для планування, проектування та створення програмно–інтенсивних систем та бізнес–процесів за допомогою візуального та графічного дизайну, створений для допомоги інженерним групам у створенні надійних і зручних у супроводі систем, заснований на OMG UML. Платформа підтримує проектування та створення програмних систем, моделювання бізнес–процесів та моделювання галузевих доменів. На основі моделі даних вона дозволяє відстежувати вимоги до проектування, а звідти – до вихідного коду.

3.1. Функціональність модуля ЕАВР

ЕАВР є CASE–інструментом для автоматизації процесу проектування діаграм DFD, IDEF0, UML–схем прецедентів, послідовностей і класів, а також ПІ розробленої ІС [60, 92–96].

В рамках процесу моделювання архітектури ІС ЕАВР пропонує користувачеві наступні функції:

1. Опис бізнес–процесів, автоматизованих розробленими ІС за допомогою схем стандарту DFD. Діаграми DFD розкладаються на будь–який рівень деталізації. Автоматично створюйте діаграми DFD на основі моделі МППР.
2. Опис функцій, що виконуються користувачами системи в рамках автоматизованих процесів, з використанням схем кейсів використання. Дозволяється будувати схеми випадків використання з нуля, тобто створювати нову діаграму, або з використанням автоматизованого перетворення з DFD, де вибираються зі списку існуючих на діаграмі DFD суб'єкти ставляться у відповідність із зовнішніми суб'єктами, а прецеденти встановлюються на

функції. Діаграми, отримані шляхом перетворення, можна редагувати. Ви можете створити необмежену кількість діаграм інцидентів використання для кожної діаграми DFD.

3. Для кожного випадку використання, виконаного користувачем, може бути дано опис послідовності елементарних операцій в системі. Для цього призначені діаграми послідовностей, які можуть бути створені для будь-якої функції діаграми прецеденту. Система дозволяє частково автоматизувати цей процес: розробнику пропонується перенести на схему послідовність обраних суб'єктів з діаграми прецеденту.

Для кожного об'єкта діаграми послідовностей один з чотирьох типів визначається відповідно до конкретного пакета (межі, актори, управління, бізнес-об'єкти).

4. Створення діаграми класів і діаграму послідовності карт (крім меж) до класів на цій діаграмі.
5. Проектування візуальних форм змодельованого програмного забезпечення (граней схем послідовностей): розміщення компонентів на формі, методи прив'язки та властивості класу до компонентів, а також збереження моделей форм у форматах .dim та .crr. Надалі ці моделі передаються програмісту для імпорту в середовище C# і подальшого розвитку алгоритмів.
6. Створення звітів про створений проект із зображеннями розроблених схем і форм. Звіти можна друкувати та експортувати до Word.
7. Збереження проекту на сервері та завантаження з MS SQL Server для редагування.

3.2. Опис CASE-засобу EABP

3.2.1 Загальна структура інструменту CASE EABP

CASE-інструмент EABP складається з трьох підсистем: створення DFD-діаграм, UML-діаграм, моделювання ПІ. Структура EABP представлена на рис. 3.1 [96].

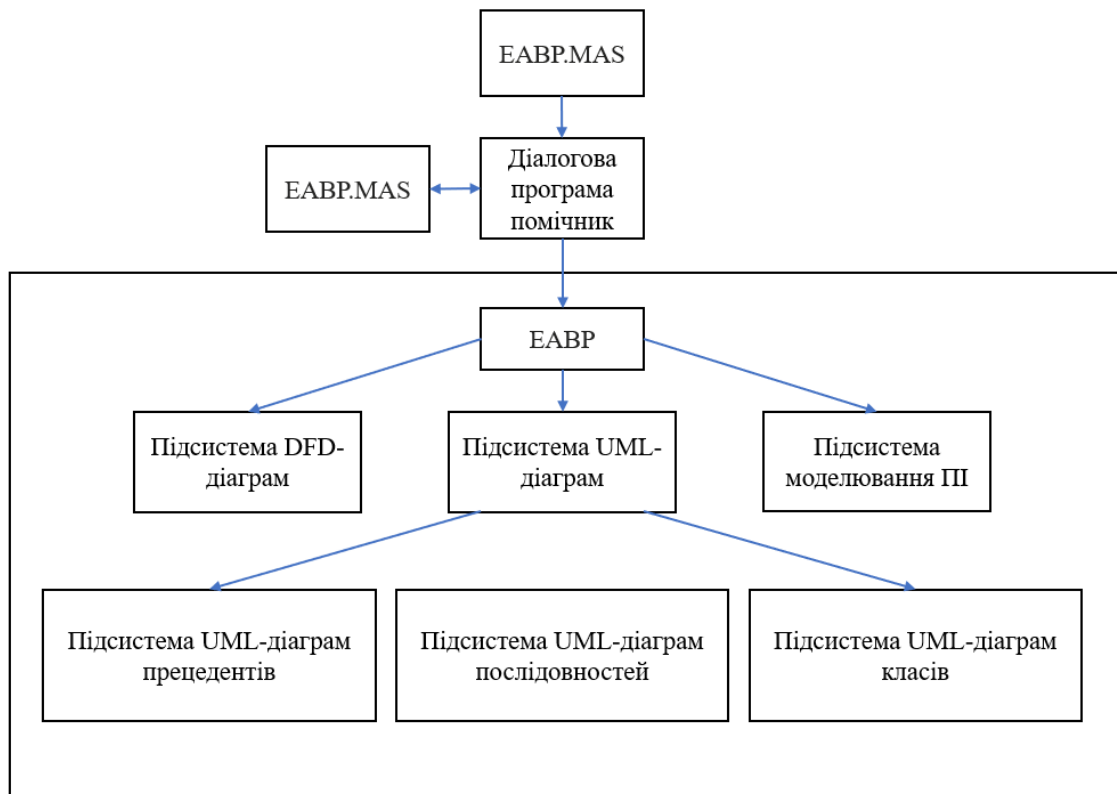


Рис. 3.1. Структура інструменту CASE EABP

Підсистеми більш детально розглянуті в наступних розділах.

Після введення облікових даних користувача і встановлення підключення до сервера відкривається основна форма системи, показана на рис. 3.2.

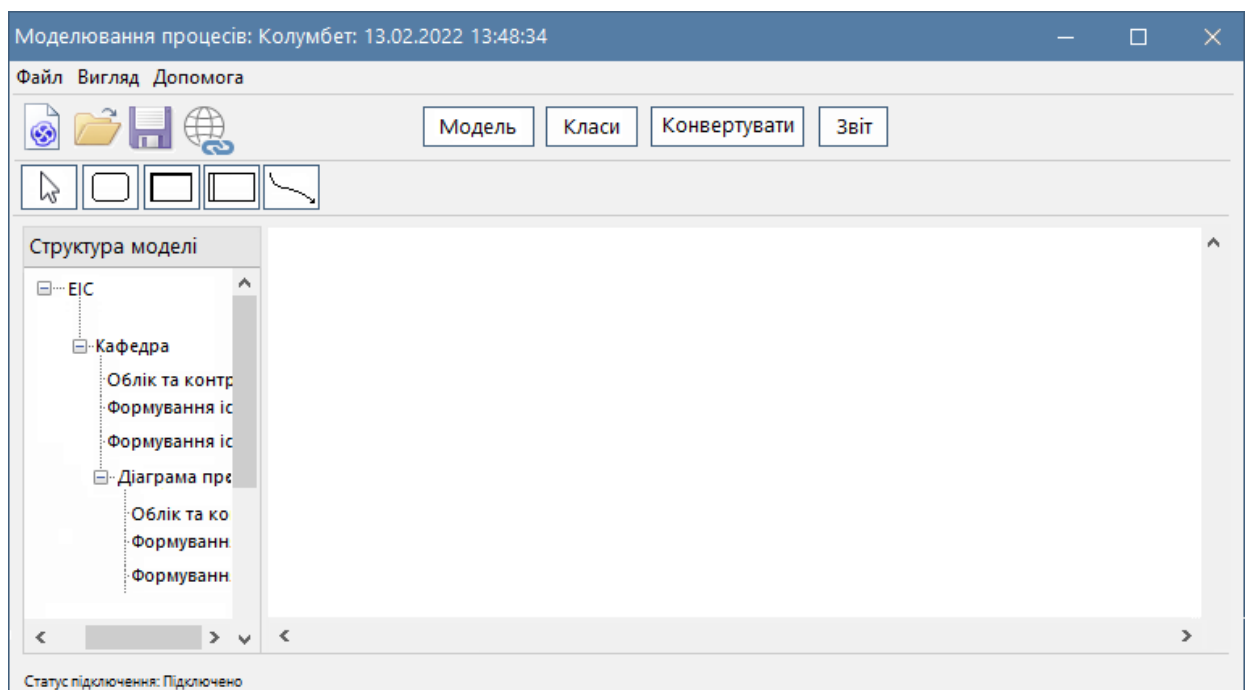


Рис. 3.2. Основна форма EABP

Основна форма складається з наступних частин:

- назва, що відображає інформацію про поточний проект (ім'я, автори, дата і час створення);
- головне меню системи надає користувачеві доступ до всіх функцій програми;
 - панелі інструментів – дублюють основні команди головного меню, в ньому містяться інструменти для створення і редагування діаграм класів;
- область створення діаграми – тут ви можете безпосередньо створювати та редагувати діаграми класів;
- рядок стану – відображає інформацію про поточний стан підключення до вибраного SQL Server.

3.2.2 Створення діаграми DFD

Після створення нового проекту кнопки з об'єктами діаграми DFD автоматично активуються в області панелі інструментів (рис. 3.3).

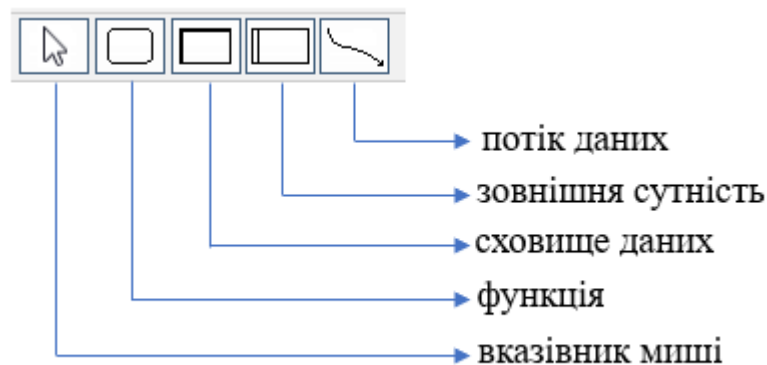


Рис. 3.3. Панель інструментів діаграми DFD

Щоб помістити об'єкт в область діаграми, натисніть на відповідну кнопку панелі інструментів і на діаграмі, де потрібно розмістити об'єкт.

Коли ви наводите курсор або натискаєте на об'єкт, він підсвічується червоним кольором. Розмір об'єкта можна змінити за правим і нижнім правим кутами об'єкта, утримуючи ліву клавішу миші (рис. 3.4).

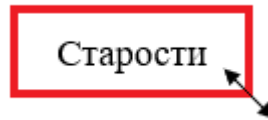


Рис. 3.4. Зміна розміру об'єкта

Щоб з'єднати об'єкти потоками даних, натисніть відповідну кнопку панелі інструментів. Коли ви наводите курсор на вказівник миші, об'єкти (включаючи межі фігури), які можна з'єднати, виділяються синім кольором; над межами форми та об'єктів курсор також змінює свій вигляд, щоб вказати, що потік може бути витікати звідси (рис. 3.5). Об'єкти, які наразі пов'язані, також виділяються синім кольором.

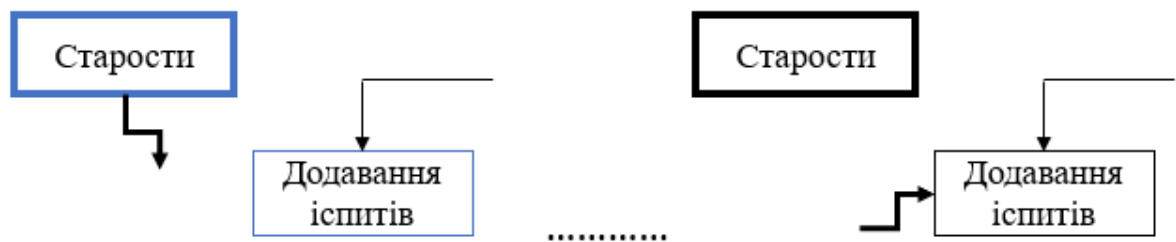


Рис. 3.5. З'єднання об'єктів за допомогою потоків даних

Контекстне меню потоку даних «Тип» може вибравати один з 5 типів (рис. 3.6).

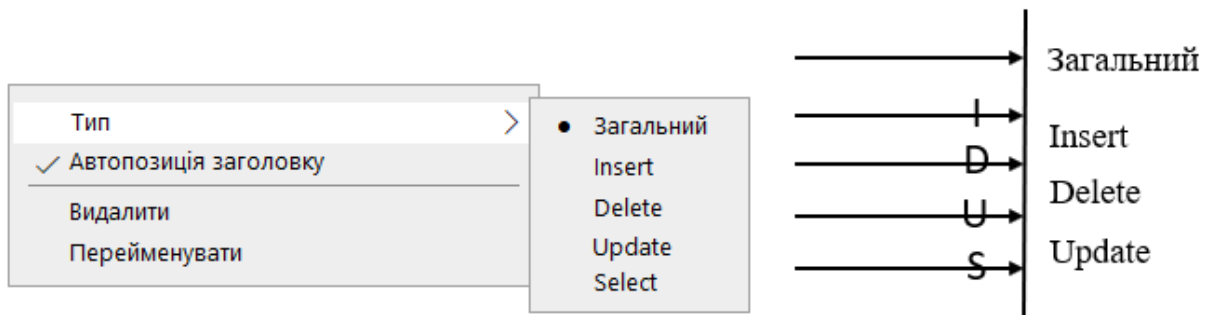


Рис. 3.6. Типи потоків даних

Переміщення об'єктів і заголовків потоків виконується лівою рукою, і всі

потоки даних, пов'язані з цим об'єктом, автоматично перебудовуються. Редагування потоків даних здійснюється шляхом перетягування вузлів або відрізків ліній (рис. 3.7). Автоматичне розміщення заголовка потоку даних викликається через пункт «Автопозиція заголовка» контекстного меню потоку (рис. 3.6).

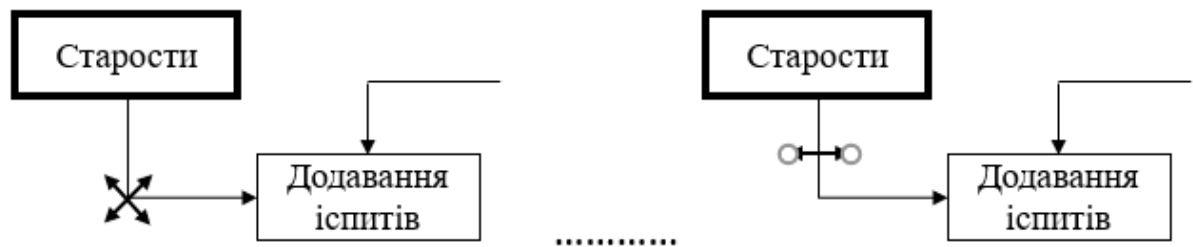


Рис. 3.7. Редагування потоків даних

Всі функції, розміщені на діаграмі DFD, вводяться в дерево проекту. Перейменування об'єктів викликається з контекстного меню об'єкта або відповідного елемента дерева (рис. 3.8).

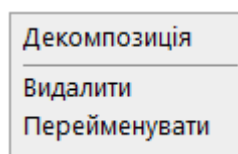


Рис. 3.8. Контекстне меню функцій діаграми DFD

Відкриється додаткове вікно для потоків даних (рис. 3.9),
Дає змогу вибрати ім'я зі списку вже наявних імен у цьому проекті.

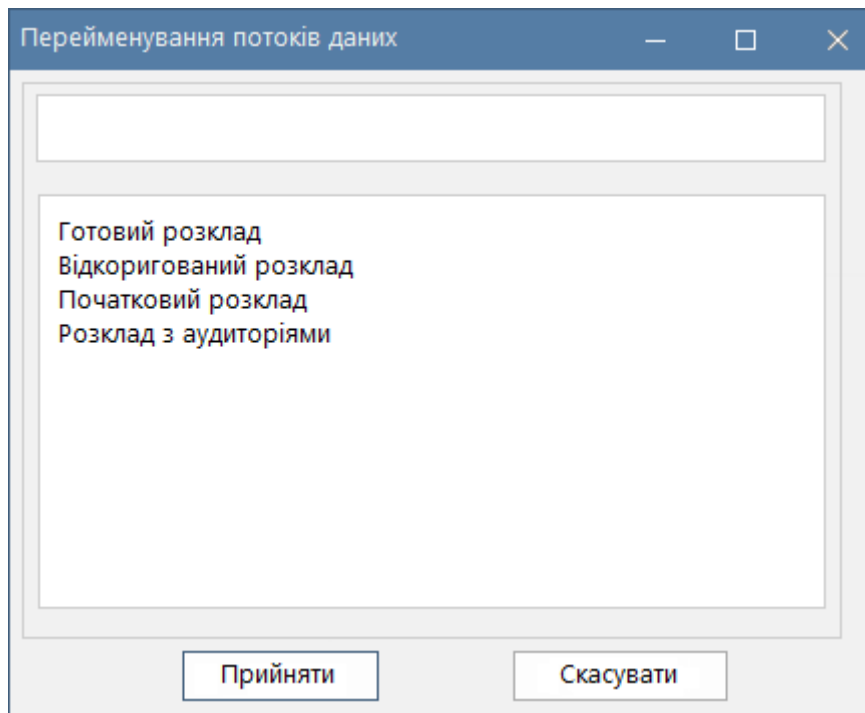


Рис. 3.9. Форма «Переименовать поток данных»

Подібні вікна відкриваються для сховищ даних і зовнішніх сутностей, якщо обрати з існуючого в контекстному меню (рис. 3.10).

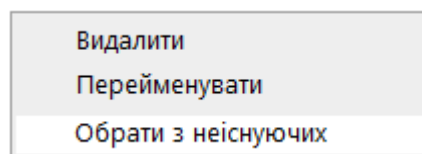


Рис. 3.10. Контекстне меню сховищ даних та зовнішніх сутностей
діаграм DFD

Щоб видалити об'єкти, виберіть «Видалити» з контекстного меню елемента дерева об'єкта або проекту (рис. 3.8). Усі теми, пов'язані з цим об'єктом, також буде видалено.

Для створення декомпозиції діаграми виберіть «Декомпозиція» в контекстному меню функції декомпозиція (рис. 3.8). Потім у вікні необхідно вказати початкову кількість елементів (функцій) в декомпозиції (рис. 3.11).

Рис. 3.11. Форма «Створення декомпозиції»

Після натискання кнопки ОК автоматично створюється нова діаграма, яка містить вказану кількість функцій.

На рис. 3.12 наведено приклад повної діаграми DFD.

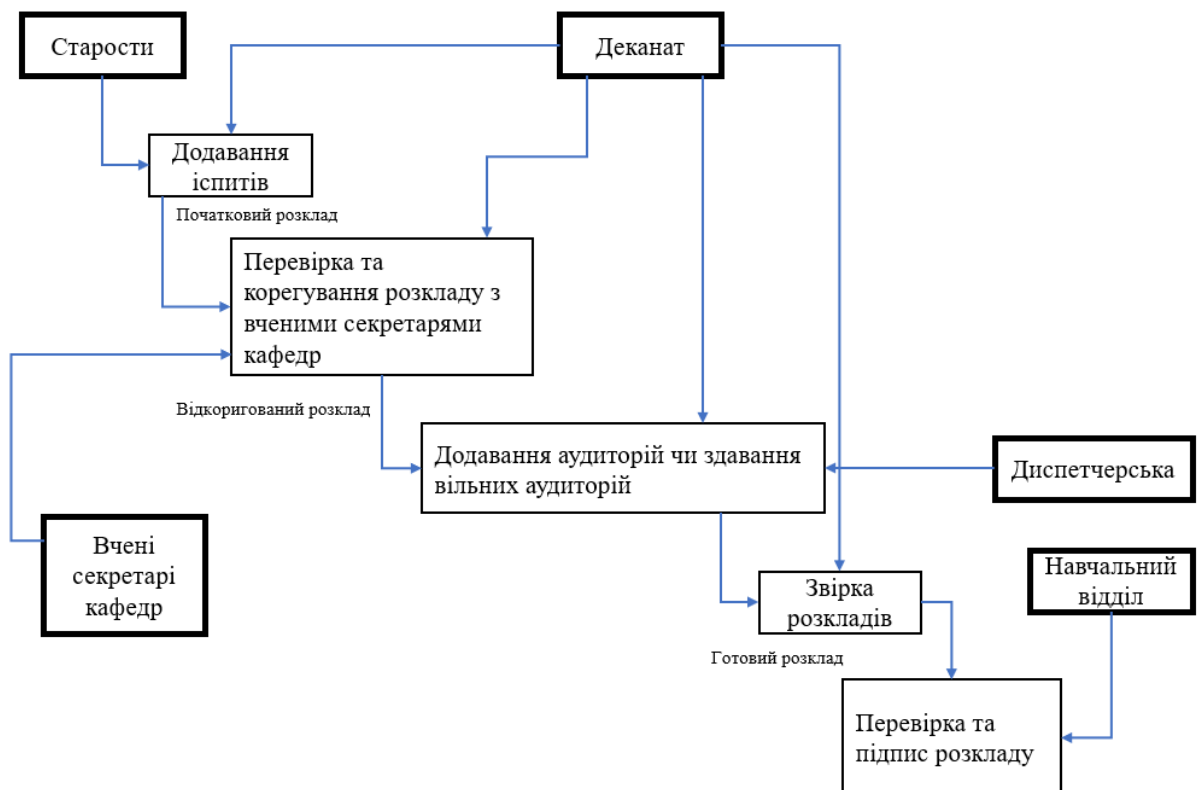


Рис. 3.12. Діаграма DFD

3.2.3. Створення діаграми прецедентів

Створення діаграми прецедентів здійснюється шляхом перетворення з діаграми DFD за розробленою методологією, заміни функцій діаграми DFD функціями діаграми прецеденту, а зовнішній об'єкт господарювання з суб'єктами.

Для початку процесу перетворення потрібно натиснути кнопку «Конвертувати» на панелі інструментів (рис. 3.13).

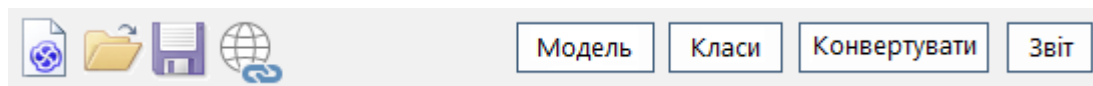


Рис. 3.13. Стандартна панель інструментів

У вікні (рис. 3.14) вибираються існуючі на даній діаграмі функції і зовнішні сутності, які необхідно відобразити на діаграмі прецедентів. За необхідності збереження зв'язків між цими об'єктами встановлюється відповідний прапорець внизу вікна.

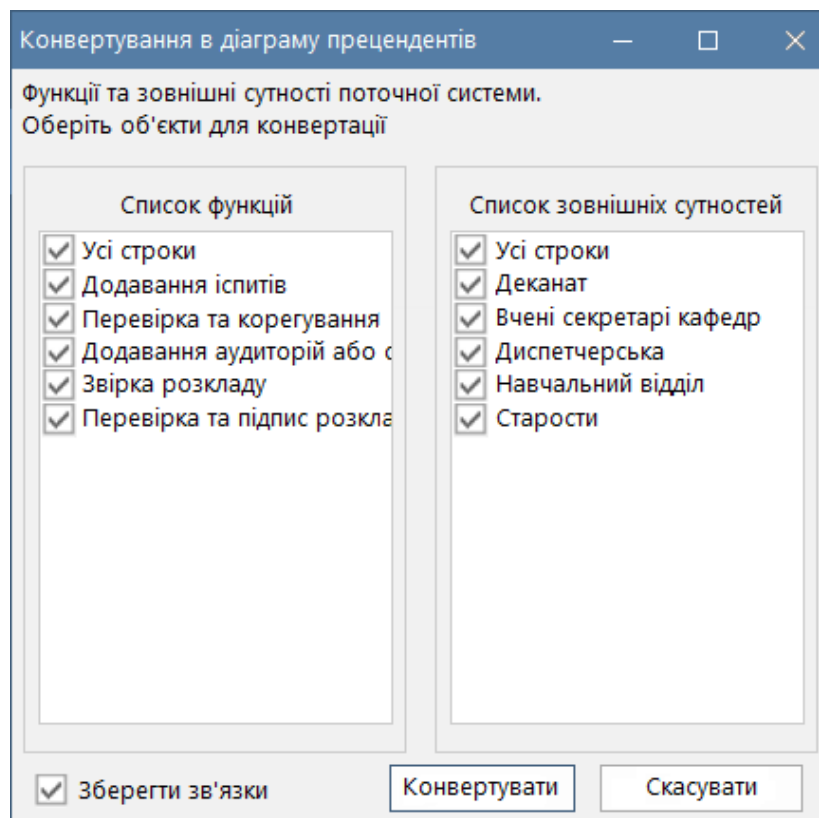


Рис. 3.14. Конвертувати в діаграму прецедентів

В результаті перетворення в дереві проекту з'являється відповідний вузол зі списком функцій, що існують на схемі прецеденту. Побудована діаграма показана на рис. 3.15.



Рис. 3.15. Діаграма прецедентів

Дозволяється створювати нову або редагувати існуючу діаграму за допомогою панелі інструментів діаграми прецедентів (рис. 3.16).

Дозволило Створіть нову діаграму або відредагуйте наявну діаграму за допомогою панелі інструментів Використовувати діаграму інцидентів (рис. 3.16).

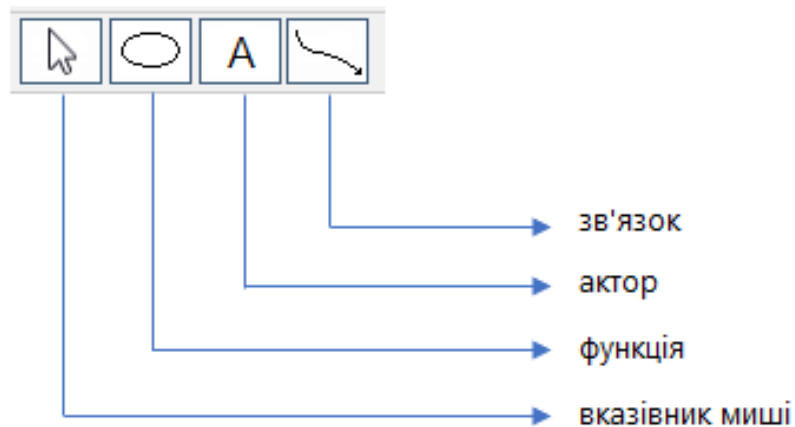


Рис. 3.16. Панель інструментів діаграми прецедентів

Редагування діаграми інцидентів використання схоже на редагування діаграм DFD, як описано вище.

3.2.4. Створення схеми послідовності

Діаграма послідовності може бути створена для будь-якої функції діаграми випадку використання з її контекстного меню (рис. 3.17).

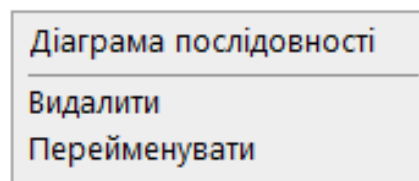


Рис. 3.17. Контекстне меню функції діаграми реєстру використання

У вікні, що відкриється (рис. 3.18), потрібно вибрати зі списку існуючих акторів, які будуть розміщені на схемі послідовності.

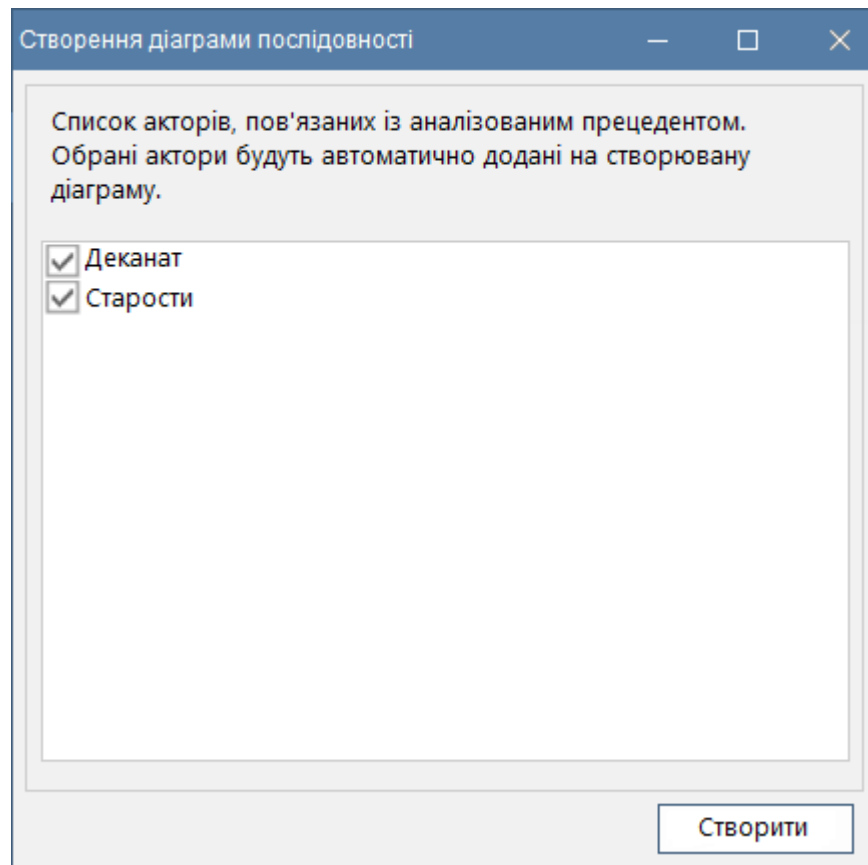


Рис. 3.18. Форма "Створення діаграми послідовності"

Після створення діаграми до дерева проектів додається відповідний вузол і активується панель інструментів для редагування цієї діаграми (рис. 3.19).

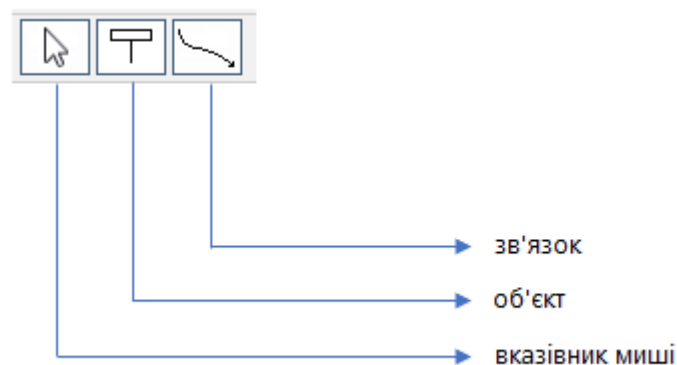


Рис. 3.19. Панель інструментів послідовної діаграми

Щоб додати об'єкт до схеми, просто натисніть на відповідну кнопку,

розташовану на панелі інструментів, при цьому об'єкт розміщується на формі автоматично після попереднього об'єкта і має невизначений тип (рис. 3.20).



Рис. 3.20. Типи об'єктів діаграми послідовності

Призначення об'єктів діаграми конкретних типів здійснюється через контекстне меню (рис. 3.21).

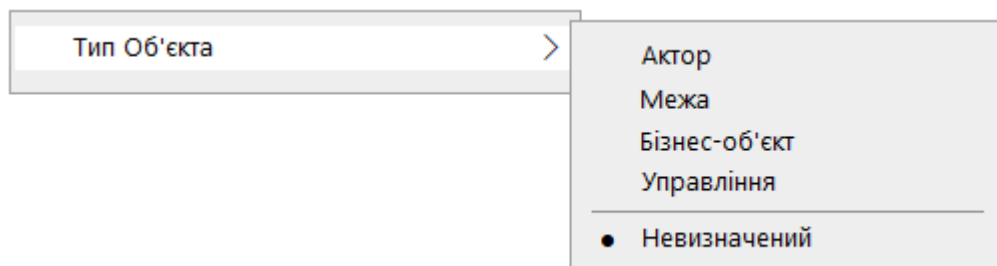


Рис. 3.21. Контекстне меню невизначеного об'єкта
діаграми послідовності

Користувачеві дозволяється міняти місцями об'єкти (перетягування лівою клавішею миші) і змінювати їх розмір. Коли ви змінюєте висоту об'єкта, висота всіх інших об'єктів змінюється.

Щоб зв'язати два об'єкти разом, необхідно встановити зв'язки між часовими осями цих об'єктів. Це схоже на зв'язок об'єктів відносинами на діаграмах і прецедентах DFD, але на відміну від них, на послідовних діаграмах зв'язок може починатися і закінчуватися на одному і тому ж об'єкті (рис. 3.22). Деякі об'єкти діаграми послідовностей (актори, бізнес-об'єкти та об'єкти керування) можна зіставити з класами, створеними в цьому проекті, за допомогою пункту Класу в

контекстному меню (рис. 3.23).

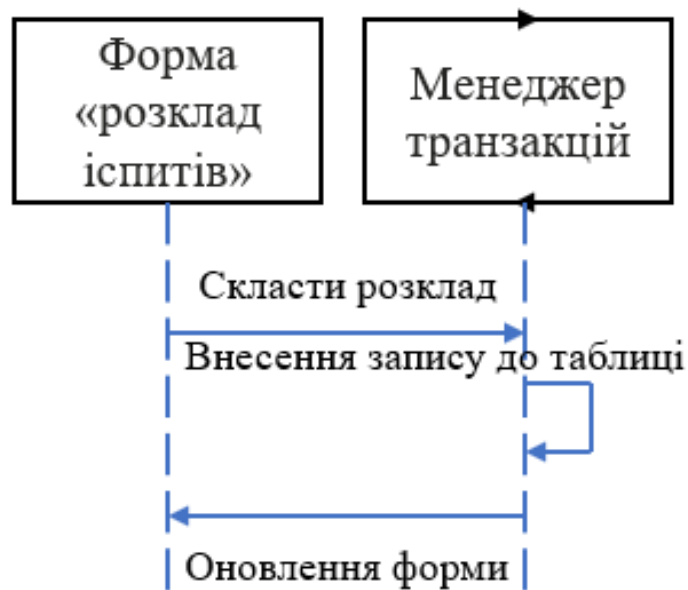


Рис. 3.22. Діаграма послідовності

Деяким об'єктам діаграми послідовності (акторам, бізнес-об'єктам та об'єктам управління) можна ставити у відповідність класи, створені в даному проекті. Це здійснюється через пункт класу контекстного меню (рис. 3.23).

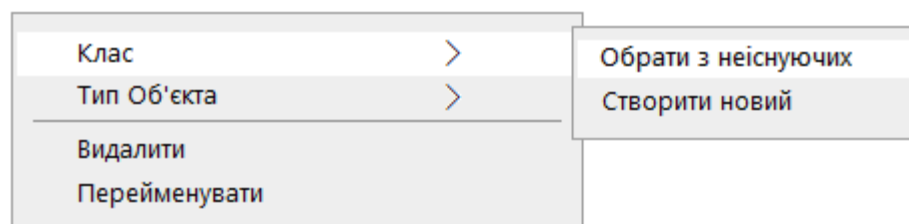


Рис. 3.23. Контекстне меню діаграми послідовностей

Якщо натиснути на пункт "Вибрати з існуючих", то відкривається вікно (рис. 3.24), що містить список усіх створених у проекті класи, що описують їх властивості і методи.

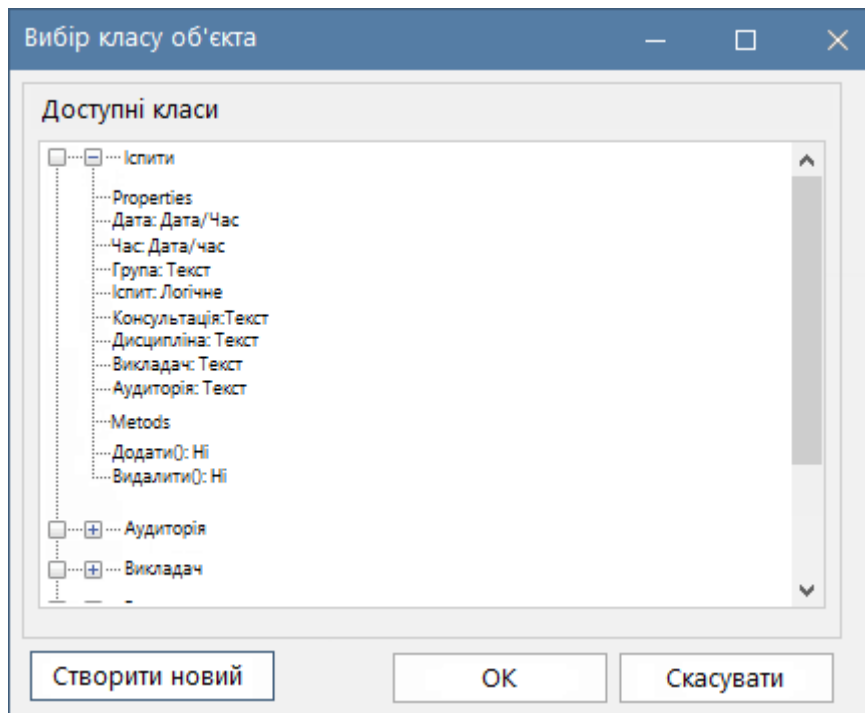


Рис. 3.24. Форма вибору класу об'єкта

При виборі «Створити нове» в контекстному меню (рис. 3.23) і в цьому вікні (рис. 3.24) ви переходите на діаграму класу для її редагування. Клас, присвоєний об'єкту, записується в його назві на схемі послідовності (рис. 3.25).

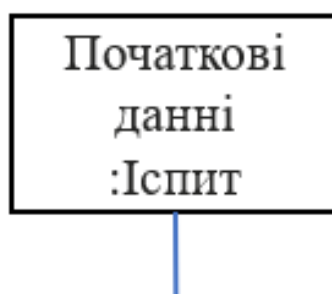


Рис. 3.25. Бізнес-об'єкт діаграми послідовності

Приклад діаграми послідовності для прецеденту планування іспитів наведено на рис. 3.26.

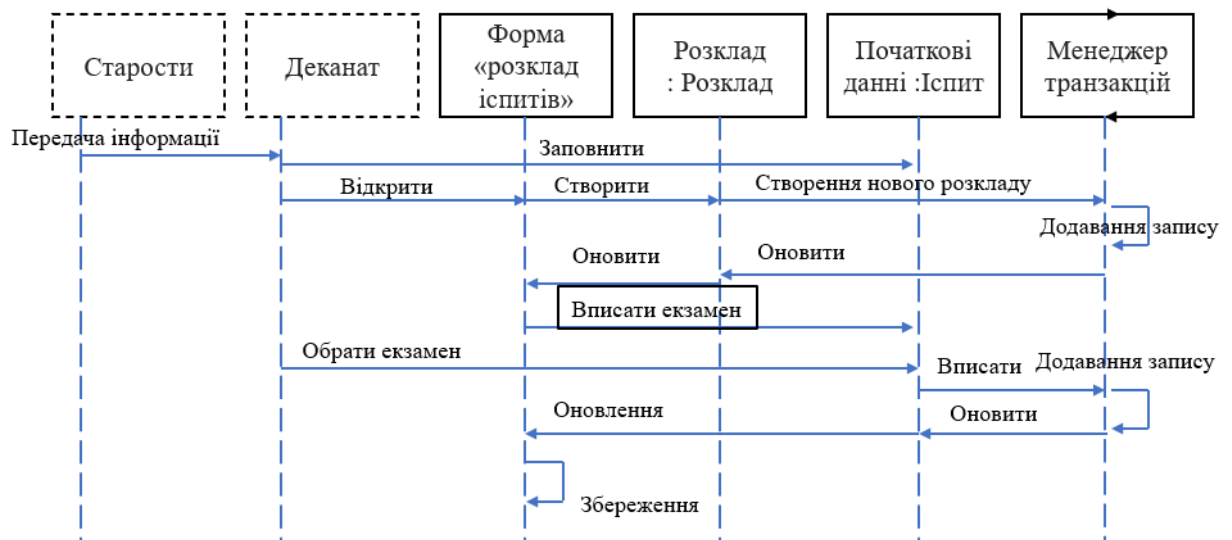


Рис. 3.26. Діаграма послідовності

Збереження проекту на сервері здійснюється через пункт головного меню «Зберегти проект» або за допомогою кнопки на стандартній панелі інструментів (рис. 3.13).

3.2.5. Створення діаграми класів

Щоб перейти до діаграми класів, необхідно вибрати відповідний пункт в головному меню або скористатися панеллю Інструмент. Ви редагуєте діаграму класів за допомогою панелі інструментів, показаної на рис. 3.27.

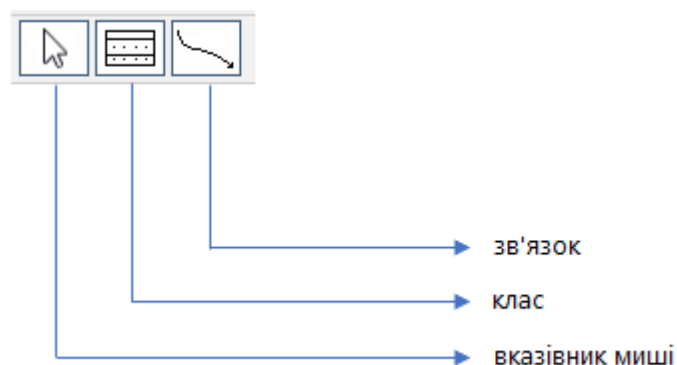


Рис. 3.27. Панель інструментів діаграми класів

Щоб помістити об'єкт в область діаграми, натисніть відповідну кнопку панелі інструментів і на діаграмі, де потрібно розмістити об'єкт.

Коли ви наводите курсор або клацнете об'єкт, він виділяється червоним. Ви можете змінити розмір об'єкта, утримуючи ліву клавішу миші за нижнім правим нижнім правим кутами об'єкта. Аналогічно, ви можете перемістити межу, яка розділяє властивості та методи класу, який ви редагуєте (рис. 3.28).

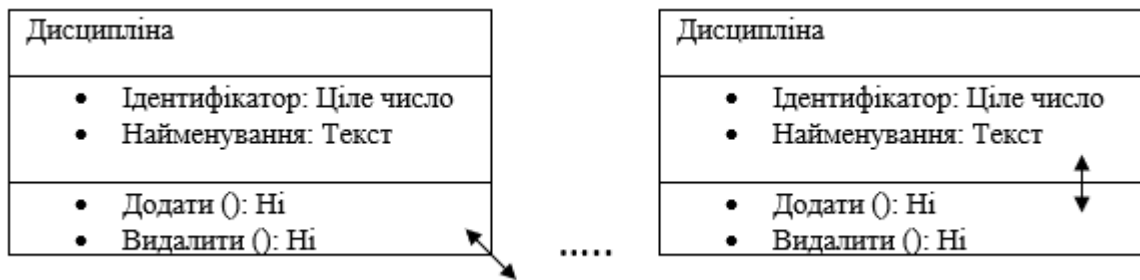


Рис. 3.28. Зміна розміру об'єкта

Якщо вибрати пункт Авторозмір контекстного меню (рис. 3.29), розмір класу та розташування розділювача межі встановлюються автоматично таким чином, щоб були видимі всі властивості та методи.

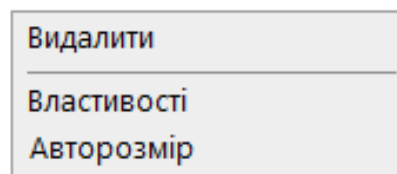


Рис. 3.29. Контекстне меню класу

Подвійне натискання по згенерованому класу або вибір пункту «Властивості» контекстного меню відкриває вікно для введення інформації про цей клас (рис. 3.30). Вікно містить 3 вкладки:

- Вкладка «Загальна»: вказати назву та опис класу;
- Вкладка «Властивості»: укажіть властивості класу з іменем, типом даних і типом доступу.

- Вкладка «Методи»: визначає методи класу, які визначають ім'я, тип повернення, тип доступу та вхідні параметри.

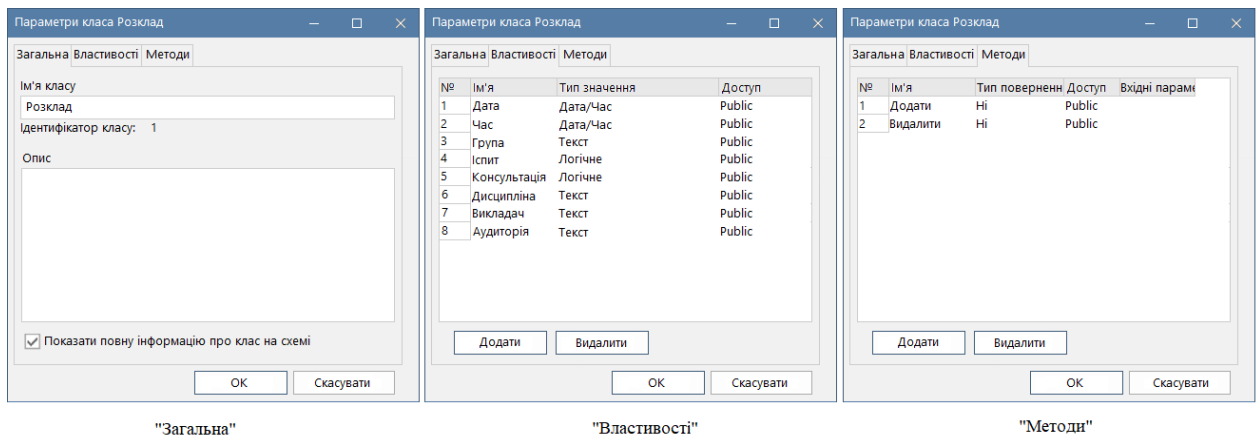


Рис. 3.30. Форма параметри класу

Щоб вказати список вхідних параметрів, двічі клацніть на відповідному полі вкладки «Методи» і після заповнення таблиці вхідних параметрів натисніть «Прийняти».

Щоб з'єднати класи з посиланнями, натисніть відповідну кнопку панелі інструментів. Коли ви наводите курсор на вказівник миші, класи (включаючи межі фігури), які можна з'єднати, виділяються синім кольором. Над межами форми та класу курсор також змінює свій вигляд, даючи зрозуміти, що потік можна намалювати звідси (рис. 3.31). В даний час підключені класи також виділяються синім кольором.

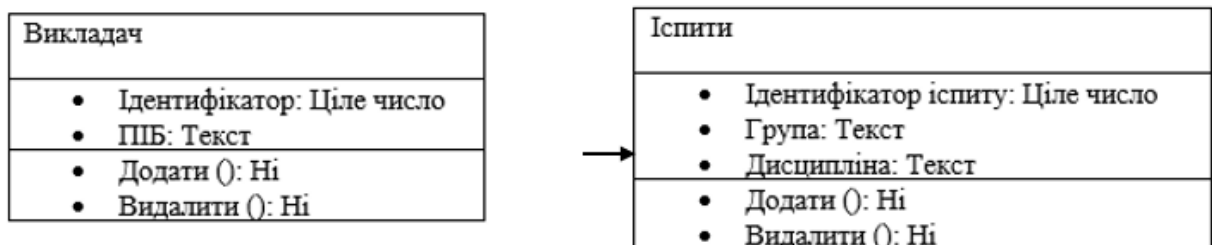


Рис. 3.31. З'єднання класів за посиланнями

Рух об'єктів – класів і заголовків посилань здійснюється за допомогою лівого натискання клавіш миші, при цьому всі з'єднання, підключені до цього об'єкта,

автоматично перебудовуються.

Редагування посилань здійснюється шляхом перетягування вузлів або відрізків ліній (рис. 3.32).

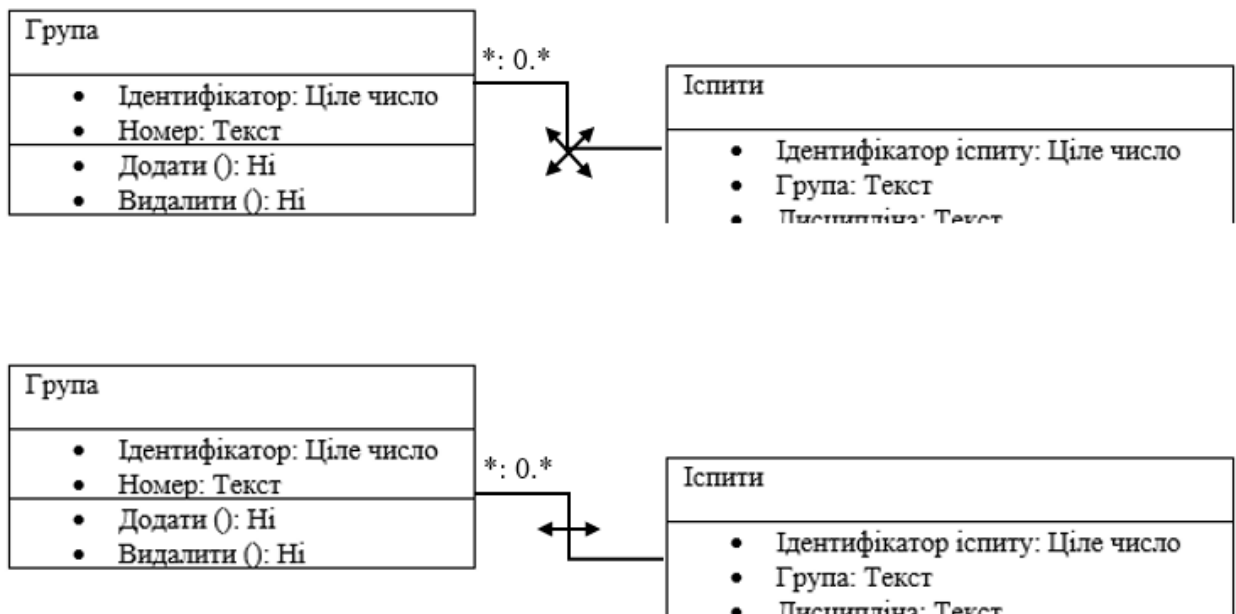


Рис. 3.32. Редагування потоків даних

Перейменування класу здійснюється у службовому вікні «Властивості» (рис. 3.30) на вкладці «Загальні», яка викликається подвійним натисканням по класу або через контекстне меню (рис. 3.29).

Перейменування посилань відкриває додаткове вікно (рис. 3.33), яке дозволяє вибрати ім'я зі списку потоків даних і зв'язків, які вже існують в цьому проекті.

The screenshot shows a dialog box titled 'Перейменування потоків даних'. It has a text input field at the top containing '*: 0..*'. Below it is a list box containing the following items:

- *: 0..*
- *: 1
- Аудиторний фонд
- Вписати іспит

At the bottom of the dialog box are two buttons: 'Прийняти' (Accept) and 'Скасувати' (Cancel).

Рис. 3.33. Форма "Перейменування потоку даних"

Щоб відобразити повну інформацію про клас на діаграмі, потрібно встановити відповідний прапорець у параметрах класу на вкладці Загальні.

Щоб видалити об'єкти, виберіть відповідний пункт у контекстному меню класу. Будь-які асоціації, пов'язані з цим класом, також будуть видалені.

Приклад діаграми класів наведено на рис. 3.34.

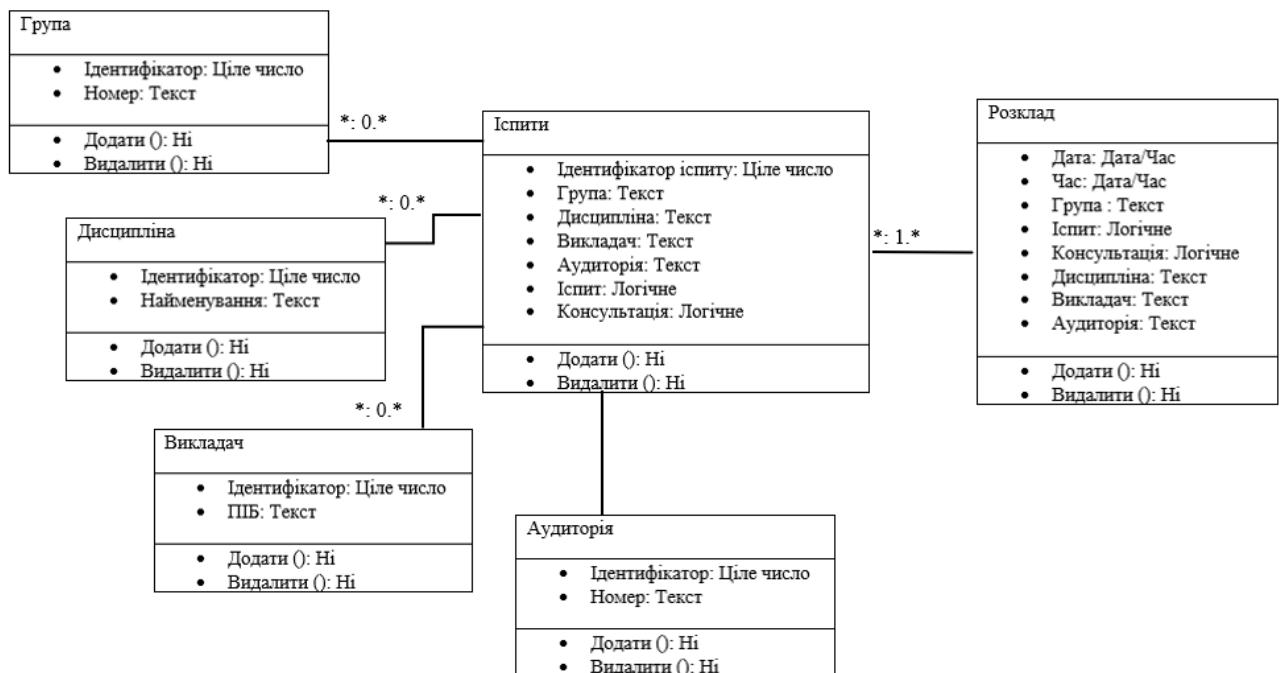


Рис. 3.34. Діаграма класу в "ЕАВР" для ІС "Розклад"

3.2.6. Підсистема моделювання інтерфейсу користувача

Ви можете моделювати інтерфейс користувача, якщо у вашому проекті є діаграма послідовності та об'єкт межі, який створюється на ньому. У контекстному меню об'єкта межі вибирається параметр Форма. Кожна межа може мати лише одну форму, тому натискання відповідного пункту меню відкриє або нову форму, або вже створену.

Менеджер користувацьких форм викликається з контекстного меню об'єкта – межі діаграми послідовності (рис. 3.35).

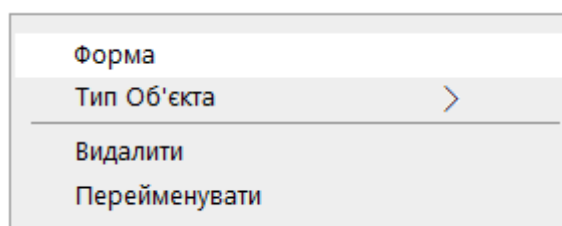


Рис. 3.35. Контекстне меню об'єкта – межі

Графічний інтерфейс менеджера форм показаний на рис. 3.36.

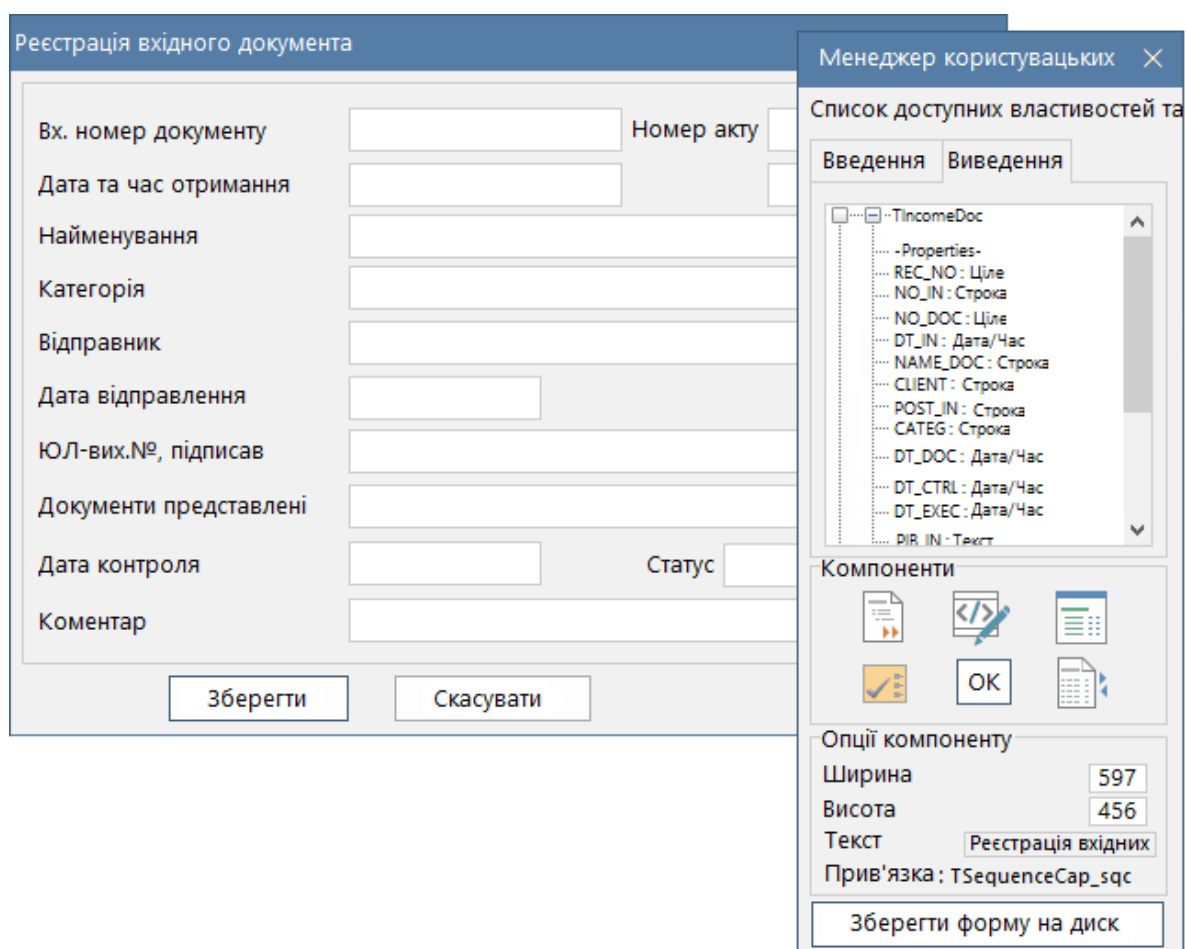








Рис. 3.36. Форма «Менеджер користувацьких форм» та прототип форми

Менеджер форм містить наступні елементи:

- Дерево вхідних і вихідних класів проектів зі списками доступних властивостей і методів цих класів;

- Панель візуальних компонентів, які користувач розміщує на формі, яку потрібно спроектувати. Розміщені компоненти можуть бути шести типів:

-  Property Label – відображає не редаговані однорядні текстові властивості класів вводу і виводу,
-  Edit – відображає редаговані однорядні властивості тексту класів вводу та виводу,
-  Memo – відображає багаторядкові текстові властивості класів вводу та виводу,
-  CheEnterprick Box – відображає логічні властивості класів вводу та виводу,
-  Button – кнопки, до яких прив'язані методи вводу і виведення класів,
-  String Grid – таблиці, поля яких можуть бути властивостями вихідних класів;

- Панель властивостей (опцій) компонентів, розміщених на формі, містить наступні елементи:

- Ширина і висота компонента. ви можете встановити ці параметри вручну або налаштувати їх з клавіатури за допомогою клавіш зі стрілками, утримуючи клавішу Shift
- Текст, що відображається в компоненті або заголовку,
- Прив'яжіться до певної властивості або методу вибраного класу.

Для компонентів String Grid цей параметр відсутній, його замінює кнопка виклику «Редактор таблиць» (рис. 3.37),

Якщо не вибрано жодного компонента, властивості форми, яку потрібно проектувати, відображаються в області параметрів;

- Натискання кнопки «Зберегти форму на диск» генерує файли *.cpr для

розробленої форми і зберігає їх на своєму комп'ютері в зазначеному місці.

Редактор таблиць (рис. 3.37) дозволяє регулювати порядок і заголовки полів таблиці. Панель інструментів редактора містить три кнопки: видалити поле, перемістити поле на одну позицію вгору та перемістити на одну позицію вниз.

Дані	Заголовок
Розклад.Дата	Дата
Розклад.Час	Час
Розклад.Група	Група
Розклад.Іспит	Іспит
Розклад.Консультація	Консультація
Розклад.Дисципліна	Дисципліна
Розклад.Викладач	Викладач
Розклад.Аудиторія	Аудиторія

Рис. 3.37. Форма редактора таблиць

Щоб розмістити елемент на формі, натисніть на обраний компонент на відповідній панелі **"Менеджер форм"** і на формі в потрібному місці. На даний момент обраний компонент виділений синім кольором, а його властивості відображаються на панелі параметрів (рис. 3.38).

Опції компоненту

Ширина: 597

Висота: 456

Текст: Button

Прив'язка: Ні

Рис. 3.38. Панель параметрів менеджера форм

Переміщення компонентів здійснюється мишею з лівою натиснутою клавішею або стрілками з клавіатури з натиснутою клавішею Ctrl. Розміри компонентів можна змінювати мишею через панель опцій або стрілками з клавіатури з натиснутою клавішею Shift. Видалення компонентів здійснюється через відповідний пункт контекстного меню.

Щоб прив'язати до компонента певну властивість чи метод, необхідно «перетягнути» цю властивість (метод) із дерева класів на відповідний компонент. При цьому, якщо дана прив'язка дозволена, то компонент виділяється червоним кольором (рис. 3.39).



Рис. 3.39. Прив'язка до кнопки методу «Додати» класу «Розклад»

Якщо ви хочете прив'язати всі властивості певного класу до компонента (таблиці), вам не потрібно прив'язувати кожен властивість окремо, просто "перетягніть" ім'я класу на компонент String Grid.

Приклад розробленої форми розкладу іспитів наведено на рис. 3.40.

Рис. 3.40. Форма планування, розроблена ЕАВР

3.3. Опис агента інтеграції EABP

Агент інтеграції EABP доступний користувачам системи EABP (меню General/Agents).

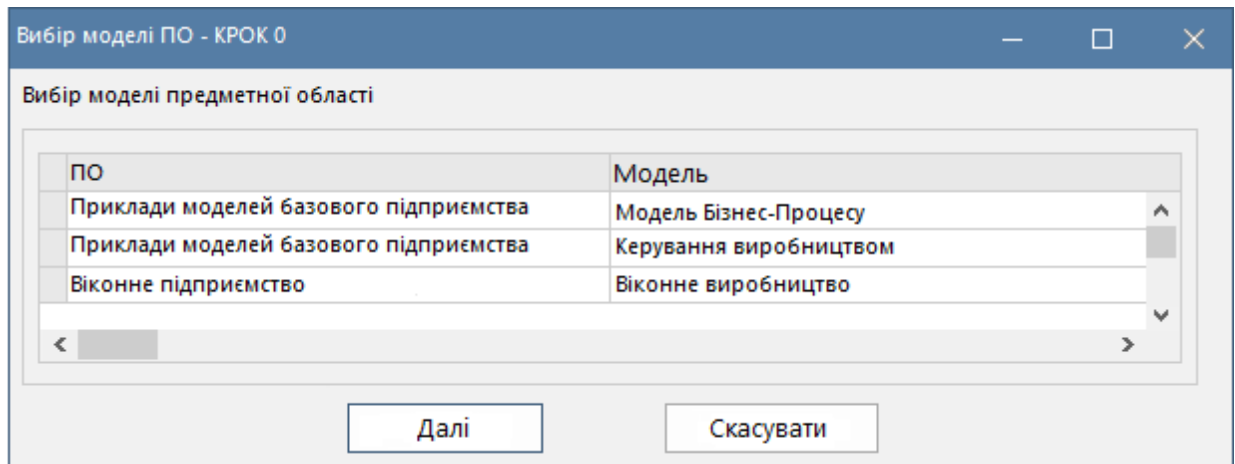


Рис. 3.41. Форма першого етапу перетворення моделі

Після цього пропонується вибрати операції моделі (рис. 3.42).

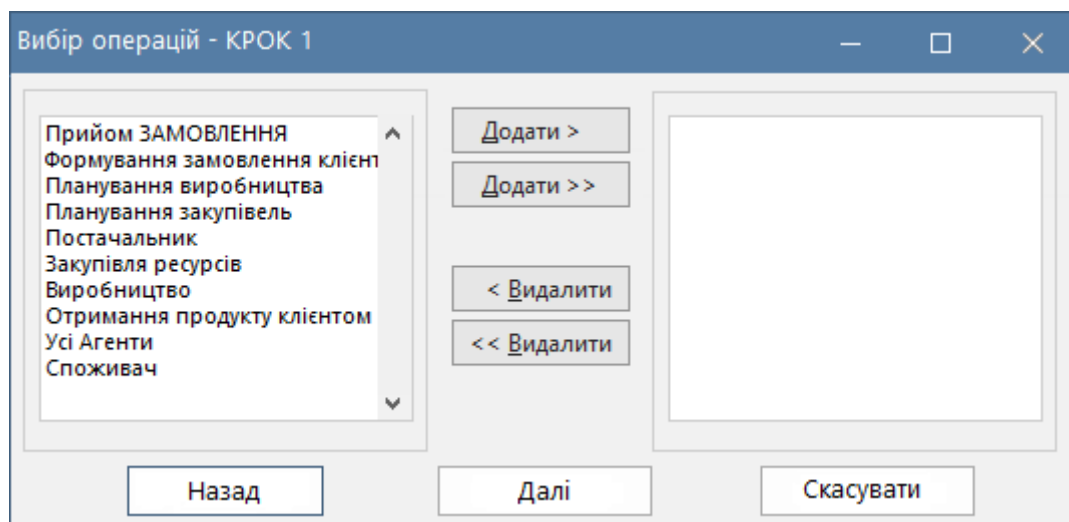


Рис. 3.42. Форма другого етапу перетворення моделі

На наступному кроці агент послідовно перетворює елементи моделі МППР у

відповідні елементи моделі ІС (див. табл. 2.11).

3.4. Методика використання модулю ЕАВР

У процесі бізнес–моделювання та проектування програмного забезпечення предметної області МППР використовуються наступний програмний модуль:

ЕАВР – мультиагентна система динамічного моделювання ситуацій з елементами інтелектуальної системи техніко–економічного проектування;

Методологія бізнес–моделювання та проектування програмного забезпечення предметної області МППР в стандарті IDEF0 показана на рис. 3.43.

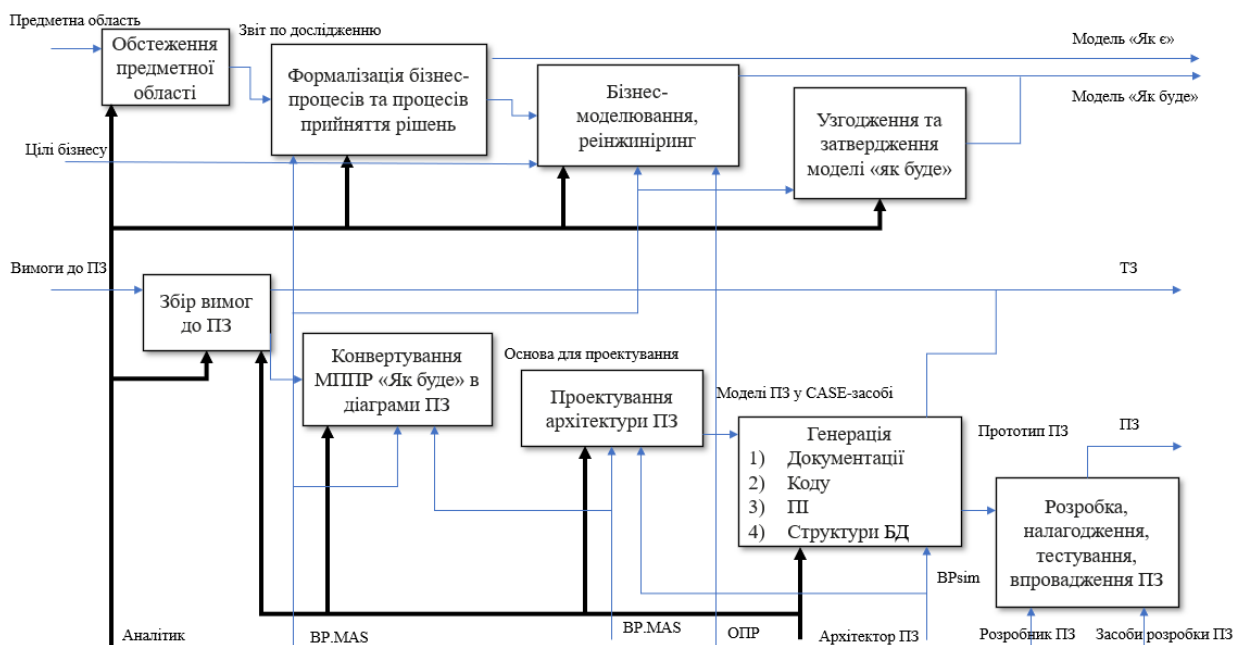


Рис. 3.43. Методологія бізнес–моделювання та проектування програмного забезпечення

За результатами опитування БП аналітик використовує ЕАВР для створення моделі БП підприємства «як було», проводить моделювання і при необхідності будує модель «як буде» Результати моделювання дозволяють обґрунтувати запропоновані зміни в бізнес–моделі. Отримана модель перетворюється в діаграми DFD ЕАВР модулю, враховуються тільки ті процеси, які будуть автоматизовані. Цей пакунок використовується для проектування ІС. На основі цих діаграм DFD

можна отримати UML–діаграми прецедентів, а також шаблони класів і послідовних діаграм, порожню структуру бази даних. Потім програмний архітектор будує діаграми послідовностей і класів, а розробник ІС – шаблони форм.

Програмна реалізація інтеграції ЕАВР можлива завдяки тому, що дані цих двох продуктів зберігаються в єдиній базі даних. Перетворення інформації з однієї системи в іншу здійснюється інтерактивними програмними агентами–помічниками.

Помічник агентів, що входять до модулю ЕАВР, виконують такі функції:

- здійснює передачу інформації між програмними продуктами в рамках комплексного вирішення єдиного завдання;
- полегшує роботу користувача, що не програмує, в процесі ознайомлення з продуктами сімейства ЕАВР;
- реалізує функцію валідації на етапах створення моделі МППР, проектування МППР ІС.

Діаграма DFD, що показує послідовність дій агента–помічника при перетворенні моделі МППР в модель ІС, представлена на рис. 3.44 [72, 73].

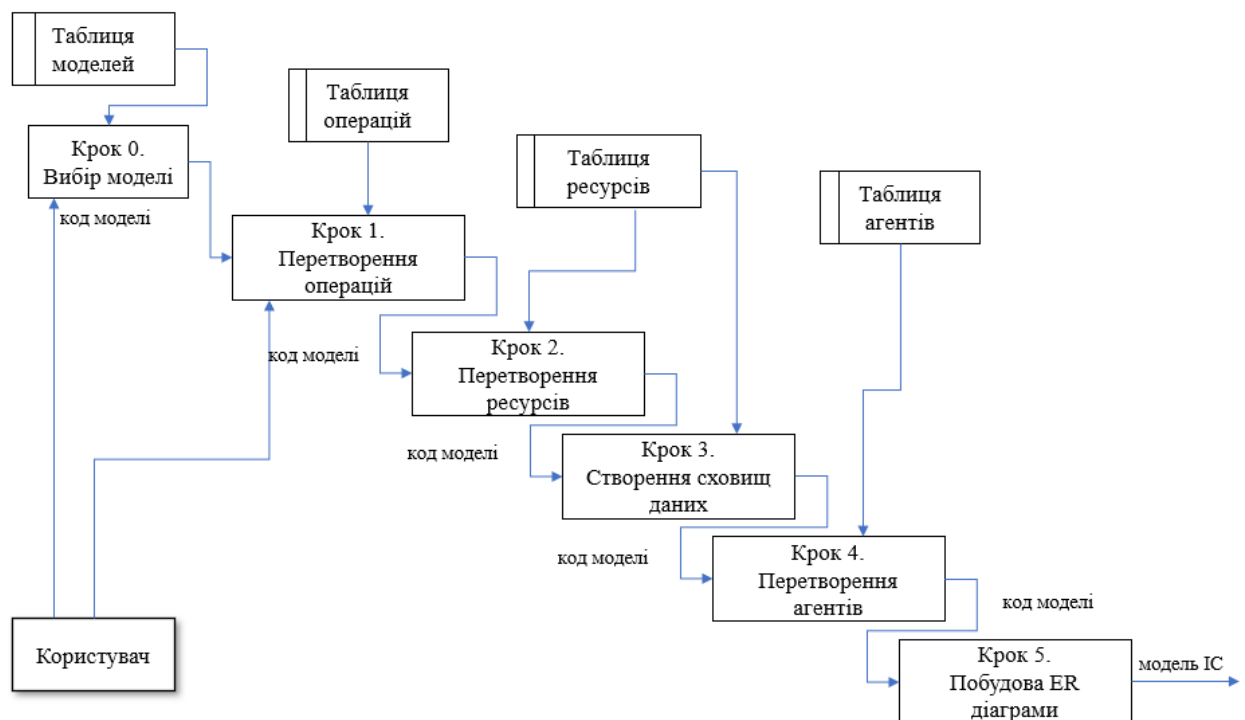


Рис. 3.44. DFD–діаграма перетворення моделі МППР в модель ІС

Висновки до розділу 3

Вирішення проблеми інтеграції моделювання, експертного, ситуаційного та мультиагентного моделювання, а також функціонального та об'єктно-орієнтованого підходу, дозволило впровадити систему підтримки прийняття рішень у сфері розробки інформаційної системи ЕАВР.

У третьому розділі наведено детальний опис можливостей ЕАВР: функціональна та об'єктно орієнтована розробка інформаційної системи, моделювання програмного інтерфейсу. Надано методологію використання модулю ЕАВР та описує агента-помічника для перетворення моделі мультиагентних процесів перетворення ресурсів в модель інформаційної системи.

В рамках процесу моделювання архітектури інформаційної системи модуль ЕАВР пропонує користувачеві наступні функції:

- опис бізнес-процесів, автоматизованих розробленими інформаційними системами за допомогою схем стандарту DFD.
- опис функцій, що виконуються користувачами системи в рамках автоматизованих процесів, з використанням схем кейсів використання. Є можливість створювати необмежену кількість діаграм інцидентів використання для кожної діаграми DFD.
- для кожного випадку використання, виконаного користувачем, може бути дано опис послідовності елементарних
- створення діаграми класів і діаграму послідовності карт до класів на цій діаграмі.
- проектування візуальних форм змодельованого програмного забезпечення: розміщення компонентів на формі, методи прив'язки та властивості класу до компонентів, а також збереження моделей форм що передаються програмісту для імпорту в середовище для подальшого розвитку алгоритмів.
- створення звітів про створений проект із зображеннями розроблених схем і форм.
- збереження проекту на сервері та завантаження з MS SQL Server для

редагування.

Перевага методу інтеграції програмних продуктів за допомогою агентів–помічників полягає у можливості для непрограмуючого користувача комплексно вирішувати завдання моделювання бізнесу та розробки інформаційної системи. Крім того, інтеграція дає можливість суттєво спростити та прискорити роботу аналітиків та проектувальників.

Дана методика та програмний модуль, розроблений на базі програмного комплексу Sparx Systems дозволяють комплексно вирішувати завдання моделювання бізнесу, техніко–економічного проектування, моделювання архітектури інформаційної системи та розробка прототипу додатку, що у результаті дозволяє суттєво спростити та прискорити роботу аналітиків, а також підвищити якість готового програмного забезпечення за рахунок автоматизації деяких етапів та скорочення впливу людського фактору.

РОЗДІЛ 4 ЗАСТОСУВАННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЕАВР ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ

В даному розділі буде показана можливість розробки інформаційних систем за допомогою розробленого програмного засобу ЕАВР, на прикладі розробки та впровадження модулів «Трансфер–агентський обмін» та «Рух контингенту» для електронно-інформаційної системи електронного документообігу, також буде проведена оцінка ефективності впровадження модулів і виконана реалізація цих модулів у додатках аналогах, з порівняльним аналізом оцінки ефективності розробленого методу.

4.1. Дослідження можливості застосування модулю ЕАВР для розробки проекту з аналізу бізнес–процесів та розробки технічного завдання на єдину інформаційну систему

Державне підприємство (ДП) є підприємством зі складною організаційною структурою (факультети, кафедри, філії та представництва), тому для успішної автоматизації бізнес–процесів ДП «Одеський науково–дослідний інститут зв’язку» необхідно було провести повне обстеження того, як вони протікають. На початку проекту було проведено інтерв'ю зі співробітниками та представниками керівництва ДП. Було проведено інтерв'ю проректорів, співробітників та керівників деканату, вчених секретарів, співробітників відділів інформаційних технологій та співробітників підрозділів з розвитку автоматизованих систем управління (АСУ) [96].

Аналіз показав, що навчальний процес частково автоматизований, але існуючі ІС не завжди пов'язані один з одним, містять дублюючу інформацію. Навчальний процес пов'язаний із підготовкою великої кількості первинної документації та звітів, у підготовці одного типу даних (стипендія, накази тощо) бере участь велика кількість структурних підрозділів ДП [96].

Результати обстеження навчального процесу із використанням структурного та об'єктно-орієнтованого аналізу виконувались у рамках НДР. В результаті аналізу бізнес процесів за допомогою CASE-засобу EABP було збудовано IDEF0 діаграм.

Обстеження бізнес процесів виявило не оптимальність виконання деяких процесів «Одеський науково-дослідний інститут зв'язку», зокрема «Хід сесії» та «Рух контингенту». Розглянемо докладніше кожен із пакетів [96].

4.1.1. Процес «Хід сесії»

Процес «Хід сесії» включає формування екзаменаційних та залікових відомостей, видачу та реєстрацію залікових та екзаменаційних листів, прийом іспитів та заліків у сесію та перездачу, введення результатів заліків у заліковий тиждень, іспитів у сесію та перездачу у сесію та поза сесію звітів щодо атестації студентів. У процесі «Ходу сесії» задіяні співробітники деканату, відділ автоматизованих систем управління, навчально-методичний відділ (НМВ), планово-фінансовий відділ (ПФВ). Основне навантаження лягає на співробітників деканату, які запроваджують вихідну інформацію та формують звіти за підсумками сесії. Роботу пов'язану з формуванням екзаменаційних та залікових відомостей та контролем підсумків сесії, виконує підрозділ АСУ. Повторне складання заліків та іспитів здійснюється за екзаменаційними та заліковими листами, які видають заступники деканів. Відділи НМВ та ПФВ надають початкову інформацію – навчальні плани, стипендіальний фонд для розподілу стипендії.

Процес «Хід сесії» супроводжується підготовкою великої кількості паперових документів, що, у свою чергу, сильно уповільнює поширення інформації до зацікавлених осіб. Виявлені проблеми дозволили сформулювати вимоги до автоматизації процесу "Хід сесії".

Було запропоновано внести зміни у процес роботи співробітників деканату (рис. 4.1): паперове заповнення семестрових журналів замінюється введенням даних до електронної версії семестрового журналу (аналогічно протоколи

4.1.2. Процес «Рух контингенту»

Процес «Рух контингенту» включає формування виписок до наказу, спискових наказів, збірних наказів з «Одеський науково–дослідний інститут зв’язку», формування різних довідок («Довідка студенту», «Довідка про те, що є студентом», «Довідка про стипендію» тощо). Основними учасниками цього процесу є працівники деканатів та деканат, декани факультетів, проректор. Процес був організований таким чином, що з моменту отримання співробітником деканату первинних документів до моменту зміни інформації про студента в БД та отримання витягу з наказу тривало багато часу.

Крім того, учасники процесу виконували багато рутинної роботи:

- співробітники деканату формували в різний час два схожі за змістом документи – заяву (допомагали написати студенту) та виписку до наказу;
- проректор три рази переглядав та візував різні документи про студента (заяву, виписку до наказу, наказ).

В результаті аналізу було запропоновано такі зміни (рисунки 4.2–4.5):

- під час написання заяви студентом співробітник деканату створює його електронний аналог у єдиній БД;
- пропонується відмовитися від формування виписки до наказу, вся необхідна інформація для видання наказу заноситься в єдину БД, отже збір віз проводитиметься один раз;
- співробітники деканат працюватимуть з електронною формою заяви і зможуть самі її коригувати, тобто. не буде гаяти час на відправлення документів до деканату для переоформлення;
- з'явиться можливість формувати та зберігати справи студентів в електронному вигляді (історія по студенту);
- у проректора підписуватиметься лише заява (для деяких видів заяв достатньо візи декана) та наказ щодо «Одеський науково–дослідний інститут зв’язку»;

- в інформаційній системі можна буде відстежувати маршрут та стадії проходження документів по процесу;
- у співробітників підрозділу АСУ зникне не властива їм функція – виконання проводок наказу в базі даних; у разі облікового наказу, відповідні інформаційні об'єкти в інформаційній системі формує співробітник деканату, надалі з ними працює співробітник деканат.

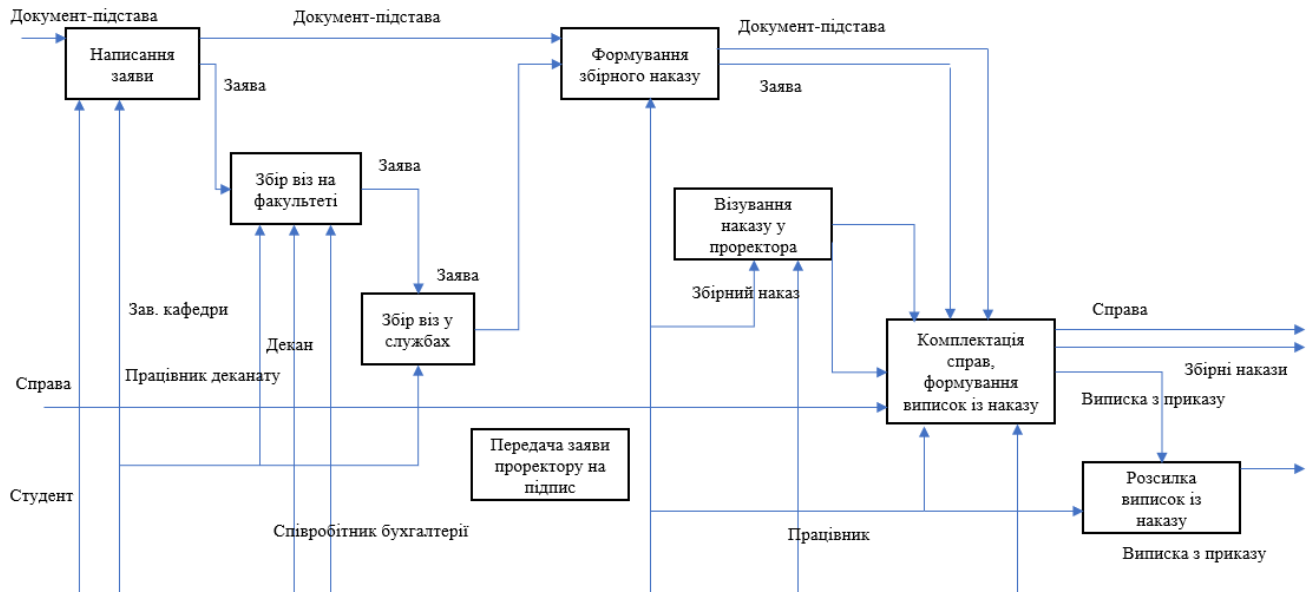


Рис. 4.2. Модель «як буде» процесу «Рух контингенту»

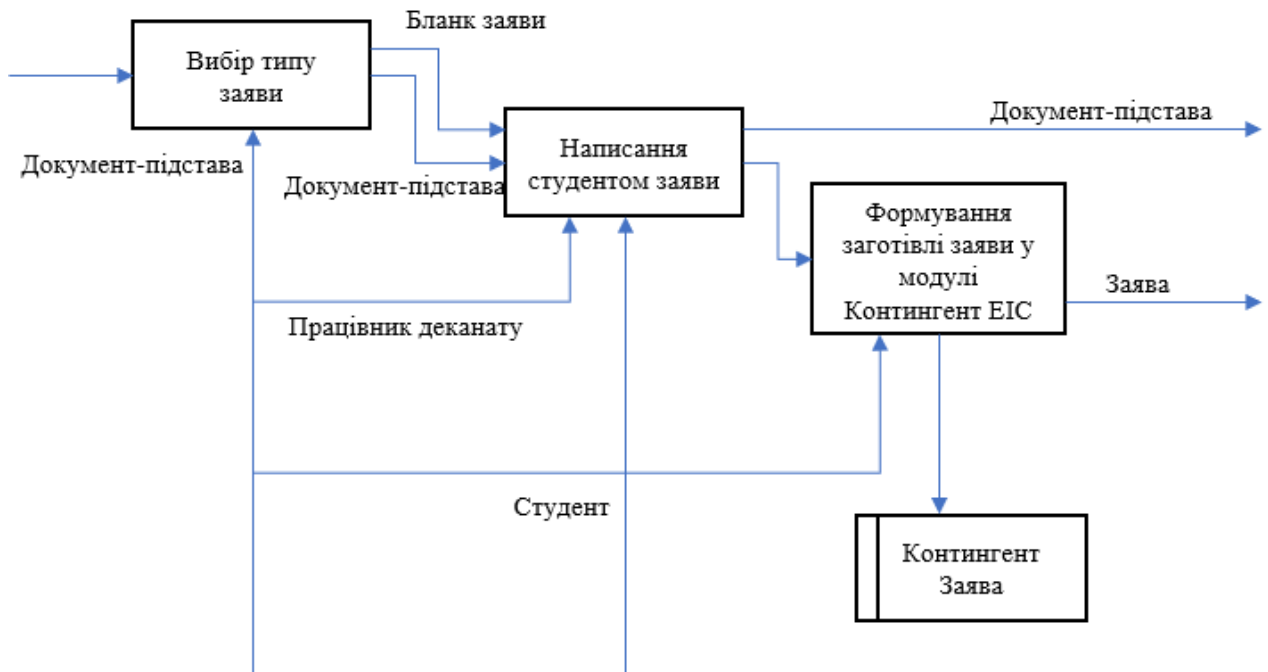


Рис. 4.3. Написання заяви



Рис. 4.4. Формування збірного наказу у деканат

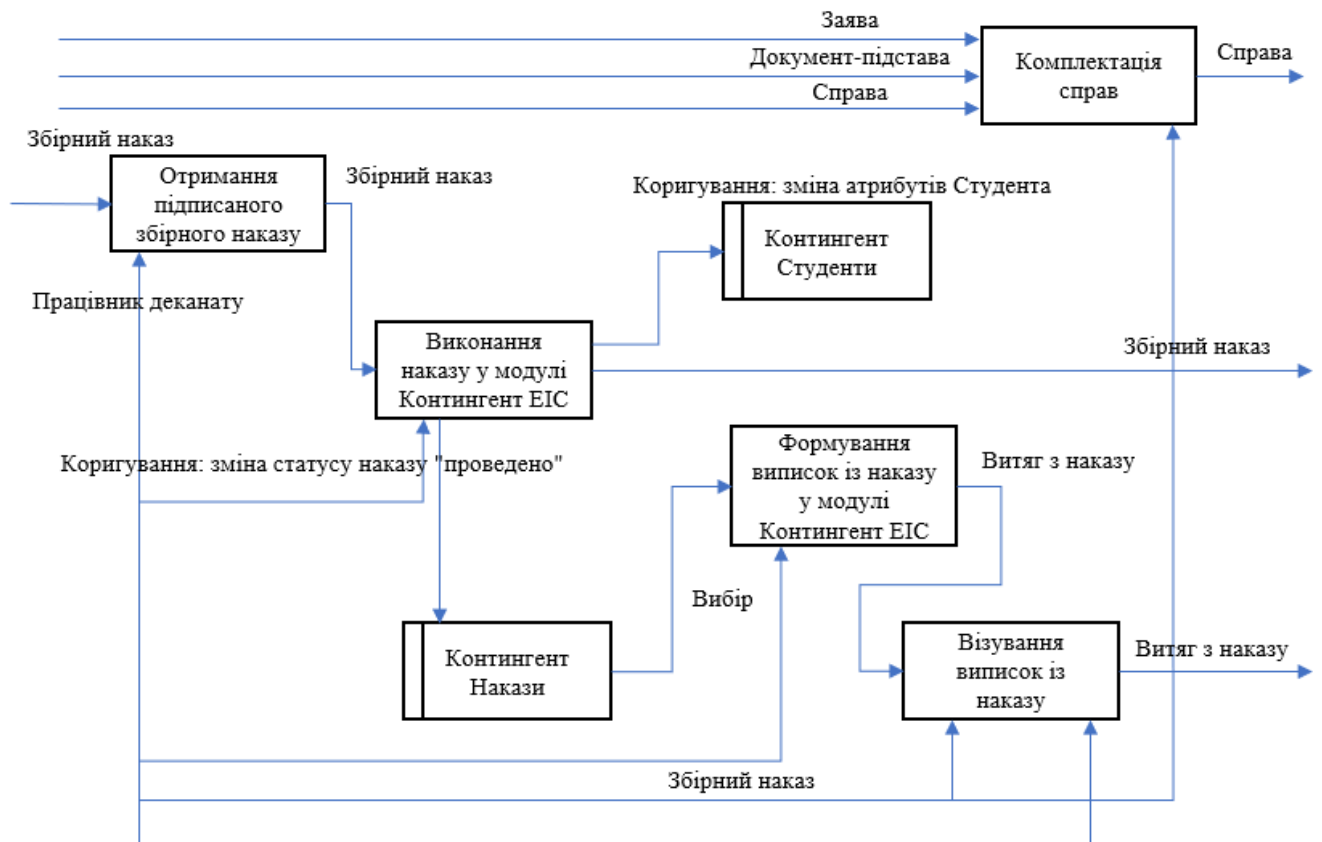


Рис. 4.5. Комплектація справ, формування виписок із наказу

Таким чином, автоматизація процесу «Рух контингенту» (модель «як буде») дозволить скоротити тимчасові трудові витрати співробітників ДП та оперативно отримувати достовірну інформацію про студентів.

За результатами обстеження було написано ТЗ на розробку ЕІС ДП. Воно включає вимоги і діаграми, що описують архітектуру системи. За угодою із Замовником використовувалися DFD–діаграми, UML–діаграми прецедентів, послідовності та класів. ТЗ містить 24 DFD–діаграми, 13 діаграм прецедентів, 19 діаграм послідовності та 1 діаграму класів. Наприклад, функціональність модуля Розклад ЕІС ДП показана на рис. 4.6.

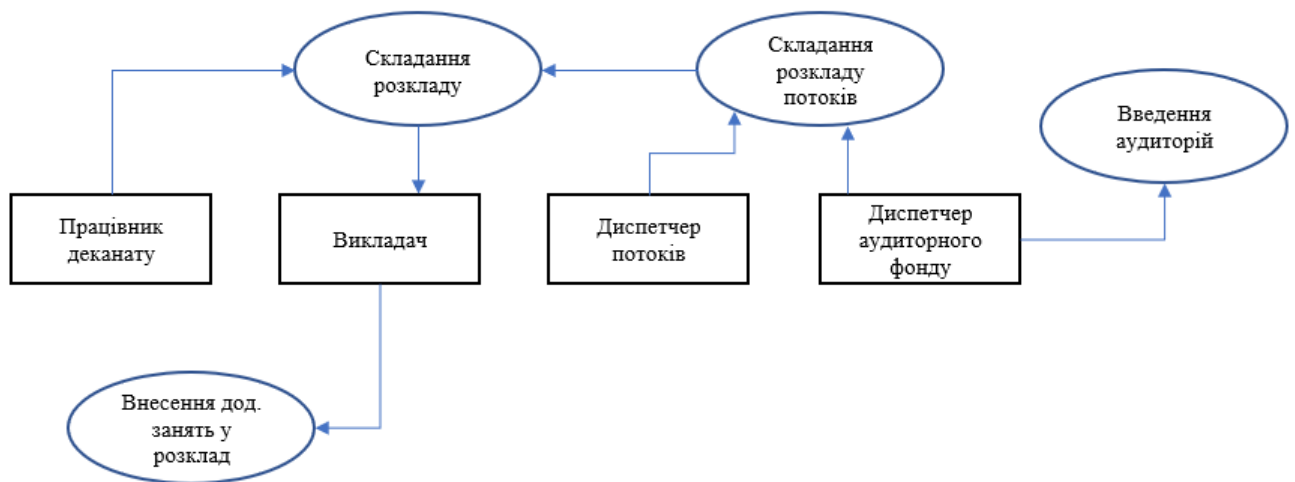


Рис. 4.6. Діаграма варіантів використання модуля Розклад

Послідовність дій для прецеденту «Упорядкування розкладу навчальних занять/іспитів» показана за допомогою діаграми послідовності на рис. 4.7.

ЕАВР дозволяє на підставі даних про бізнес–об'єкти інформаційної системи здійснювати проектування прототипів екранних форм проектованої системи, які також були включені до технічного завдання. На рис. 4.8 наведено приклад спроектованої форми введення розкладу.

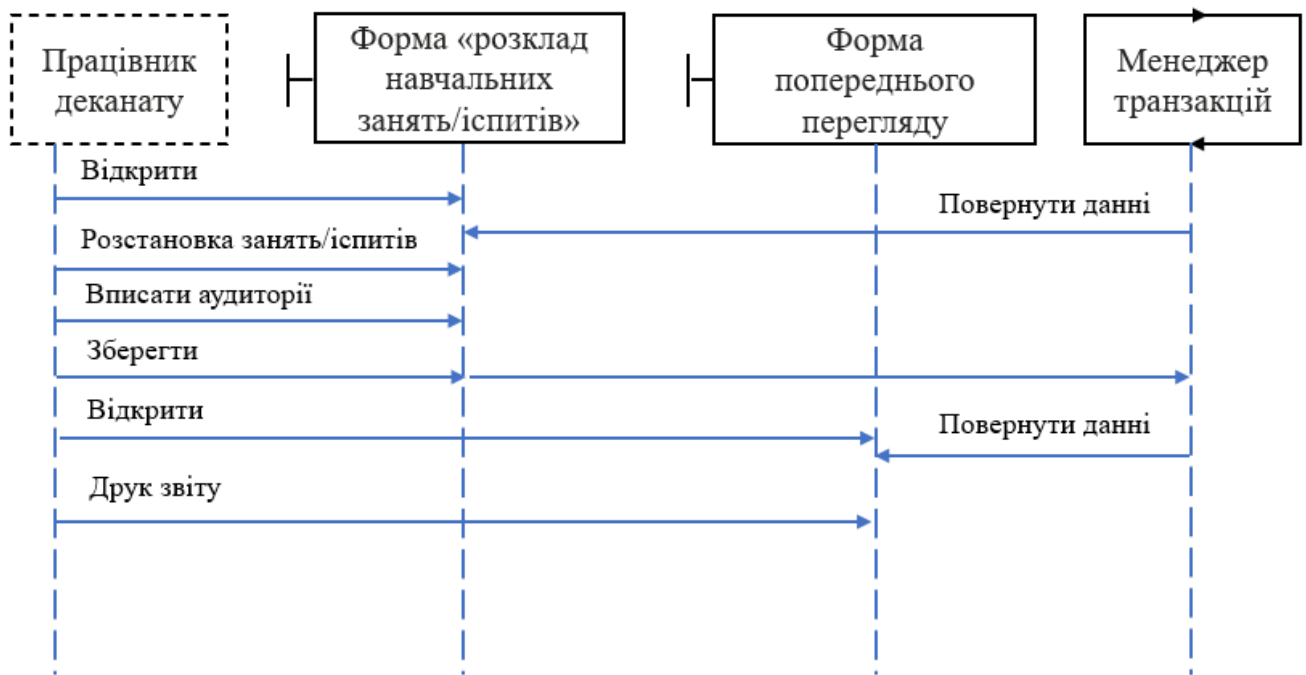


Рис. 4.7. Діаграма послідовності для прецеденту «Складання розкладу навчальних занять/іспитів»

Рис. 4.8. Спроектована форма введення розкладу іспитів у ЕАВР

4.1.3. Оцінка ефективності впровадження модуля «Рух контингенту»

Для оцінки ефективності впровадження модуля «Рух контингенту» ЕІС університету було побудовано імітаційну модель процесу руху контингенту «як було» і «як буде» у системі динамічного моделювання ситуацій (СДМС) ЕАВР. Дані для моделі були одержані в результаті опитування співробітників деканат, підрозділу автоматизованої системи керування та чотирьох деканатів, які півроку опрацювали з новим модулем ЕІС.

На рис. 4.9 показано модель процесу руху контингенту «як було» та «як буде». За допомогою перемикача, розташованого праворуч (рис. 4.9), можна вибрати варіант моделі, що програється: моделі «як було» або моделі «як буде», який визначає час роботи кожного вузла.

Вузли моделі є етапами обробки документів. Вузли 1–10 моделюють обробку документа на факультетах, вузли 11–14 – обробку в деканаті накопичених за тиждень документів, підготовлених факультетами.

Внаслідок автоматизації процесу руху контингенту зникла необхідність проходження документом деяких етапів обробки, тому при виборі моделі «як буде» в імітації не беруть участь вузли з номерами 7, 9, 10, 15 (рис. 4.9).

Потім була промодельована робота деканатів та деканат протягом місяця (160 год) у двох варіантах: до та після впровадження модуля «Рух контингенту». Результати проведення експериментів подано у вигляді графіків.

Графік «Кількість оброблених документів» (рис. 4.10) показує максимальну за заданих умов кількість документів, що пройшли всі етапи від написання заяви студентом, до розсилки виписки з наказу працівниками деканат.

Графік "Накопичення документів для пакету" (рис. 4.11) відображає процес накопичення за тиждень на факультетах документів, з яких у деканат формується пакет.

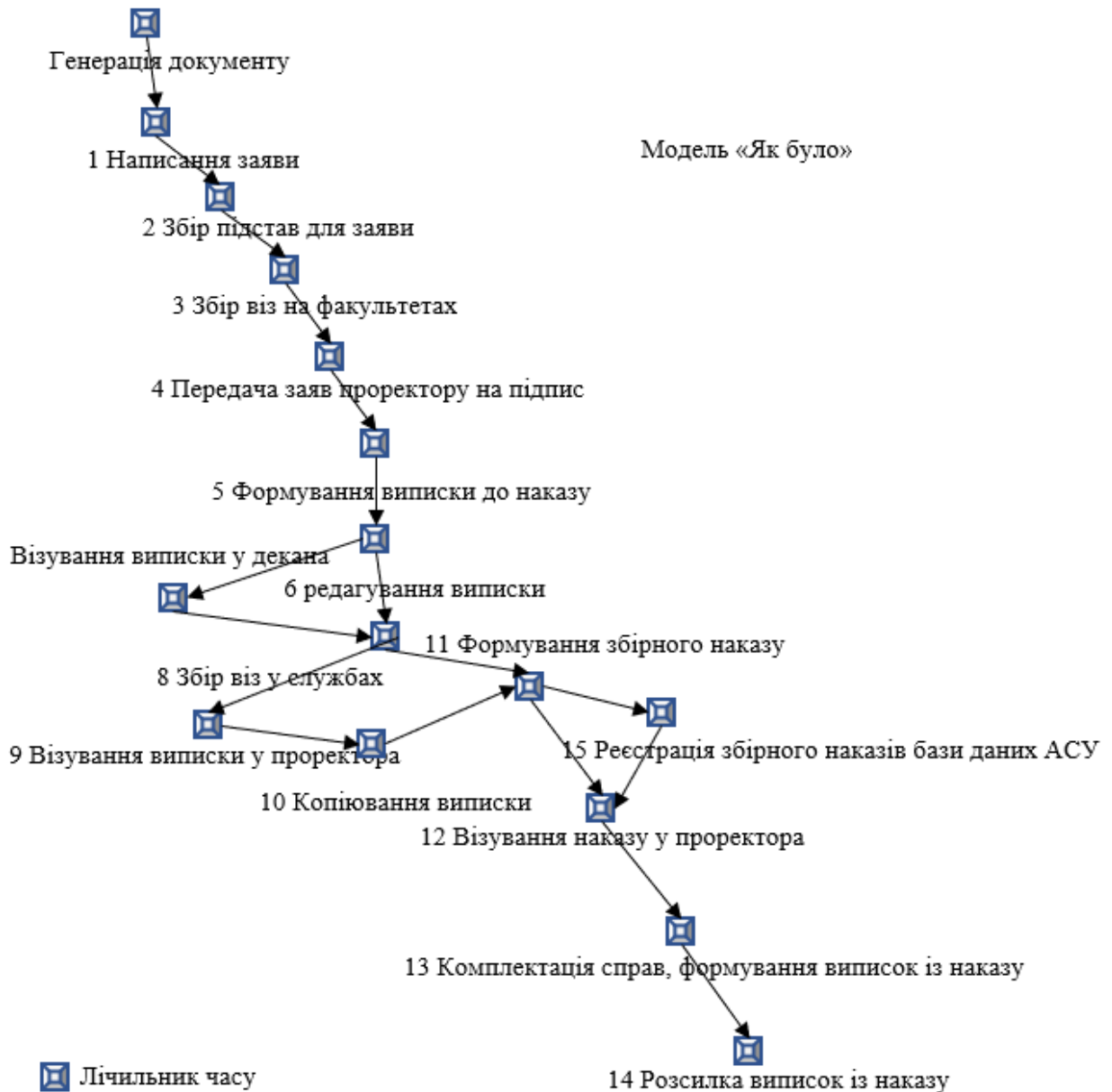


Рис. 4.9. Модель процесу рух контингенту

Графік "Накопичення документів для пакету" (рис. 4.12) відображає процес накопичення за тиждень на факультетах документів, з яких у деканат формується пакет (збірний наказ).

Впровадження модуля «Рух контингенту» дозволило уникнути втрати документів у підрозділі АСУ.

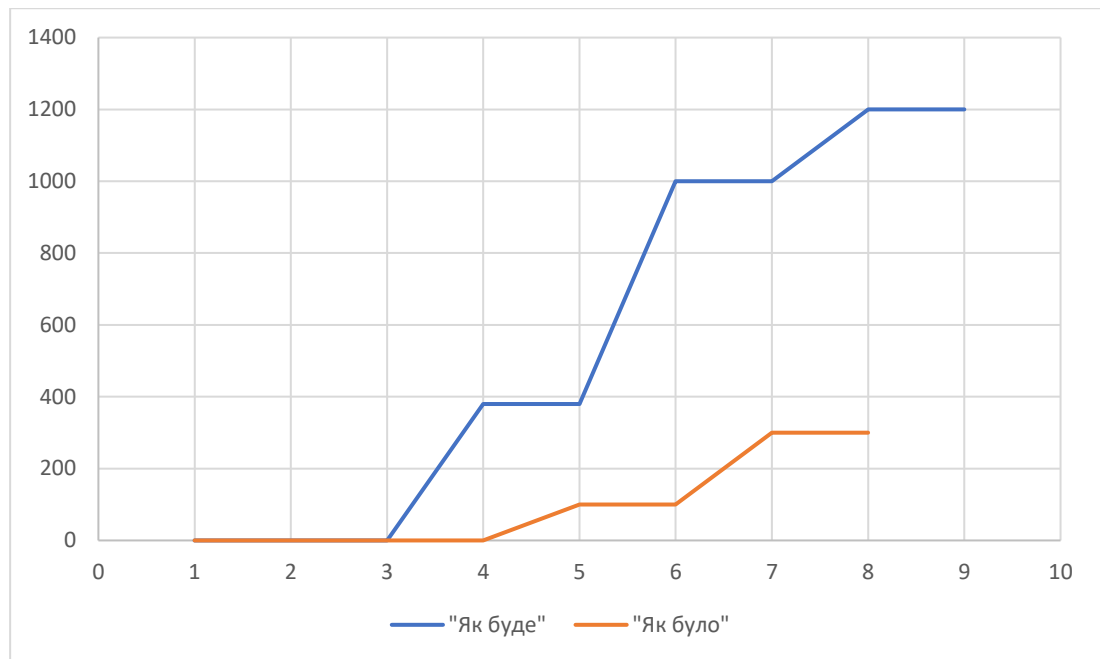


Рис. 4.10. Кількість оброблених документів за період

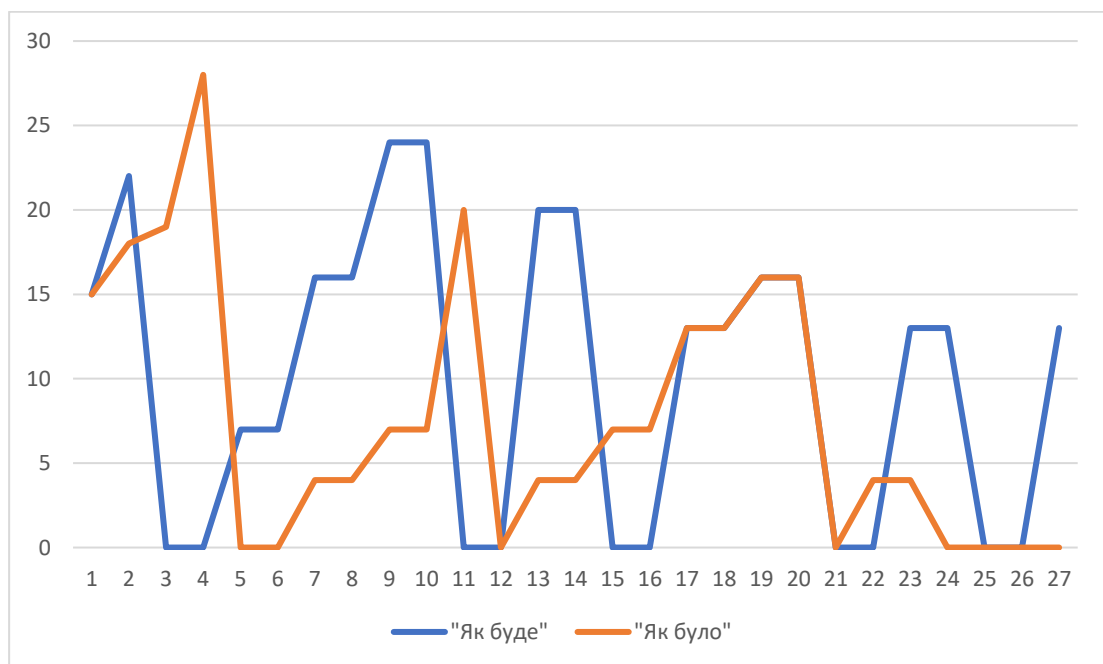


Рис. 4.11. Накопичення документів для пакету

Узагальнені результати експериментів наведено у табл. 4.1.

Оцінки ефективності впровадження модуля «Рух контингенту» ЕІС університет

Показники	Модель «Як було»	Модель «Як буде»
Кількість оброблюваних документів протягом місяця, шт.	390	1270
Кількість втрачених документів протягом місяця, прим.	12	0
Продуктивність праці працівників деканату, документ/год	0,4	0,5
Продуктивність праці співробітників, документ/год	2,4	7,9

Таким чином, завдяки автоматизації процесу «Рух контингенту» продуктивність співробітників деканату підвищилася на 26%.

4.2. Розробка додаткових модулів для системи ведення реєстрів документів «Planaria»

СВР «Planaria» призначена для ведення реєстрів документів організації. Для аналізу діяльності реєстратора було побудовано імітаційну модель у СДМС ЕАВР. Було проведено імітаційні експерименти визначення оптимальної кількості операторів, котрі займаються реєстрацією вхідних документів. Для перевірки роботи запропонованого методу було виконано роботу зі створення частини ІС, відповідальної реєстрацію вхідних документів.

4.2.1. Реєстрація вхідних документів

Модель процесу реєстрації вхідних документів показана рис. 4.12. Далі вона була конвертована в ЕАВР, в результаті процес реєстрації вхідних документів був представлений DFD–діаграмою (рис. 4.13).



Рис. 4.12. Модель процесу реєстрації вхідних документів

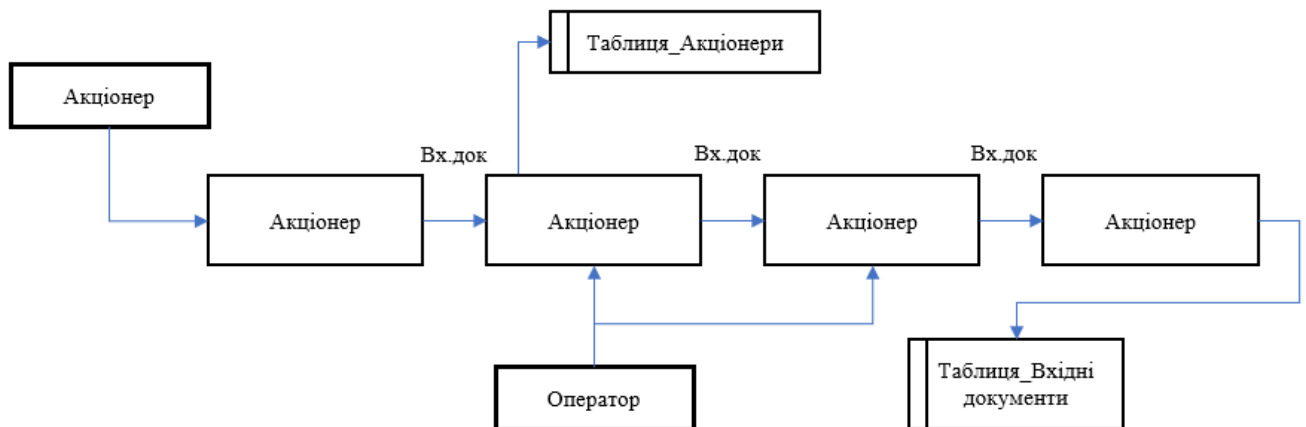


Рис. 4.13. DFD–діаграма процесу реєстрації вхідного документа

Також було отримано діаграму варіантів використання підсистеми «Реєстрація вхідних документів» (рис. 4.14) та діаграма основних класів (рис. 4.15).

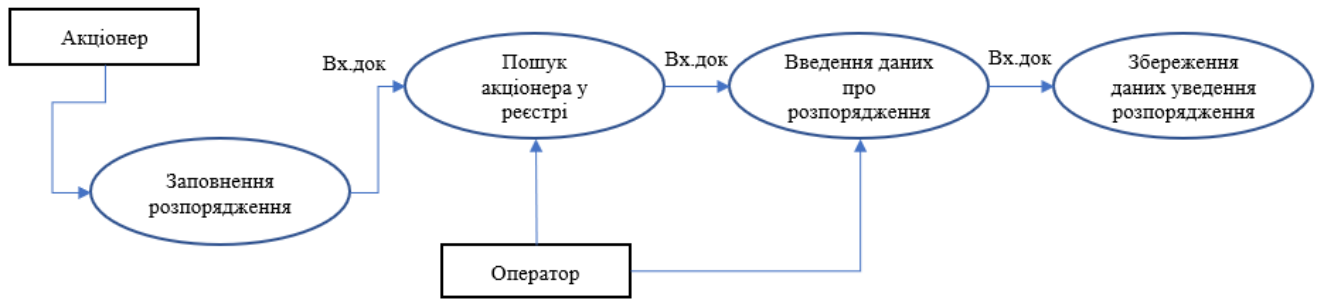


Рис. 4.14. UML–діаграма прецедентів процесу реєстрації вхідного документа

Далі було побудовано діаграму послідовності реєстрації вхідного документа (рис. 4.16) та діалогову форму (рис. 4.17).

TfrmIncDoc	TIncomeDoc	TPerson
<ul style="list-style-type: none"> • FRMNAME • CAPTION • btnOk • btnCancel • nedNo_IN • edNo_Doc • edDT_IN • edDt_Doc • edTime • edName_doc • edClient • edPost_inf • edCateg • edDt_doc • edDt_Ctrl • edDt_exec • edFio_in • edStatus • edNo_IN_TA • edDT_IN_TA • edNO_DOC_TA • edClommJn 	<ul style="list-style-type: none"> • REC_NO • NO_IN • NO_DOC • DT_IN • NAME_DOC • CLIENT • POST_IN • CATEG • DT_DOC • DT_CTRL • DT_EXEC • FIO_IN • STATUS • NO_IN-TA • NO_DOC_TA • DT_IN_TA • COMM_IN 	<ul style="list-style-type: none"> • DEPO_SCH • TYP_LS • IO_PERS • F_PERS • S_N • TYP_DOC • DT_DOC • GIVEN • ADDRESS • PADDRESS • INN

Рис. 4.15. Основні класи

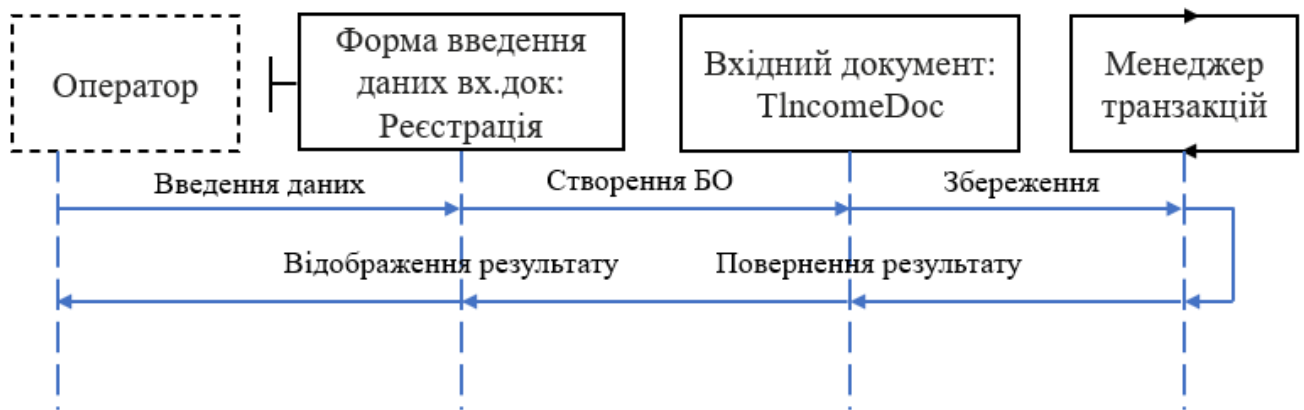


Рис. 4.16. UML–діаграма послідовності процесу «Реєстрація вхідного документа»

Реєстрація вхідного документа			
Вх. номер документа	<input type="text"/>	Номер акту	<input type="text"/>
Дата та час отримання	<input type="text"/>		<input type="text"/>
Найменування	<input type="text"/>		
Категорія	<input type="text"/>		
Відправник	<input type="text"/>		
Дата відправлення	<input type="text"/>		
ЮЛ-вих.№, підписав	<input type="text"/>		
Документи представлені	<input type="text"/>		
Дата контролю	<input type="text"/>	Статус	<input type="text"/>
Коментар	<input type="text"/>		
Зберегти		Скасувати	

Рис. 4.17. Шаблон діалогової форми «Реєстрація документа, що входить»

В результаті проектування окремої функції СВР «Planaria» в ЕАВР вдалося описати весь БП реєстрації вхідних документів, описати основні класи та отримати форму ПІ.

Додавання нового одиничного документа

Вх. номер документа	<input type="text"/>	Номер акту	<input type="text"/>
Дата та час отримання	<input type="text"/>		<input type="text"/>
Найменування	<input type="text"/>		
Категорія	<input type="text"/>		
Відправник	<input type="text"/>		
Дата відправлення	<input type="text"/>		
ЮЛ-вих.№, підписав	<input type="text"/>		
ПІБ представника	<input type="text"/>		
Документи представлені	<input type="text"/>		
Категорія відправника	<input type="text"/>		
Дата контролю	<input type="text"/>	Термін	<input type="text"/>
Статус	<input type="text"/>		
Коментар	<input type="text"/>		
Вх. № ТА	<input type="text"/>		
Дата пйому В ТА	<input type="text"/>	Дата відправки реєстратору	<input type="text"/>

☒ Відображати лише не заповнені акти

Створити одиничний Створити активований Копіювати Зберегти Закрити

Рис. 4.18. Діалогова форма реєстрації вх. документа

Отримані шаблони програмних модулів доопрацьовані до функціональності, аналогічної тій, яка вже реалізована у СБР «Planaria», що і підтверджує працездатність модулю та методу.

4.2.2. Модуль "Трансфер–агентський обмін"

Функціонування облікової системи ринку важко уявити без інституту трансфер–агенту (ТА). Пункти трансфер–агенту виконують такі функції:

- приймають від документів та передають реєстратору інформацію, необхідну для виконання операцій у реєстрі документів;

– приймають від реєстратора та передають акціонеру інформацію з реєстру (виписки, довідки, повідомлення).

Модуль трансфер–агенту призначений для автоматизації цих функцій. Цей модуль є програмним агентом, що володіє властивостями співпраці та комунікації. Два агенти обмінюються між собою інформацією із засобів передачі повідомлень певного формату. У заголовку повідомлення вказується, від якого трансфер–агенту пункту за яким емітентом, яка інформація надійшла.

Ґрунтуючись на цих даних, сторона, що приймається, виконує певні дії. Розглянемо докладніше принцип взаємодії двох агентів.

Перед першою взаємодією їм необхідно обмінятися службовими повідомленнями, що містять таку інформацію: відкриті ключі шифрування; даними щодо емітентів, за якими здійснюватиметься обмін інформацією; дані щодо випусків цінних паперів цих емітентів. Після обміну службовими повідомленнями агенти готові взаємодії.

Користувач за допомогою модуля «Ведення реєстрів» формує інформаційні повідомлення, які зберігаються у черзі вихідних повідомлень. ТА модуль періодично опитує стан черги.

Виявлені повідомлення шифруються, архівуються та надсилаються електронною поштою. Отримані електронною поштою повідомлення розміщуються модулем трансфер–агенту в чергу вхідних повідомлень. Потім вони розраховуються, розшифровуються та розподіляються за потрібними реєстрами. Відпрацьовані повідомлення надходять до архіву.

Діаграма прецедентів для програмного агента трансфер–агента представлена рис. 4.19.

Діаграми послідовності прийому та надсилання поштових повідомлень представлені на рисунках 4.20 та 4.21.

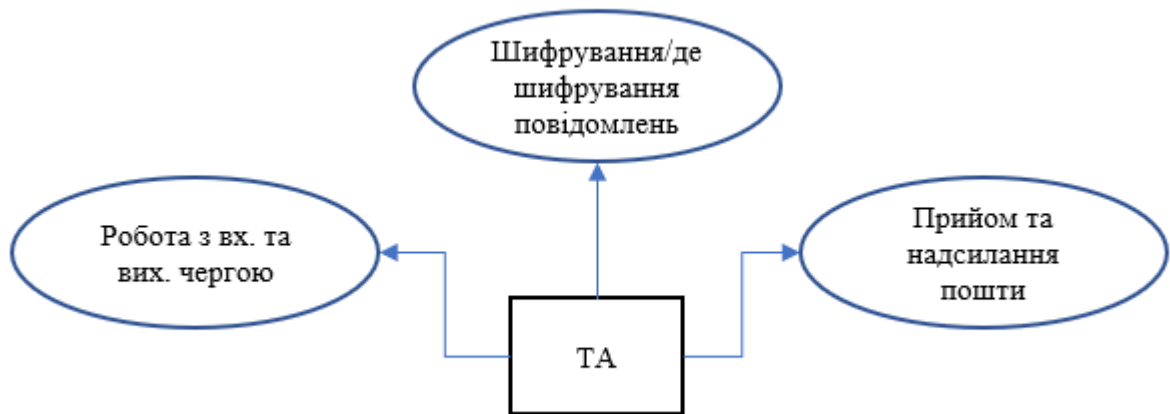


Рис. 4.19. Діаграма прецедентів програмного агента ТА

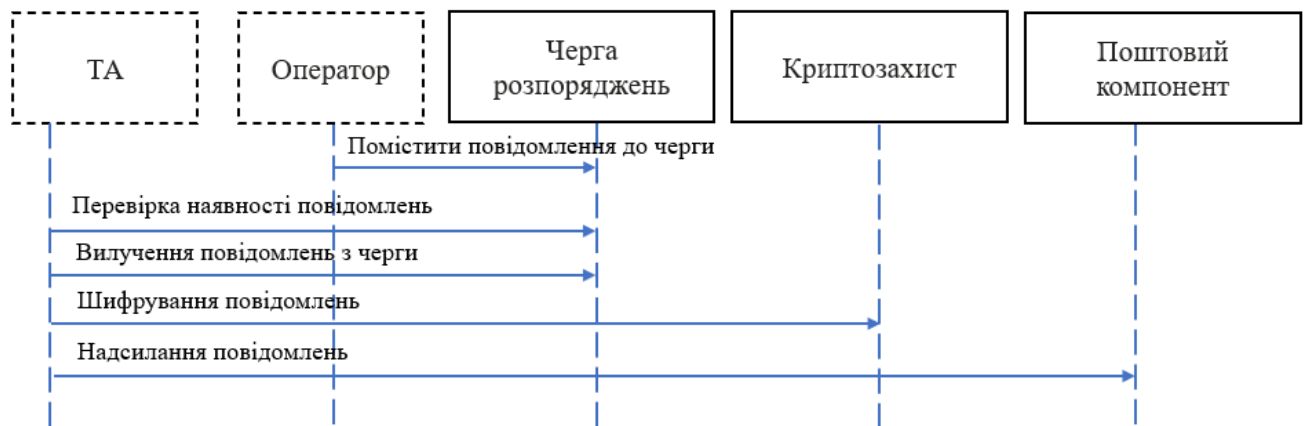


Рис. 4.20. Діаграма послідовності надсилання поштових повідомлень

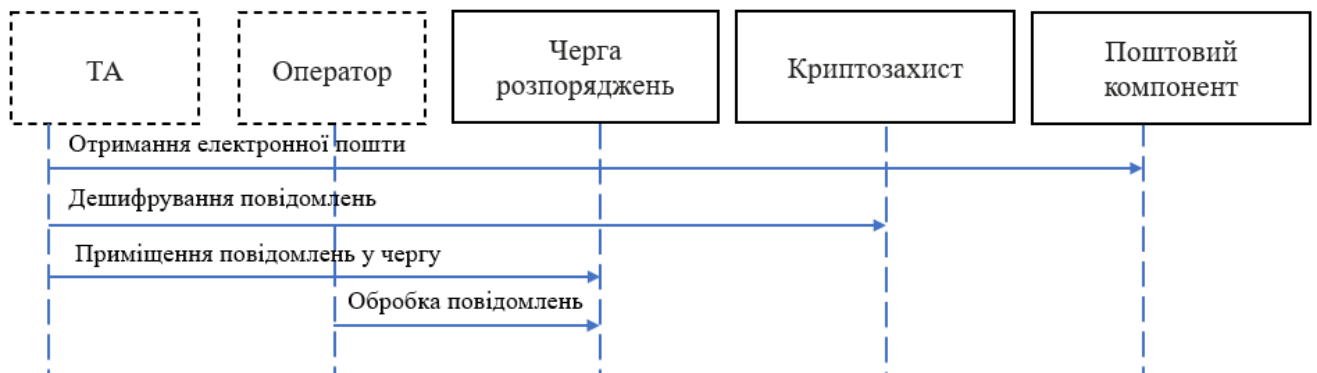


Рис. 4.21. Діаграма послідовності надсилання поштових повідомлень

У результаті проектування програмного агента ЕАВР вдалося описати всю

необхідну інформацію для його реалізації. Отриманий ТА–модуль СВР «Planaria» виконує функції прийому та відправлення пошти, роботу з чергами розпоряджень, тобто. виконує необхідні дії в організацію ТА обміну.

4.2.3. Система «Електронний документообіг»

Для покращення якості роботи реєстратора необхідне впровадження електронного документообігу (ЕДО) між реєстратором та депозитарієм.

Додатковий модуль СВР «Planaria» – «Електронний документообіг» дозволяє автоматично приймати електронні документи (ЕД) до бази даних СВР «Planaria» та формувати електронні виписки, довідки та повідомлення. Для підтвердження прийому ЕД формується технічне повідомлення (ТП), у якому вказується успішність прийому чи виниклі прийоми помилки. DFD–діаграма процесу прийому ЕД показано рис. 4.22.

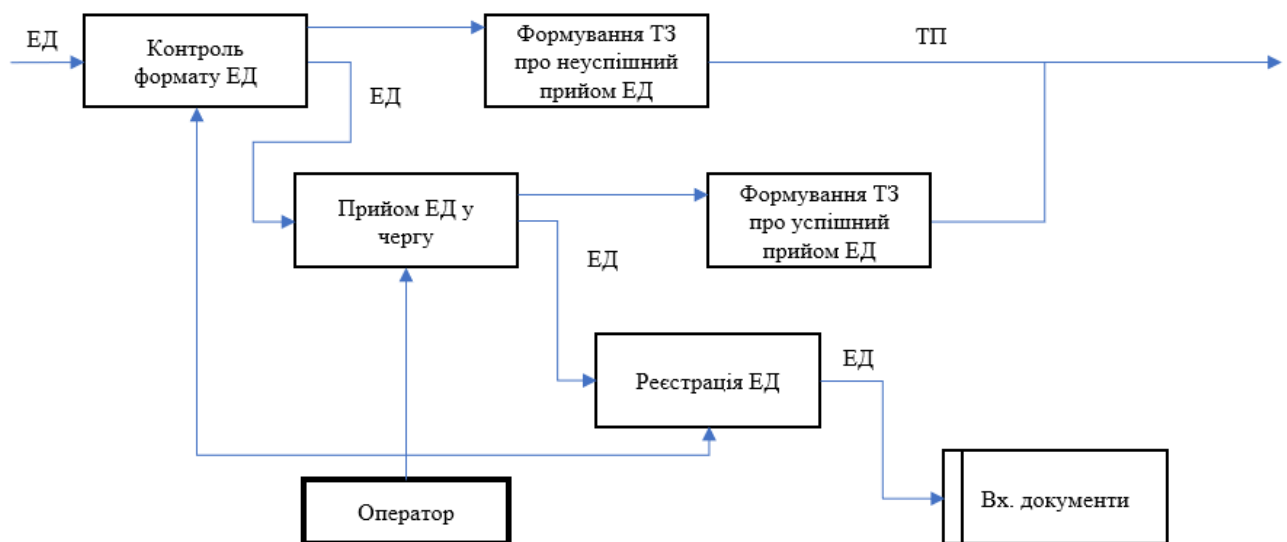


Рис. 4.22. DFD–діаграма процесу прийому електронних документів

У цьому процесі одна зовнішня сутність – оператор, яка конвертується у відповідну роль. Діаграма прецедентів цієї ролі представлено рис. 4.23. За

допомогою EABP намальовано діаграму послідовності прийому документів та ТС (рис. 4.24).

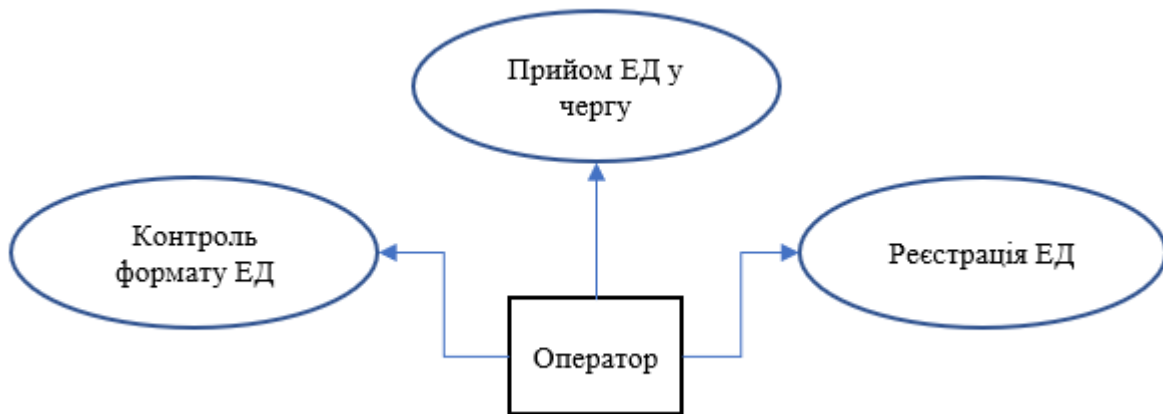


Рис. 4.23. Діаграма прецедентів для ролі Оператор

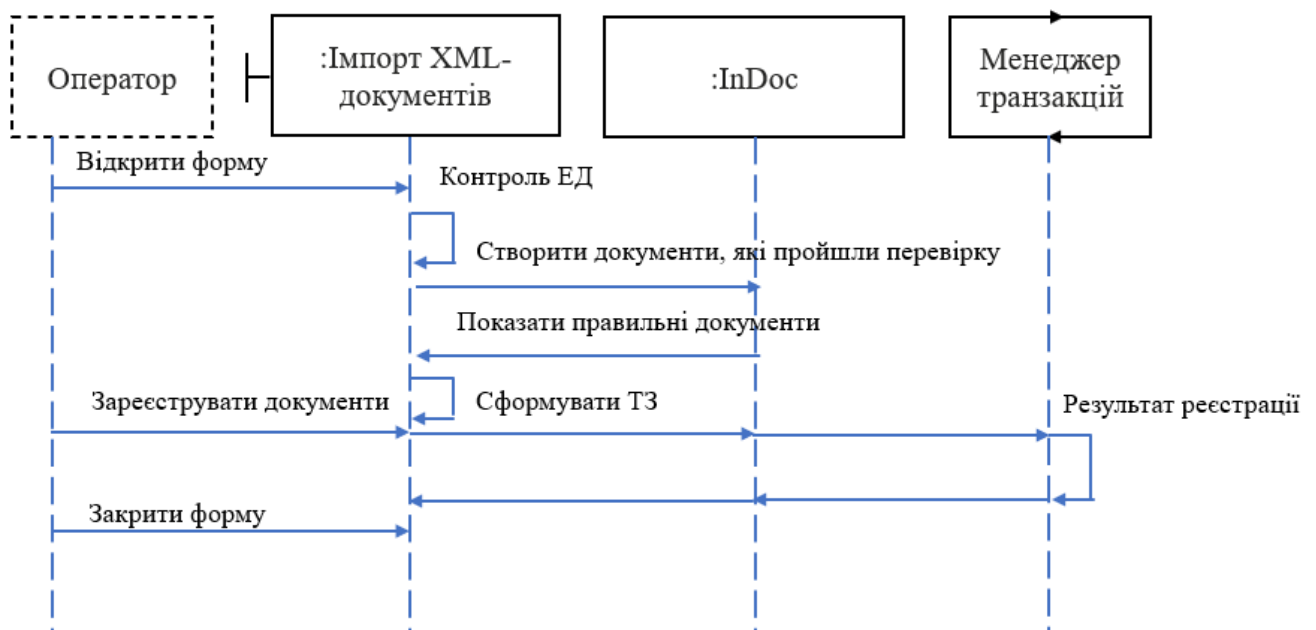


Рис. 4.24. Діаграма послідовності прийому документів

Формування ЕД можливе за бажанням оператора. Крім того, є необхідність масового формування виписок з реєстру на перше та п'ятнадцяте число кожного місяця, а також довідок про операції на день, в якому були проведені операції. Виконання цього рутинного завдання було вирішено покласти на програмного агента. DFD-діаграма процесу формування ЕД представлена рис. 4.25.

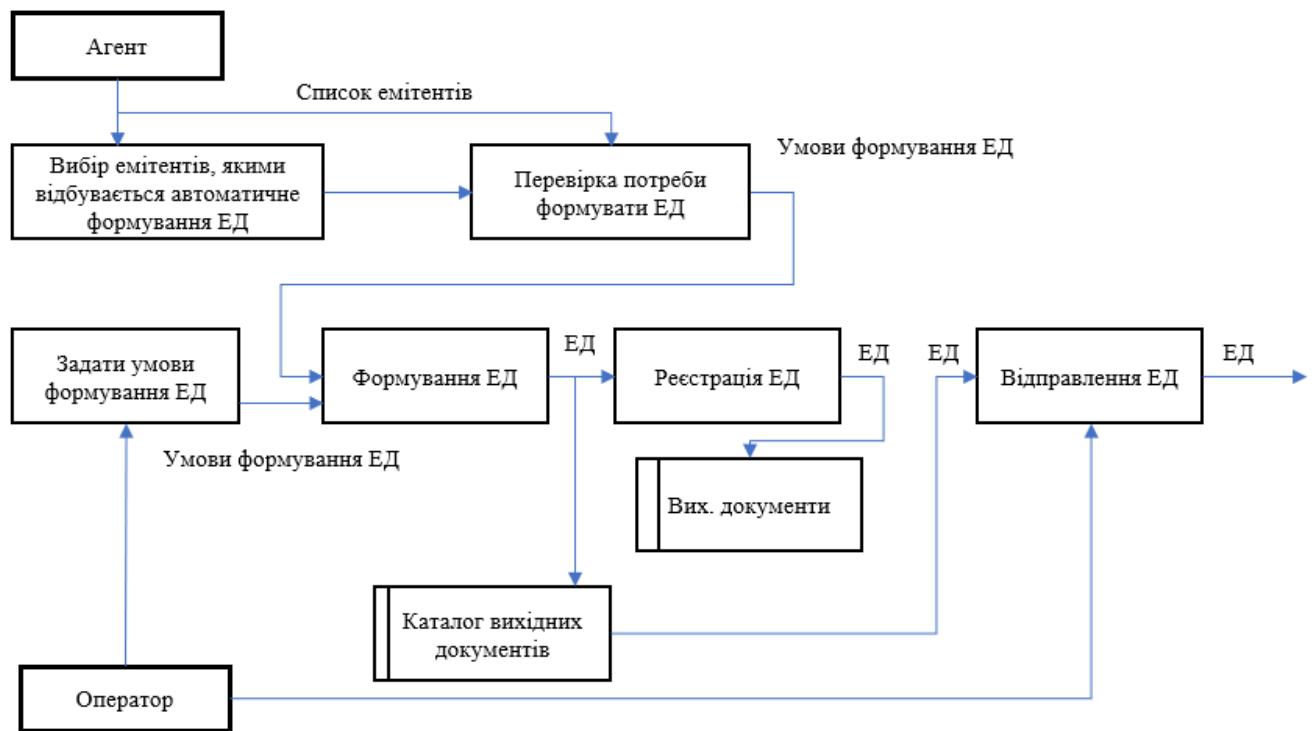


Рис. 4.25. DFD–діаграма процесу формування ЕД

Сценарій поведінки агенту був представлений у вигляді послідовності діаграми (рис. 4.26).



Рис. 4.26. Сценарій поведінки агенту

Впровадження електронного документообігу у роботі реєстратора дозволяє суттєво скоротити трудомісткість прийому списків клієнтів номінального власника та прискорити отримання ним виписок та довідок за своїми запитами.

В результаті проектування додаткового модуля та програмного агенту в EABP вдалося описати необхідну інформацію для їх реалізації. Запропонований метод та модуль EABP дозволили спростити передачу результатів роботи між аналітиком та розробником ПЗ за рахунок перетворення даних моделі БП на модель ІС.

4.3. Експериментальні оцінки

Для оцінки ефективності запропонованого методу та CASE–засобу проекти, описані в розділі 3, розроблені з використанням додатків–аналогів (AllFusion, Rational Rose) та EABP [78, 96,101]. Весь процес розробки було поділено на такі етапи:

1. Побудова моделі МППР: опис процесів перетворення (перетворювачі та ресурси).
2. Побудова моделі МППР: опис інтелектуальних агентів.
3. Побудова DFD–діаграм шляхом конвертації із МППР.
4. Побудова діаграми прецедентів.
5. Побудова діаграми класів: базові класи.
6. Побудова діаграми класів: інші класи.
7. Наповнення бази знань екземплярами.
8. Побудова діаграми послідовності.
9. Проектування структури БД: проектування таблиць.
10. Проектування структури БД: проектування зв'язків.
11. Проектування структури БД: опис тригерів.
12. Проектування структури БД: генерування БД.
13. Проектування інтерфейсу користувача.
14. Генерація коду програми: створення структури програми.
15. Генерація коду програми: використання алгоритмів агентів у коді.
16. Генерація коду програми: генерація інтерфейсів користувача.

На рис. 4.27 показано тривалість етапів розробки для проекту з аналізу БП та розробки технічного завдання на ЕІС «Одеський науково–дослідний інститут зв’язку» в системах аналогів та ЕАВР. Етап 3 в системах аналогах виконувався вручну, етапи 13, 15–16 в них не виконувались.

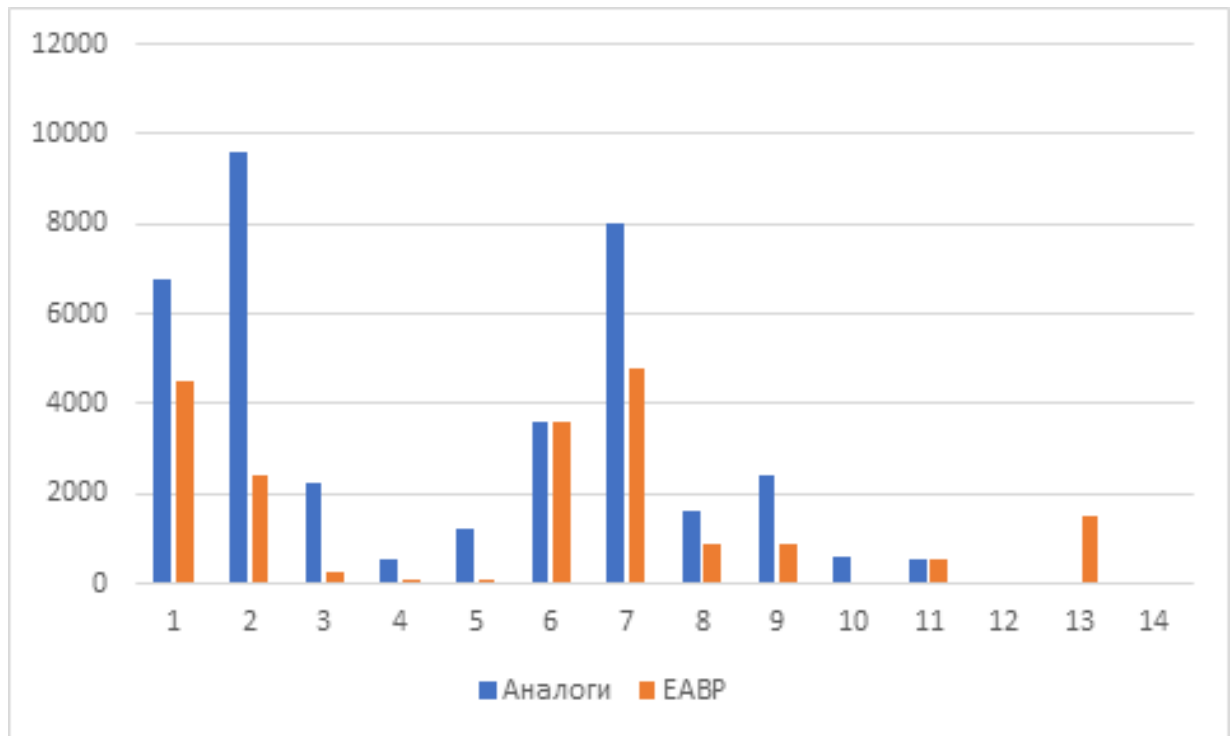


Рис. 4.27. Тривалість етапів розробки для проекту з аналізу БП та розробки технічного завдання на ЕІС ДП «Одеський науково–дослідний інститут зв’язку»

Як видно з результатів оцінки тривалості етапів розробки, розроблений програмний засіб дає значний (в середньому більше 40%) ефект прискорення процесу розробки інформаційної системи, а деякі етапи на системах аналогах доводилось проводити вручну (побудова DFD–діаграм шляхом конвертації із МППР), чи взагалі відмовитись від їх проведення за неможливістю реалізації завдань з проектуванням інтерфейсу користувача, та генерації коду програм:

- використання алгоритмів агентів у коді.
- генерація інтерфейсів користувача.

Оцінка ефективності використання ЕАВР для проекту ЕІС ДП «Одеський
науково–дослідний інститут зв’язку»

№ етапу	Тривалість у системах – аналогах, хв	Тривалість в ЕАВР, хв	Ефект, %
1	6700	4450	32
2	9550	2350	74
3	2200	200	88
4	550	60	87
5	1150	95	91
6	3550	3550	0
7	8010	4810	40
8	1630	910	45
9	2420	920	64
10	650	65	89
11	550	550	0
12	25	25	0
13	–	1510	–
14	25	25	0

Як видно з результатів оцінки тривалості етапів розробки для проекту ЕІС ДП, розроблений програмний засіб дає значний (в середньому більше 60%) ефект прискорення процесу розробки інформаційної системи, а деякі етапи на системах аналогах доводилось проводити вручну(побудова DFD–діаграм шляхом конвертації із МППР), чи взагалі відмовитись від їх проведення за неможливістю реалізації завдань, наприклад побудова діаграми класів відмінних від базових

На рис. 4.28 показано тривалість етапів розробки для проекту Модуль «Трансфер–агентський обмін».

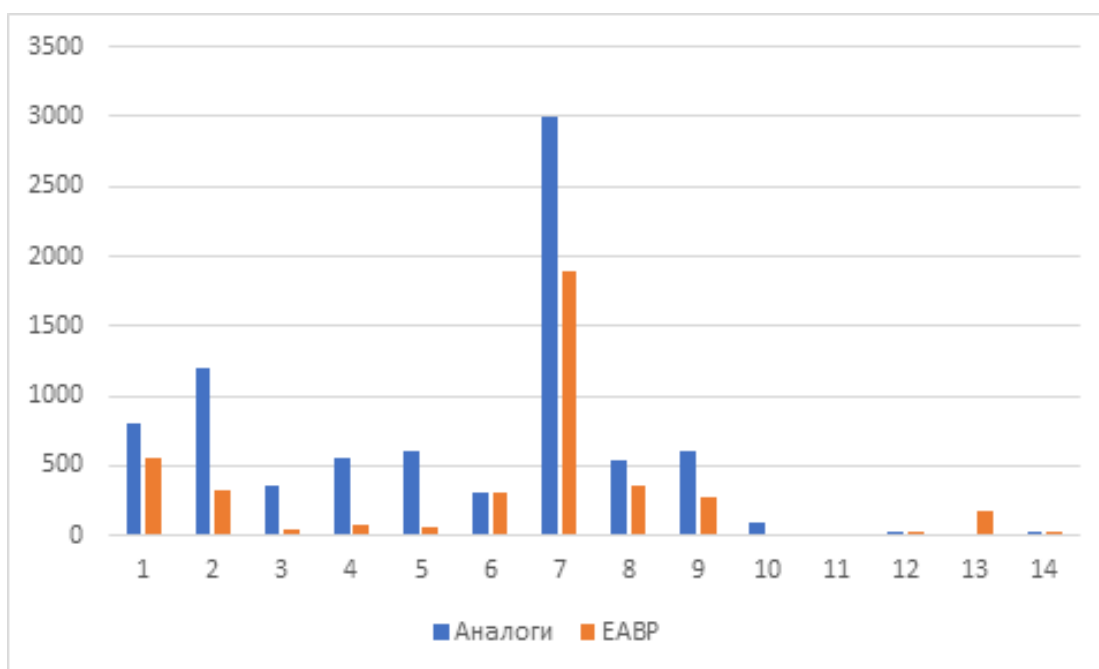


Рис. 4.28. Тривалість етапів розробки для проекту Модуль " Трансфер–агентський обмін "

Таблиця 4.3

Оцінка ефективності використання ЕАВР для проекту Модуль " Трансфер–агентський обмін "

№ етапу	Тривалість у системах – аналогах, хв	Тривалість в ЕАВР, хв	Ефект, %
1	800	550	30
2	1210	340	74
3	365	46	86
4	565	70	87
5	605	55	90
6	310	310	0
7	3100	2000	37
8	550	360	35
9	610	285	54
10	110	15	90
11	0	0	0
12	25	25	0
13	—	190	—
14	25	25	0

На рис. 4.29 показано тривалість етапів розробки для проекту Модуль «Електронний документообіг».

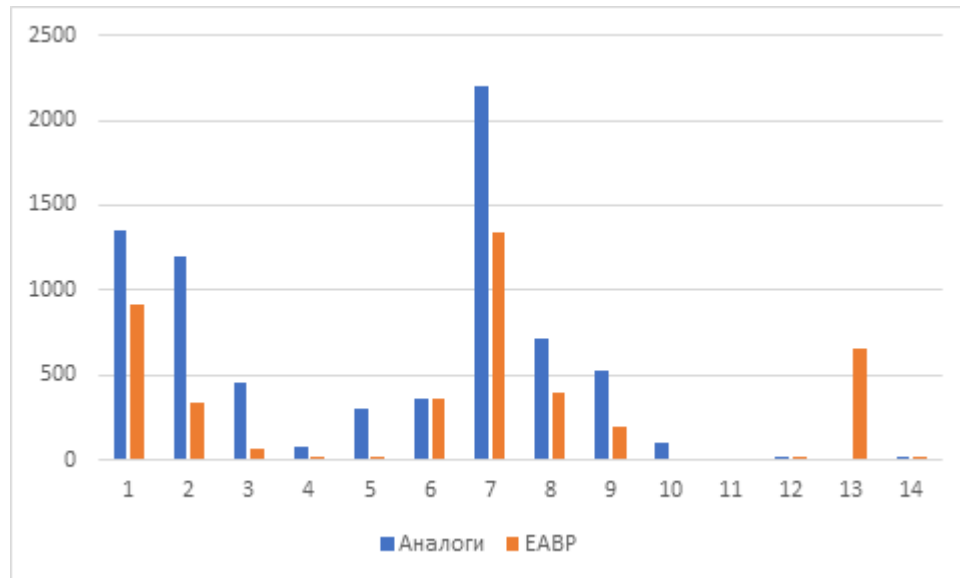


Рис. 4.29. Тривалість етапів розробки для проекту Модуль "Електронний документообіг"

Таблиця 4.4

Оцінка ефективності використання ЕАВР для проекту Модуль «Електронний документообіг»

№ етапу	Тривалість у системах – аналогах, хв	Тривалість в ЕАВР, хв	Ефект, %
1	1360	920	33
2	1300	435	72
3	460	70	86
4	85	15	81
5	305	24	93
6	350	350	0
7	2210	1350	39
8	730	410	45
9	530	205	63
10	115	15	90
11	0	0	0
12	25	25	0
13	—	670	—
14	25	25	0

На рис. 4.30 показано тривалість етапів розробки для проекту Модуль «Вхідні документи».

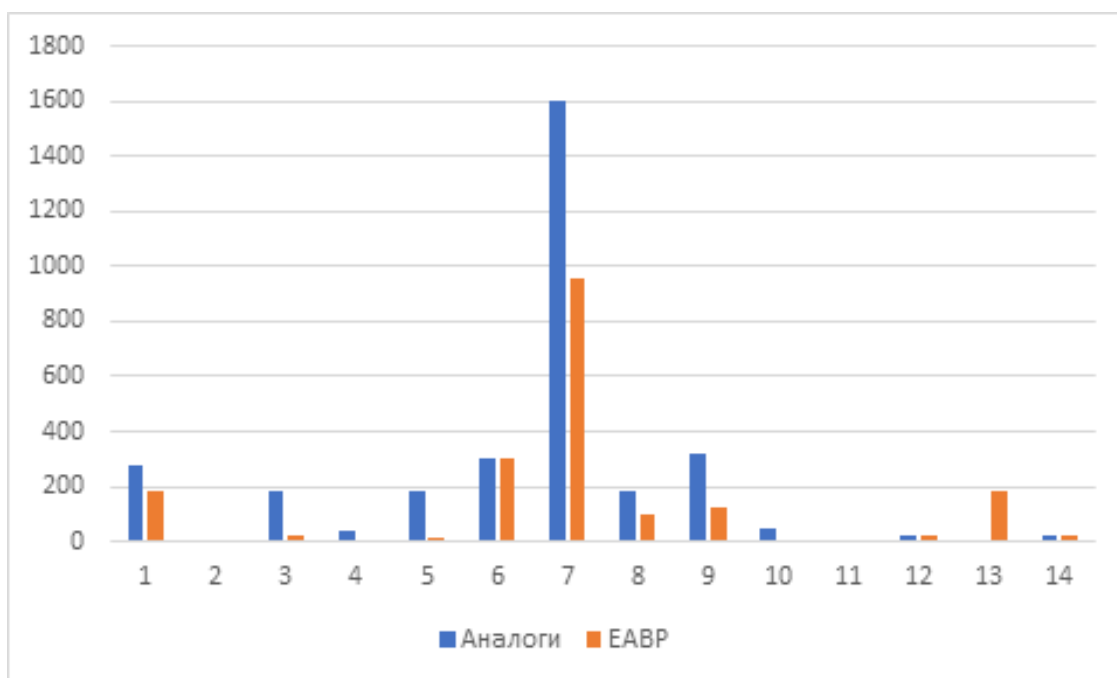


Рис. 4.30. Тривалість етапів розробки для проекту Модуль «Вхідні документи»

Таблиця 4.5

Оцінка ефективності використання ЕАВР для проекту Модуль «Вхідні документи»

№ етапу	Тривалість у системах – аналогах, хв	Тривалість в ЕАВР, хв	Ефект, %
1	290	190	35
2	—	—	—
3	190	30	90
4	45	10	87
5	170	16	90
6	310	310	0
7	1610	970	40
8	190	110	45
9	330	130	62
10	60	6	90
11	10	10	0
12	25	25	0
13	—	190	—
14	25	25	0

Для порівняльного аналізу тривалості етапів розробки для різних проектів та засобів було побудовано графік із логарифмічною шкалою по осі Y, наведений на рис. 4.31.

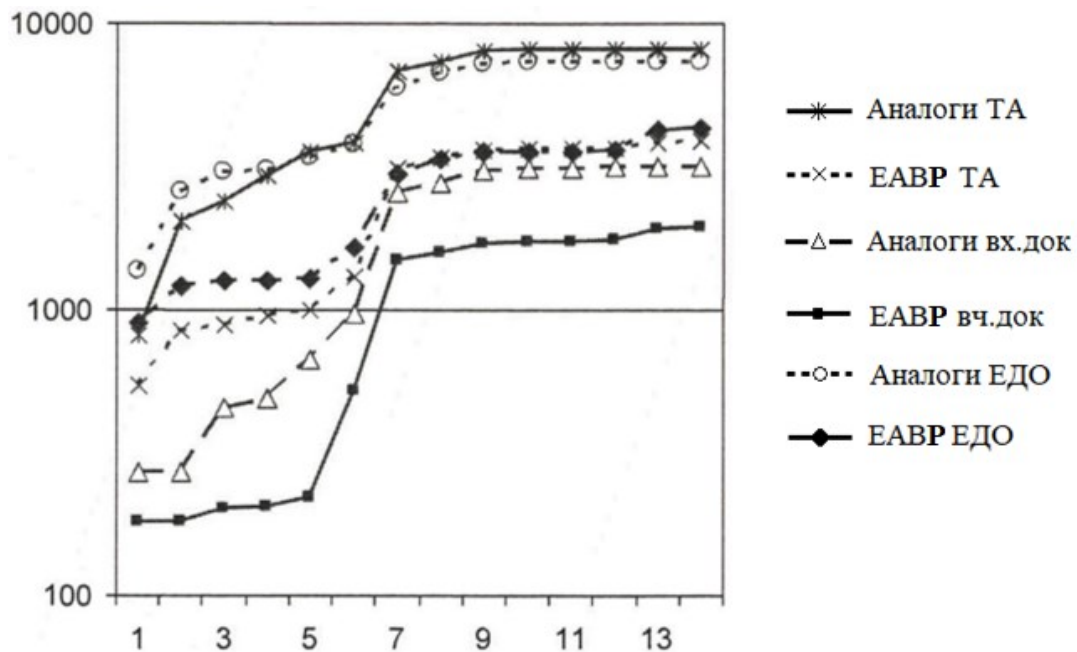


Рис. 4.31. Тривалість етапів розробки для всіх проектів та засобів розробки

Висновки до розділу 4

В даному розділі була показана можливість розробки інформаційних систем за допомогою розробленого програмного засобу ЕАВР на прикладі розробки та впровадження модулів «Трансфер–агентський обмін» та «Рух контингенту» для електронно-інформаційної системи електронного документообігу, також була проведена оцінка ефективності впровадження модулів і виконана реалізація цих модулів у додатках аналогах, з порівняльним аналізом оцінки ефективності розробленого методу.

Для оцінки ефективності методу підтримки прийняття рішень у області розробки інформаційних систем та модулю ЕАВР була проведена паралельна розробка інформаційної системи за допомогою CASE–засобів Process Modeler та Rational Rose (система аналог) та ЕАВР. Для підприємства «Одеський науково–дослідний інститут зв’язку» було розроблено декілька додаткових модулів для системи ведення реєстрів «Planaria 2», також було проведено аналіз бізнес–

процесів та розробка технічного завдання на електронно-інформаційну систему ДП «Одеський науково-дослідний інститут зв'язку».

В результаті оцінки тривалості етапів розробки для проекту ЕІС ДП «Одеський науково-дослідний інститут зв'язку» було виявлено, що розроблений програмний засіб дає значний ефект прискорення процесу розробки інформаційної системи, а деякі етапи на системах аналогах доводилось проводити вручну чи взагалі відмовитись від їх проведення за неможливістю реалізації завдань на розробку.

Використання ЕАВР дає такі переваги: інтеграція бізнес-моделювання, структурного та об'єктно-орієнтованого підходу, конвертація одних діаграм в інші. Усе це призводить до скорочення часу переходу між етапами проекту, порівняно з використанням існуючих систем-аналогів. Середній ефект становив 42%. Також, завдяки автоматизації бізнес-процесів – підвищується ефективність праці кінцевих користувачів.

ВИСНОВКИ

В результаті дисертаційних досліджень вирішено актуальне наукове завдання щодо розробки методів і засобів в області створення інформаційних систем для аналізу варіантів реалізації процесів в організаційно-технічних системах за допомогою імітаційного моделювання, скорочення часу на впровадження інформаційних систем шляхом автоматизації процесів переходу між етапами розробки, а також зменшення кінцевих витрат при розробці інформаційних систем. Дане наукове завдання має суттєве значення для теоретичних основ та інструментальних засобів створення та використання систем підтримки прийняття рішень. Відсутність аналогічних рішень у нашій країні та за кордоном робить результати досліджень пріоритетними.

В дисертації одержані такі основні результати:

1. На підставі проведеного аналізу виявлено існування протиріч між вимогами до розроблюваного методу підтримки прийняття рішень: між вимогою до підвищення ефективності інформаційної системи з використанням мультиагентного підходу, що потребує додаткових витрат як часу так і коштів, та вимогою на зменшення витрат на розробку ефективного математичного та програмного забезпечення інформаційної системи. Ці обставини не дають можливості до збільшення ефективності процесу прийняття рішення при розробці інформаційних систем. Дане протиріччя підтверджує актуальність поставленого наукового завдання.

2. Вперше розроблено архітектуру програмного забезпечення для автоматизації процесу проектування діаграм DFD, IDEF0, UML–схем прецедентів, послідовностей і класів, а також програмного інтерфейсу при проектуванні інформаційної системи, що базується на удосконаленому методі підтримки прийняття рішень та розробленому методі трансформації моделі організаційно–технічної системи в модель інформаційної системи, використання якої дозволяє спростити процес розроблення програмного забезпечення інформаційних систем та надає можливість створення прототипу форм інтерфейсу користувача.

3. Вперше розроблено концептуальну модель предметної області організаційно–технічної системи, що заснована на фреймово–семантичному представленні знань та дозволяє більш повно відобразити предметну область організаційно–технічної системи, впорядкувати та вдвічі прискорити процес розробки програмного забезпечення інформаційної системи.

4. Удосконалено метод підтримки прийняття рішень для завдання розробки інформаційної системи на основі інтеграції структурних, агентних та об'єктно–орієнтованих підходів, який відрізняється від існуючих методів використання інформаційної складової опису предметної області на основі аналізу інформаційних потоків для побудови концептуальної моделі інформаційної системи, що дозволяє значно прискорити та спростити розробку програмних комплексів у середньому в 1,8 разів.

5. Вперше розроблено метод трансформації моделі організаційно–технічної системи в модель інформаційної системи, що сформована як сукупність діаграм функціонального та об'єктно–орієнтованого підходу на основі використання концептуальних моделей предметної області. Даний метод дозволяє побудувати взаємно–однозначну відповідність між концептуальною моделлю предметної області організаційно–технічної системи та моделлю інформаційної системи для забезпечення ефективної взаємодії між фахівцями–предметниками та ІТ–фахівцями.

6. Розроблено метод підтримки прийняття рішень при розробці інформаційних систем, предметної області організаційно-технічних систем, який відрізняється від існуючих:

- застосуванням на етапі обстеження предметної області мультиагентного імітаційного моделювання для аналізу процесів організаційно-технічних систем;
- розглядом розподіленої інформаційної системи у вигляді мультиагентної системи;
- застосуванням фреймово–семантичної моделі подання знань на основі фрейм–концептів та концептуальних графів, з метою формалізації знань про предметні області розробки інформаційної системи та організаційно-технічних систем;

- інтеграцією структурного, об'єктно–орієнтованого та мультиагентного підходу для вирішення задач підтримки прийняття рішень під час автоматизації процесів організаційно–технічної системи;

- перетворенням мультиагентної імітаційної моделі організаційно–технічної системи в основу моделі архітектури інформаційної системи та її елементів, представлення архітектури у вигляді структурних діаграм та діаграм об'єктно–орієнтованого підходу для забезпечення ефективної взаємодії між фахівцями–предметниками та ІТ–фахівцями;

- аналізом ефективного розподілу бази даних за наявності ризику тривалих відмов у обслуговуванні розподіленої інформаційної системи.

7. Проведено аналіз ефективності запропонованого методу та реалізація його у пакеті розробленого модулю. Він показав, що відбувається скорочення переходу між етапами розробки інформаційних систем. Середній ефект становив 42 % у частині розробки моделі інформаційної системи.

8. Результати досліджень прийняті до впровадження в ДП «Одеський науково–дослідний інститут зв'язку» (два акти від 31.08.2022 р.); у «Одеський науково–дослідний інститут зв'язку». З його допомогою було підготовлено технічне завдання на електронну інформаційну систему, розроблено модулі системи ведення реєстрів «Planaria» та «Planaria 2». Економічний ефект заощадження коштів від застосування запропонованих моделей «як буде» і автоматизації процесу «Рух контингенту» становить 26 % на рік

9. Розроблені системи підтримки прийняття рішень відрізняються від існуючих:

- інтеграцією структурного та об'єктно–орієнтованого підходів з імітаційного моделювання мультиагентних процесів перетворення ресурсів;

- можливістю конвертувати структурні діаграми на діаграми об'єктно–орієнтованого підходу;

- можливістю створення моделі поведінки програмного агенту;

- можливістю створення прототипу форм інтерфейсу користувача непрограмуємим користувачем.

10. Мета досліджень щодо скорочення часу та зниження витрат на розробку і впровадження інформаційних систем досягнута та всі часткові завдання вирішені повністю. Наукові результати досліджень є внеском у розвиток теоретичних і прикладних основ розроблення та дослідження моделей і методів підтримки прийняття рішень при розробці інформаційних систем на основі мультиагентного підходу.

11. Перспективними шляхами подальших досліджень у зазначеному напрямку може бути широке коло питань щодо розробки нових та удосконалення існуючих методів підтримки прийняття рішень на основі мультиагентних підходів, підвищення достовірності прийнятого рішення та скорочення часу на прийняття рішення в галузі розробки інформаційних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kolumbet V., Svynchuk O. Multiagent methods of management of distributed computing in hybrid clusters. *Сучасні інформаційні системи*. 2022. Том 6. № 1. С. 32 – 36.

<http://ais.khpi.edu.ua/article/view/254197/251480>

2. Колумбет В.П., Сторчак К.П. Метод трансформації моделі організаційно–технічної системи в модель інформаційної системи. *Науково–практичний журнал «Зв'язок»*. 2022. Вип. № 1 (155). С. 33 – 36.

<http://con.dut.edu.ua/index.php/communication/issue/view/159>

3. Барабаш О.В., Колумбет В.П. Дослідження систем масового обслуговування на основі імітаційного моделювання з урахуванням мультиагентного підходу. *Інфокомунікаційні та комп'ютерні технології*. 2022. Том 2. № 04. С. 115 – 121.

<https://visn-icct.uu.edu.ua/index.php/icct/issue/view/4>

4. Барабаш О.В., Колумбет В.П. Інтелектуальний аналіз та обробка текстових документів на основі мультиагентного підходу. *Інформатика та математичні методи в моделюванні*. 2022. Том 12. № 4. С. 296 – 305.

http://immm.op.edu.ua/files/archive/n4_v12_2022/immm_n4_v12_2022.pdf

5. Барабаш О.В., Колумбет В.П. Оптимізація обробки запитів кластерами в реальному часі із застосуванням мультиагентного підходу. Матеріали XIX міжнародної науково–практичної конференції молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики», м. Київ, 20 – 23 квітня 2021 р. Київ, КПІ ім. Ігоря Сікорського, 2021. Том 2. С. 121 – 122.

<https://tef.kpi.ua/files/pdf/2021-tom2-tezy.pdf>

6. Колумбет В.П., Барабаш О.В. Використання методу потенціалів для керування МРТС в середовищі з перешкодами. XIV міжнародна науково–практична конференція «Інтегровані інтелектуальні робототехнічні комплекси (ІІРТК–2021)»: збірка тез, м. Київ, 18 – 19 травня 2021 р. Київ, НАУ, 2021. С. 34 – 36.

<https://drive.google.com/drive/folders/1mVxrvWRYfsG2O8jgbTwtXHhkGn9UobCq>

7. Колумбет В.П., Барабаш О.В. Мультиагентний підхід до оцінки стану безпеки інформаційних систем. XVII Міжнародна науково–практична конференція «Військова освіта і наука: сьогодення та майбутнє»: збірник тез доповідей, м. Київ, 26 листопада 2021 р. Том 1. Київ, ВІКНУ, 2021. С. 47 – 48.

<https://mil.univ.kiev.ua/page/lib/32>

8. Колумбет В.П. Методичні та теоретичні засади підтримки прийняття рішень, моделювання та розвитку інформаційних систем з використанням мультиагентного підходу. Міжнародна науково–практична інтернет–конференція «Розвиток освіти, науки та бізнесу: результати 2020»: тези доповідей, м. Дніпро, 3 – 4 грудня 2020 р. Том 1. Дніпро, 2020. С. 513 – 515.

<http://www.wayscience.com/wp-content/uploads/2020/12/Part-1-Conference-Results-2020..pdf>

9. Колумбет В. П. Мультиагентний підхід до управління ресурсами в розподілених обчислювальних середовищах. VII Міжнародна науково–практична конференція “*Priority Directions of Science and Technology Development*”, м. Київ, 21 – 23 березня 2021 р. Київ, Україна. С. 340 – 344.

<https://sci-conf.com.ua/wp-content/uploads/2021/03/PRIORITY-DIRECTIONS-OF-SCIENCE-AND-TECHNOLOGY-DEVELOPMENT-21-23.03.21.pdf>

10. Колумбет В.П., Барабаш О.В. Мультиагентний підхід для управління розподіленими обчисленнями на рівні GRID–системи. International scientific conference “*Innovation around us ‘2021*” March 9–10, 2021, Published by: SWorld in conjunction with D.A. Tsenov Academy of Economics, Svishtov, Bulgaria. С. 5 – 8. DOI: 10.30889/2709–183X.2021–07–00

<https://www.sworld.com.ua/konferbg7/cp-bg-7.pdf>

11. Barabash O.V., Svynchuk O.V., Kolumbet V.P. Simulation Modeling of Distributed Random Processes Using a Multiagent Approach. Conference proceedings. International scientific conference “*The current stage of development of scientific and technological progress'2021*”. February 8 – 9, 2021. Karlsruhe, Germany. P. 12 – 16.

<https://www.sworld.education/konfergel5/sbor-ge15.pdf>

12. Колумбет В.П., Барабаш О.В. Розв'язання навігаційної задачі для агента в неоднорідному середовищі. Наукові праці Четвертої міжнародної науково-практичної конференції «Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій» 1–2 лютого 2022 р., м. Київ, НУХТ, 2022. С. 87 – 88.

http://kist.ntu.edu.ua/konferencii/31_konf_2022.pdf

13. Колумбет В.П., Барабаш О.В. Переваги моделі мультиагентного підходу в процесі перетворення ресурсів при розробці інформаційних систем. II Міжнародна науково-практична інтернет-конференція «Інтеграція освіти, науки та бізнесу в сучасному середовищі: зимові диспути»: тези доповідей, м. Дніпро, 4 – 5 лютого 2021 р. Том 1. Дніпро, 2021. С. 486 – 488.

<http://www.wayscience.com/konferentsiya-2-4-5-lyutogo-2021/>

14. Barabash O.V., Svynchuk O.V., Kolumbet V.P. Features of Simulation Modeling of Distributed Random Processes Using the Multiagent Approach. *Modern engineering and innovative technologies*. Karlsruhe, Germany. Issue No 15. Part 1. February 2021. P. 34 – 41.

<https://www.moderntechno.de/index.php/meit/issue/view/meit15-01/meit15-01>

15. Barabash O.V., Svynchuk O.V., Kolumbet V.P. Simulation Modeling of Distributed Random Processes Using a Multiagent Approach. Conference proceedings. International scientific conference “*The current stage of development of scientific and technological progress'2021*”. February 8 – 9, 2021. Karlsruhe, Germany. P. 12 – 16.

<https://www.sworld.education/konfergel5/prog-ge15.pdf>

16. Братушка С. М., Новак С. М., Хайлук С. О. Системи підтримки прийняття рішень. Навчальний посібник. Державний вищий навчальний заклад “Українська академія банківської справи Національного банку України”. Суми: ДДП “УАБС НБУ”, 2010. 265 с. ISBN 978–966–8958–56–4

17. Литвин В.В. Мультиагентні системи підтримки прийняття рішень, що базуються на прецедентах та використовують адаптивні онтології. *Штучний інтелект*. Національний університет «Львівська політехніка», м. Львів, Україна 2'2009 С. 24–33

18. Стеценко І.В. Моделювання систем: навч. посіб. [Електронний ресурс, текст] / І.В. Стеценко; М–во освіти і науки України, Черкас. держ. технол. ун–т. Черкаси : ЧДТУ, 2010. 399 с. ISBN 978–966–402–073

https://dut.edu.ua/uploads/l_1740_60135475.pdf

19. Даревич Р.Р., Досин Д.Г., Литвин В.В. Метод автоматичного визначення інформаційної ваги понять в онтології бази знань. *Відбір та обробка інформації*, 2005. Вип. 22 (98). С. 105 – 111.

20. Бабенко М.В., Братута М.Ю. Використання методик визначення вартості розробки програмного забезпечення при аналізі беззбитковості створення інформаційних систем. *Збірник наукових праць Дніпровського державного технічного університету (технічні науки)*, 2016. Том 1 № 28. С. 123 – 127.

<http://sj.dstu.dp.ua/article/view/145770>

21. SQL Server 2014 Express edition [Електронний ресурс]: <https://www.microsoft.com/en-us/server-cloud/products/sql-server>

22. Голуб О.Ф. Аналіз методології IDEF для розвитку бізнес–процесів. *Радіоелектроніка та молодь у XXI столітті : матеріали 23 Міжнар. молодіж. форуму*, 16–18 квітня 2019 р. Харків: ХНУРЕ, 2019. Т. 2. С. 49–50. [Електронний ресурс]

<https://openarchive.nure.ua/handle/document/22187>

23. Данілін А.В. Стандарти і єдина архітектура інформаційних технологій. Microsoft Press [Електронний ресурс]

<http://www.microsoft.com/Ukraine/Government/Analytics/IntegrationTechnologies/Standards.mspx>

24. Слюсар В.І. (2018). Методологія ідентифікації критичних вимог до ОВТ. Зб. матеріалів VI міжнародної науково–практичної конференції “Проблеми координації воєнно–технічної та оборонно–промислової політики в Україні. Перспективи розвитку озброєння та військової техніки”, 2018. Київ. С. 53 – 56. doi:10.13140/RG.2.2.36335.6928

25. Субботін С.О., Олійник А.О., Олійник О.О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей:

Монографія Під заг. ред. С.О. Субботіна. Запоріжжя: ЗНТУ, 2009. 375 с.

26. А. Н. Калашян, Г. Н. Калянов Структурні моделі бізнесу: DFD–технології. М.: Фінанси та статистика, 2009. 256 с.

27. Stephen A. White, Derek Miers BPMN Modeling and Reference Guide – Future Strategies Inc., 2008. 226 с.

28. Блажко О.В., Лісовиченко О.І., Поховський Є.С., Ямпольський Л.С. Моделювання процесів взаємодії компонентів комп'ютерно–інтегрованих систем із застосуванням апарату процесних алгебр. *Адаптивні системи автоматичного управління*, 2003. № 6 (26). С. 9 – 28.

29. Nirenburg S., Raskin V. Ontological Semantics. Cambridge: MA, 2004.

30. Гнатієнко Г., Снитюк В. Експертні технології прийняття рішень. Київ, Маклаут, 2008. 444 с.

31. Магеровська Т. В., Сенік В. В. Інформатика: навчальний посібник. Львів: ЛьвДУВС, 2014. 348 с.

32. Фаулер М., Скотт К. UML. Основи: короткий посібник зі стандартної мови об'єктного моделювання: Пер. с анг. М.: СИМВОЛ–ПЛЮС, 2013. ISBN:978–5–93286–060–1,5–93286–060–X

33. UML – The Unified Modeling Language. [Електронний ресурс]:

<http://www.uml.org>

34. Paul Hensgen, Umbrello UML Modeller, Підручник з Umbrello UML Modeller. Версія 2.11.0. 2013 рік. [Електронний ресурс]

<https://docs.kde.org/stable5/uk/umbrello/umbrello/index.html>

35. Дзінько А.М. Агентно–орієнтований підхід до розв'язання логістичних задач диспетчеризації матеріальних потоків. *Адаптивні системи автоматичного управління*, 2012. № 21(41). С. 18 – 22.

36. Michael Hammer, James Champy. Reengineering the Corporation: A Manifesto for Business Revolution (Collins Business Essentials), October 10, 2006. 272 pages ISBN–10: 0060559535 ISBN–13: 978–0060559533

37. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник. Черкаси. Черкаський національний університет ім. Б.

Хмельницького, 2017. 434 с. ISBN 978–966–920–208–6

<http://eprints.cdu.edu.ua/1481/1/pro.pdf>

38. MAS–Common KADS. [Електронний ресурс]:

<http://portal.acm.org/citation.cfm?id=749430>

39. Нестеренко А.О., Вислоух С.П. Імітаційне моделювання виробничого процесу складання. *Молодий вчений*, 2018. № 3 (55).

<http://molodyvcheny.in.ua/files/journal/2018/3/91.pdf>

40. Похілько О.Ф., Горбачов І.В., CASE–технологія моделювання процесів з використанням засобів BPWin та ERWin, 2009. 122 с.

41. Кен Ауер, Рой Міллер. Екстремальне програмування: постановка процесу з перших кроків та до переможного кінця. СПб: Пітер, 2003. ISBN 5–318–00132–7

42. Шаповалова С. І. Моделювання дій інтелектуальних агентів в середовищі зі змінними характеристиками. *Східно–Європейський журнал передових технологій*. 2010. № 45. с. 64 – 69.

43. Colin J. Neill and Phillip A. Laplante Requirements Engineering: The State of the Practice – m–n IEEE Software November/December 2003, 40–45.

44. Єфременко Т. М. Реінжиніринг бізнес–процесів : конспект лекцій для студентів денної і заочної форм навчання. Готельно–ресторанна справа. Харків. нац. ун–т міськ. госп–ва ім. О. М. Бекетова. Харків : ХНУМГ ім. О. М. Бекетова, 2019. 100 с.

<http://eprints.kname.edu.ua/53805/1/2019%20147%D0%9B%20147%D0%9B.pdf>

45. Weiss G. Multiagent systems (2nd ed). The MIT Press. – Cambridge, MA, 2013. p. 310

46. Ямпольський Л. С., Ткач Б. П., Лісовиченко О. І. Системи штучного інтелекту в плануванні, моделюванні та управлінні. Підруч. для студ. вищ. навч. закл. ДП «Вид. дім «Персонал», 2011. 544 с. ISBN 978–966–608–961–1

https://maup.com.ua/assets/files/lib/book/syst_shtuc_int.pdf

47. Жуковицький І.В. Скалозуб В.В., Устенко А.Б. Інтелектуальні засоби управління парками технічних систем залізничного транспорту. Монографія. Дніпро, Вид–во ПФ «Стандарт – Сервіс», 2018, 190 с. ISBN 978–617–7382–11–4

48. Самодурова Д.А. Інтелектуальні Агенти та Мультиагентні системи у виробництві. *Економічний вісник Донбасу*, 2019. № 2 (56). С. 179 – 186. DOI: 10.12958/1817–3772–2019–2(56)–1179–186

<http://dspace.nbu.gov.ua/handle/123456789/158414>

49. Wooldridge M., Jennings N. Intelligent Agents: Theory and Practice. The Knowledge Engineering Review. 1995. Vol. 10, №2. P. 115 – 152.

50. Мінухін С.В., Беседовський О.М., Знахур С.В. Методи і моделі проектування на основі сучасних CASE–засобів: навчальний посібник. Харків: Вид. ХНЕУ, 2008. 272 с

<http://repository.hneu.edu.ua/handle/123456789/3086>

51. Коваленко О.С., Добровська Л.М. Проектування інформаційних систем: Загальні питання теорії проектування ІС. Навч. посіб. для студ. КПІ ім. Ігоря Сікорського. Київ. КПІ ім. Ігоря Сікорського, 2020. 192 с. [Електронний ресурс].

https://ela.kpi.ua/bitstream/123456789/33651/1/PIS_KL.pdf

52. Крохмалюк В.В., Січко Т.В. Програма комп'ютерного моделювання BPwin (allfusion process modeler) як основний інструмент в моделюванні інформаційних систем. *Комп'ютерні технології обробки даних*, 2021. С. 38 – 40.

53. CA ERwin Modeling Suite [Електронний ресурс] Режим доступу <https://www.erwin.com/services/erwin-product-training/>

54. Дубейковський В.І. Ефективне моделювання з CA ERwin® Process Modeler: BPwin; AllFusion Process Modeler: практичний посібник, 2009. 121 с.

55. AllFusion Component Modeler 5.0 [Електронний ресурс] Режим доступу https://www.heureka.com/upload/CA_Component_Modeler5.pdf

56. Барабаш О.В., Колумбет В.П. Мультиагентні методи управління розподіленими обчисленнями. International periodic scientific journal (online) SWorld Journal Published by: SWorld & D.A. Tsenov Academy of Economics. Svishtov, Bulgaria. Issue No 7. Part 1. March 2021 С. 104 – 110.

<https://www.sworldjournal.com/index.php/swj/issue/view/swj07-01/swj07-01>

57. Колумбет В.П. Мультиагентний підхід до розрахунку надійності рішень імовірнісних задач. Modern engineering and innovative technologies. Karlsruhe,

Germany. Issue No 16. Part 2. P. 103 – 111.

<https://www.moderntechno.de/index.php/meit/issue/view/meit16-02/meit16-02>

58. Колумбет В.П. Мультиагентний підхід до оптимізації розподілу ресурсів у великих інформаційних системах. *International periodic scientific journal (online) SWorld Journal Published by: SWorld & D.A. Tsenov Academy of Economics. Svishtov, Bulgaria.* Issue No 8. Part 1. May 2021. С. 92 – 96.

<https://www.sworldjournal.com/index.php/swj/issue/view/swj08-01/swj08-01>

59. Barabash O., Kolumbet V. Multiagent approach to computer management in a heterogeneous distributed computer environment. *Системи управління, навігації та зв'язку.* 2022. № 67. Том 1. С. 38 – 42.

<http://journals.nupp.edu.ua/sunz/article/view/2439/1908>

59. Kolumbet V.P., Svynchuk O.V. Simulation of resource distribution in large information systems based on multiagent approach. *Телекомунікаційні та інформаційні технології.* 2021. № 3 (72). С. 4 – 11.

<http://tit.dut.edu.ua/index.php/telecommunication/article/view/2389>

60. Скулиш М.А. Математична модель пошуку оптимального обсягу ресурсів віртуального вузла обслуговування. *Сучасні інформаційні системи,* 2018. Том 2, № 2. С. 30 – 34. doi: 10.20998/2522-9052.2018.2.05

61. Гладун А.Я., Рогушина Ю.В. Онтологічний підхід до проблем підвищення якості розроблення національних стандартів України. Стандартизація, сертифікація, якість. 2016. № 2. С. 19 – 28.

62. Омеляненко В.А. Мультиагентний підхід до розробки інтелектуальних систем управління проектами. Управління проектами та розвиток виробництва: зб. наук. пр. СНУ ім. Володимира Даля, 2016. № 3 (59). С. 5–13.

63. Буров Є.В. Опрацювання знань у когнітивній інформаційній системі, керованій моделями. *Східноєвропейський журнал передових технологій.* 2009. № 6 (7(42)). С. 40 – 49.

64. Ткаченко О., Ткаченко К., Ткаченко О. Лінгвістичні онтології: проектування та використання в навчальних інтелектуальних системах. Цифрова платформа: інформаційні технології в соціокультурній сфері. 2021. Т. 4, № 1. С. 97 – 111.

<https://doi.org/10.31866/2617-796X.4.1.2021.236950>

65. Terry Quatrani. Visual Modeling with Rational Rose and UML. Addison–Wesley Professional. ISBN 978–0–201–72932–0.

66. Wendy Boggs, Michael Boggs. Mastering UML with Rational Rose, 2002. 848 p. ISBN–13: 978–0782140170

67. IBM Rational [Електронний ресурс]:

<http://www-01.ibm.com/software/ru/rational/>

68. Петрик М.Р., Мудрик Ф.Я. Архітектура програмного забезпечення (на базі використання CASE–засобів IBM(sad)) навчальний посібник, *Тернопіль: ТНТУ імені Івана Пулюя*, 2017. 100с.

<https://elartu.tntu.edu.ua/handle/lib/36052>

69. В. Александров Д.В., Костров А.В., Макаров Р.І., Хорошева Є.Р. Методи і моделі інформаційного менеджменту: навч. посібник. М.: *Фінанси і статистика*. 336 с. 2007

70. SAP to Acquire Sybase, Inc (пресреліз). SAP AG. [Електронний ресурс]:

<https://web.archive.org/web/20100714223339/http://www.sap.com/about/newsroom/press.epx?PressID=13202>

71. Пітер Джексон. Введення в експертні системи. М.–СП–Київ: Вільямс, 2001 622 с.

72. Сорока М.Ю. Метод адаптації поведінки агентів в інтелектуальній навчальній системі підготовки диспетчерів управління повітряним рухом. *Системи управління, навігації та зв'язку*. 2020. Т. 2, № 60. С. 17 – 20.

<https://doi.org/10.26906/SUNZ.2020.2.017>

73. Minsky M. A framework for Representing Knowledge in The Psychology of Computer Vision, P.H. Winston (ed.), McGraw–Hill 1975.

74. Швецов А.Н. Модели и методы построения корпоративных интеллектуальных систем поддержки принятия решений: дис. д–ра техн, наук: 05.13.01 / А.Н. Швецов. Санкт–Петербург, 2004. 461 с.

75. Sowa J.F. Conceptual graphs for a database interface. *IBM Journal of Research and Development*, 1976. № 20:4. P. 336 – 357.

76. Sowa J.F. Conceptual Structures: Information Processing in Mind and Machine. Reading, MA: Addison Wesley, 1984. 481 p.
77. Sowa J.F. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Pacific Grove, CA: Brooks Cole Publishing Co., 2000. 594 p.
78. Budaev D., Amelin K., Voschuk G. Realtime task scheduling for multi-agent control system of UAV's group based on network-centric technology. Int. Conference on Control, Decision and Information Technologies (CoDIT'2016). 2016. P. 378 – 381.
79. Філатова Т.В., Журан О.А., Івченко І.Ю. Проектування інформаційних систем та рішення інформатизації переходу компаній в онлайн сферу. *Інформатика та математичні методи в моделюванні*. 2021 Том 11, № 4. С. 365 – 372. <https://doi.org/10.15276/imms.v11.no4.365>
80. Кобозєва А.А. Розробка інструментальних засобів автоматизованого проектування комплексу технічних засобів інформаційної системи. *Кібербезпека в Україні: правові та організаційні питання*. Матеріали міжнародної науково-практичної конференції, 26 листопада 2020 р. Одеса. 2021. С. 86 – 89.
81. Касілов О.В., Крамська К.І. Моделі і метод синтезу агентної інформаційно пошукової системи. *Сучасні інформаційні системи*, 2020. Т. 4, № 2. С. 94 – 99. <https://doi.org/10.20998/2522-9052.2020.2.14>
82. Bukhvalov O., Gorodetsky V., Karsaev O, Koudryavtsev G., Samoylov V. Privacy – Preserved Distributed Coordination of Production Scheduling in B2B Networks: A Multi-agent Approach. Proc. 7th IFAC Conference on Manufacturing Modeling, Management, and Control, 2013, Volume 7. Part 1. P. 2122 – 2127.
83. Грицунов О. В. Інформаційні системи та технології: навч. посіб. Харків: ХНАМГ, 2010. 222 с.
84. IBM, Dell, Fujitsu & Stratus Get Highest Marks in ITIC Reliability Survey [Електронний ресурс] <http://itic-corp.com/blog/2013/02/ibm-dell-fujitsu-stratus-get-highest-marks-in-itic-reliability-survey/#more-1021>
85. Азарова А.О., Лисак Н.В. Комп'ютерні мережі та телекомунікації :

навчальний посібник. Вінниця : ВНТУ, 2012. 293 с. ISBN 978–966–641–487–1

<https://press.vntu.edu.ua/index.php/vntu/catalog/download/208/379/413–1?inline=1>

86. Introduction to mathematical probability J.V. Uspensky, Mc–Groqhill, NY–Landon, 1937.

87. Y. L. Tong A rearrangement inequality for the longest run with an application to network reliability, Journal of Applied Probability 22 (1985). P. 386–393.

88. Section 14.1 Problems and Snapshots from the World of Probability by Blom, Holst, and Sandell

89. Rumbaugh J., Jacobson I., Booch G. The Unified Modeling Language Reference Manual 2nd Edition. Addison–Wesley Professional; 2nd edition (July 15, 2004) English. 721.

90. В Касілов О.В. Мультиагентні системи та технології в ігрових додатках: довідник модуля. «Друкарня Мадрид», 2018. 82 с. ISBN 978–617–7683–03–1

<https://repository.kpi.kharkov.ua/server/api/core/bitstreams/afc70cc3–b499–47f4–ae30–df4764d201c8/content>

91. Sparx Enterprise Architect [Електронний ресурс]
https://sparxsystems.com/enterprise_architect_user_guide/14.0/index/index.html
(date of access: 05.09.2020).

92. Gorodetsky V., Karsaev O., Samoylov V., Konushy V. Support for Analysis, Design and Implementation Stages with MASDK // LNCS, 2009. 5386, pp. 272–287,

93. Швецов А.Н., Сергушичова М.А. Проектування прикладних мультиагентних систем з використанням пакету DISIT журнал «Інформаційні технології» №8, 2009 С. 54 – 60.

94. Іващук О.Т. Економіко–математичне моделювання. Тернопіль: Економічна думка, 2008. 704 с.

95. Самойленко М.І., Кузнецов А.І., Костенко О.Б. Теорія ймовірностей: Підручник. Харків: ХНАМГ, 2008. 194 с.

<https://core.ac.uk/download/pdf/11315952.pdf>

96. Dorokhov A. Simulation simple models and comparison with queuing theory.

[Електронний ресурс]

<http://www.macs.hw.ac.uk/~jphillips/DAS/DataAnalysisTopic8.pdf>

97. Омеляненко В. А., Теоретичні аспекти використання мультиагентного підходу для аналізу інноваційно–технологічних систем [Електронний ресурс] *Економіка: реалії часу*. Науковий журнал. 2016. № 4 (26). С. 142 – 148. Режим доступу до журн.: <http://economics.opu.ua/files/archive/2016/n4.html>

98. Примостка А.О. Мультиагентний підхід до проектування інформаційно–аналітичних систем *Бізнес Інформ*, 2017. № 1. С. 274 – 280. – Режим доступу: http://nbuv.gov.ua/UJRN/binf_2017_1_45.

99. Помазун О.М. Особливості побудови групової системи підтримки прийняття рішень з реінжинірингу бізнес–процесів. *Вісник Запорізького національного університету*, 2009. № 1 (4). С. 83 – 88.

100. Wright A, Sittig D. Sequencing Infrastructure Investments under Deep Uncertainty Using Real Options Analysis. *Journal of Biomedical Informatics*. 2008, Vol. 41 (6), pp. 982 – 990.

101. Batool K., Niazi M.A. Modeling the internet of things: a hybrid modeling approach using complex networks and agent–based models. *Complex Adaptive Systems Modeling*. 2017. Vol. 5, No. 1. P. 4.

<http://dx.doi.org/10.1186/s40294-017-0043-1>

102. Стеценко І.В., Бойко О.В. Система імітаційного моделювання засобами сіток Петрі. *Математические машины и системы*, 2009. Т. 1. №. 1.

103. Agrawal D., Jaiswal L. H., Singh I., Chandrasekaran K., “An Evolutionary Approach to Optimizing Cloud Services”, *Computer Engineering and Intelligent System*, 2012. 3:4, P. 47 – 54.

104. Неруш В.Б., Курдеча В.В. Імітаційне моделювання систем та процесів: Електронне навчальне видання. Конспект лекцій: НН ІТС НТУУ «КП», 2012.с. 115

105. Додонов В.О., Ланде Д.В., Путятін В.Г. Мультиагентний підхід до моделювання інформаційно–аналітичної системи. *Реєстрація, зберігання і обробка даних*, 2016. Т.18, №2. С. 22 –30.

106. Яременко В.С. Огляд наявних мультиагентних систем для задач

інтелектуального аналізу даних. *Вчені записки ТНУ імені Вернадського. Серія: технічні науки*, 2018. Том. 29 (68), ч.2, №3. С. 47 – 55.

107. Axak N. Development of multi-agent system of neural network diagnostics and remote monitoring of patient. *Eastern-European Journal of Enterprise Technologies*. 2016. № 4/9 (82) P. 4 – 11.

108. Савченко В.А. Застосування мультиагентних технологій для інтелектуального аналізу даних в геоінформаційних системах оперативного управління. *Індуктивне моделювання складних систем: Зб. наук. пр. К.: МННЦ ІТС НАН та МОН України*, 2011. Вип. 3. С. 174 – 182.

109. Маклаков С.В. Моделювання бізнес-процесів із BPwin, 2002. 209 с.

110. Савченко В.А. Визначення продуктивності мультиагентної системи підтримки прийняття рішень на основі побудови часового профілю. *Моделювання та інформаційні технології: Зб. наук. пр. К.: ІПМЕ ім. Г.Є. Пухова НАН України*, 2011. Вип. 59. С. 67 – 72.

111. Zambonelli F., Jennings N., Wooldridge M. Developing Multiagent systems: The GAIA methodology // *ACM Transactions on Software Engineering and Methodology*, 2003. 12(3), P. 417 – 470.

112. Жерновий Ю.В. Імітаційне моделювання систем масового обслуговування: практикум. Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. 307 с.

113. Буров Є.В. Ефективність застосування онтологічних моделей для побудови програмних систем. *Математичні машини і системи*. 2013. № 1. С 44 – 55.

ДОДАТОК А

ЗАТВЕРДЖУЮ

Заступник директора ДП «ОНДІЗ»

з наукової роботи, к.т.н.

Леонід ЛЯХОВЕЦЬКИЙ

31.08.2022

АКТ



про впровадження результатів дисертаційної роботи

Колумбета Вадима Петровича

Комісія у складі: начальник лабораторії цифрових систем передавання Державного підприємства «Одеський науково-дослідний інститут зв'язку» (ДП «ОНДІЗ») Лабунський О.А., провідний науковий співробітник ДП «ОНДІЗ», к.т.н. Лашко А.Г., склала цей акт про те, що результати, отримані в дисертаційному дослідженні аспіранта КПІ ім. Ігоря Сікорського Колумбета Вадима Петровича, реалізовані при удосконаленні методу підтримки прийняття рішень при розробці інформаційних систем на основі мультиагентного підходу в області створення інформаційних систем та вирішенню проблеми розробки методу підтримки прийняття рішень.

Удосконалено методи підтримки прийняття рішень на основі інтеграції структурних, агентурних та об'єктно-орієнтованих підходів, імітаційного моделювання.

В результаті автоматизації процесу «Рух контингенту» продуктивність роботи з документообігом в установі підвищилась. Розрахунковий економічний ефект заощадження від застосування запропонованих моделей «як буде» і автоматизації процесу «Рух контингенту» досягає 26% на рік.

Даний акт не є підставою для фінансових взаєморозрахунків

Члени комісії

начальник лабораторії цифрових
систем передавання ДП «ОНДІЗ»

О.А. Лабунський

провідний науковий співробітник
ДП «ОНДІЗ», к.т.н.

А.Г. Лашко

ЗАТВЕРДЖУЮ

Заступник директора ДП «ОНДІЗ»

з наукової роботи, к.т.н.

Леонід ПАХОВЕЦЬКИЙ

31.08.2022



АКТ

про впровадження результатів дисертаційної роботи

Колумбета Вадима Петровича

Комісія у складі: начальник лабораторії цифрових систем передавання Державного підприємства «Одеський науково-дослідний інститут зв'язку» (ДП «ОНДІЗ») Лабунський О.А., провідний науковий співробітник ДП «ОНДІЗ», к.т.н. Лашко А.Г. склала цей акт про те, що результати, отримані в дисертаційному дослідженні аспіранта КПІ ім. Ігоря Сікорського Колумбета Вадима Петровича, реалізовані при удосконаленні методу підтримки прийняття рішень при розробці інформаційних систем на основі мультиагентного підходу в області створення інформаційних систем та вирішенню проблеми розробки методу підтримки прийняття рішень. Удосконалено методи підтримки прийняття рішень на основі інтеграції структурних, агентурних та об'єктно-орієнтованих підходів, імітаційного моделювання.

Перелік реалізації та впровадження результатів дослідження:

В результаті проектування окремої функції СЗР «PLANARIA» в Enterprise Architect BPsim вдалося описати весь БП реєстрації вхідних документів, описати основні класи та отримати форму ПП. Отримані шаблони програмних модулів доопрацьовані до функціональності, аналогічної тій, яка вже реалізована у СЗР «PLANARIA», що і підтверджує працездатність пакета та методу.

Даний акт не є підставою для фінансових взаєморозрахунків

Члени комісії

начальник лабораторії цифрових систем передавання ДП «ОНДІЗ»

О.А. Лабунський

провідний науковий співробітник ДП «ОНДІЗ», к.т.н.

А.Г. Лашко

ДОДАТОК Б

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Kolumbet V., Svyinchuk O. Multiagent methods of management of distributed computing in hybrid clusters. *Сучасні інформаційні системи*. 2022. Том 6. № 1. С. 32 – 36.

<http://ais.khpi.edu.ua/article/view/254197/251480>

2. Колумбет В.П., Сторчак К.П. Метод трансформації моделі організаційно–технічної системи в модель інформаційної системи. *Науково–практичний журнал «Зв'язок»*. 2022. Вип. № 1 (155). С. 33 – 36.

<http://con.dut.edu.ua/index.php/communication/issue/view/159>

3. Барабаш О.В., Колумбет В.П. Дослідження систем масового обслуговування на основі імітаційного моделювання з урахуванням мультиагентного підходу. *Інфокомунікаційні та комп'ютерні технології*. 2022. Том 2. № 04. С. 115 – 121.

<https://visn-icct.uu.edu.ua/index.php/icct/issue/view/4>

4. Барабаш О.В., Колумбет В.П. Інтелектуальний аналіз та обробка текстових документів на основі мультиагентного підходу. *Інформатика та математичні методи в моделюванні*. 2022. Том 12. № 4. С. 296 – 305.

http://immm.op.edu.ua/files/archive/n4_v12_2022/immm_n4_v12_2022.pdf

Наукові праці, які засвідчують апробацію матеріалів дисертації:

5. Барабаш О.В., Колумбет В.П. Оптимізація обробки запитів кластерами в реальному часі із застосуванням мультиагентного підходу. Матеріали XIX міжнародної науково–практичної конференції молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики», м. Київ, 20 – 23 квітня 2021 р. Київ, КПІ ім. Ігоря Сікорського, 2021. Том 2. С. 121 – 122.

<https://tef.kpi.ua/files/pdf/2021-tom2-tezy.pdf>

6. Колумбет В.П., Барабаш О.В. Використання методу потенціалів для керування МРТС в середовищі з перешкодами. XIV міжнародна науково–практична конференція «Інтегровані інтелектуальні робототехнічні комплекси (ІІРТК–2021)»: збірка тез, м. Київ, 18 – 19 травня 2021 р. Київ, НАУ, 2021. С. 34 – 36.

<https://drive.google.com/drive/folders/1mVxrvWRYfsG2O8jgbTwtXHhkGn9UobCq>

7. Колумбет В.П., Барабаш О.В. Мультиагентний підхід до оцінки стану безпеки інформаційних систем. XVII Міжнародна науково–практична конференція «Військова освіта і наука: сьогодення та майбутнє»: збірник тез доповідей, м. Київ, 26 листопада 2021 р. Том 1. Київ, ВІКНУ, 2021. С. 47 – 48.

<https://mil.univ.kiev.ua/page/lib/32>

8. Колумбет В.П. Методичні та теоретичні засади підтримки прийняття рішень, моделювання та розвитку інформаційних систем з використанням мультиагентного підходу. Міжнародна науково–практична інтернет–конференція «Розвиток освіти, науки та бізнесу: результати 2020»: тези доповідей, м. Дніпро, 3 – 4 грудня 2020 р. Том 1. Дніпро, 2020. С. 513 – 515.

<http://www.wayscience.com/wp-content/uploads/2020/12/Part-1-Conference-Results-2020..pdf>

9. Колумбет В. П. Мультиагентний підхід до управління ресурсами в розподілених обчислювальних середовищах. VII Міжнародна науково–практична конференція “*Priority Directions of Science and Technology Development*”, м. Київ, 21 – 23 березня 2021 р. Київ, Україна. С. 340 – 344.

<https://sci-conf.com.ua/wp-content/uploads/2021/03/PRIORITY-DIRECTIONS-OF-SCIENCE-AND-TECHNOLOGY-DEVELOPMENT-21-23.03.21.pdf>

10. Колумбет В.П., Барабаш О.В. Мультиагентний підхід для управління розподіленими обчисленнями на рівні GRID–системи. International scientific conference “*Innovation around us ‘2021*” March 9–10, 2021, Published by: SWorld in conjunction with D.A. Tsenov Academy of Economics, Svishtov, Bulgaria. С. 5 – 8. DOI: 10.30889/2709–183X.2021–07–00

<https://www.sworld.com.ua/konferbg7/cp-bg-7.pdf>

11. Barabash O.V., Svynchuk O.V., Kolumbet V.P. Simulation Modeling of Distributed Random Processes Using a Multiagent Approach. Conference proceedings. International scientific conference “*The current stage of development of scientific and technological progress'2021*”. February 8 – 9, 2021. Karlsruhe, Germany. P. 12 – 16.

<https://www.sworld.education/konferge15/sbor-ge15.pdf>

12. Колумбет В.П., Барабаш О.В. Розв’язання навігаційної задачі для агента в неоднорідному середовищі. Наукові праці Четвертої міжнародної науково–практичної конференції «*Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій*» 1–2 лютого 2022 р., м. Київ, НУХТ, 2022. С. 87 – 88.

http://kist.ntu.edu.ua/konferencii/31_konf_2022.pdf

Відомості про апробацію результатів дисертації

1. XIX міжнародна науково–практична конференція молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики» (м. Київ, 20 – 23 квітня 2021 р.); Форма участі – Дистанційно.

2. XIV міжнародна науково–практична конференція «Інтегровані інтелектуальні робототехнічні комплекси (ІРТК–2021)» (м. Київ, 18 – 19 травня 2021 р.); Форма участі – Дистанційно.

3. XVII Міжнародна науково–практична конференція «Військова освіта і наука: сьогодення та майбутнє» (м. Київ, 26 листопада 2021 р.); Форма участі – Дистанційно.

4. Міжнародна науково–практична інтернет–конференція «Розвиток освіти, науки та бізнесу: результати 2020» (м. Дніпро, 3 – 4 грудня 2020 р.); Форма участі – Дистанційно.

5. VII Міжнародна науково–практична конференція “Priority Directions of Science and Technology Development” (м. Київ, 21 – 23 березня 2021 р.); Форма участі – Дистанційно.

6. International scientific conference "Innovation around us '2021" (Svishtov, Bulgaria, 9 – 10 березня 2021 р.); Форма участі – Дистанційно.
7. International scientific conference "The current stage of development of scientific and technological progress'2021". (Karlsruhe, Germany, 8 – 10 лютого 2021 р.). Форма участі – Дистанційно.
8. Четверта міжнародна науково–практична конференція *«Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій»* 1–2 лютого 2022 р., м. Київ, НУХТ, 2022. Форма участі – Дистанційно.