

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Береговська Христина Василівна

Кваліфікаційна наукова
праця на правах рукопису

УДК 004.94+316.77:004

ДИСЕРТАЦІЯ

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ УПРАВЛІННЯ
АДАПТИВНОЮ СИСТЕМОЮ РОЗУМНОГО
БУДИНКУ**

122 – Комп'ютерні науки

Подається на здобуття наукового ступеня доктора філософії

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

_____/ Х.В. Береговська /
(підпис, ініціали та прізвище здобувача)

Науковий керівник:
Теслюк Василь Миколайович
доктор технічних наук, професор

Львів – 2025

АНОТАЦІЯ

Береговська Х.В. Інформаційна технологія управління адаптивною системою розумного будинку. — Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки». — Національний університет «Львівська політехніка», Львів, 2025.

Зміст анотації.

Дисертаційне дослідження присвячене розробленню інформаційної технології управління адаптивною системою розумного будинку (РБ), яка здатна діяти проактивно, реалізуючи випереджувальне керування, що базується на навчанні з використанням статистичних даних про поведінку користувача. Розроблені моделі та методи поєднують теоретико-графові та стохастичні підходи на основі мереж Петрі-Маркова (МПМ), доповнені функціональними компонентами (сенсорами, актуаторами), що забезпечує нову якість управління, реакції на події та персоналізацію в РБ.

У першому розділі «Аналіз моделей, методів і засобів інформаційних технологій управління системами “Розумний будинок”» виконано комплексний огляд підходів до побудови та моделювання сучасних «розумних» систем, їх основні функції та властивості. Досліджено переваги та недоліки ряду актуальних методів проєктування — класичних структурних (SADT, IDEF0, SSADM) та сучасних - імітаційних, аналітичних, інтелектуальних. Обґрунтовано, що існуючі методи не здатні забезпечити повноцінну поведінкову адаптацію та динамічне самонавчання систем РБ, що унеможливило реалізацію дій «на випередження». Виявлено обмеження у ПЗ, яке здебільшого реалізує лише сценарне управління без гнучкості пристосування до змін поведінки користувача. Проаналізовано технічні характеристики апаратної бази (мікроконтролери, ПЛК), що застосовуються в побутовій автоматизації, визначено потенціал їх використання в адаптивних системах. У результаті зроблено висновок про доцільність створення нової інформаційної технології

управління системами РБ на основі формалізованих стохастичних моделей, зокрема мереж Петрі-Маркова, які забезпечують інтеграцію подієвої логіки, ймовірнісного характеру взаємодій і механізмів навчання.

У другому розділі «Моделі адаптивної системи “Розумний будинок” на основі мереж Петрі-Маркова та доповнених функціональними компонентами» представлено основні теоретичні засади, на яких побудовано математичну модель РБ, направлену на підвищення оперативності реагування системи. В розділі обґрунтовано вибір методології побудови моделей системи РБ, що поєднує в собі структурно-динамічні властивості мереж Петрі з ймовірнісними характеристиками моделей Маркова. Представлено моделі взаємодії користувача з системою на основі сценаріїв функціонування РБ з врахуванням випадкової природи подій, часових залежностей та впливу зовнішнього середовища. При цьому значну увагу зосереджено на реалізації принципу роботи на випередження, коли система здатна активувати відповідні пристрої та переходити в певні стани ще до фактичного виконання дій користувачем. Для цього імплементовано механізм корекції коефіцієнтів ймовірностей переходів, використовується набір сенсорів та виконавчих пристроїв, гнучка декомпозиція на функціональні рівні. Побудовані графи досяжності станів підтверджують цілісність та скінченність моделей, демонструють їх здатність до адаптації та підтримки різноманітних сценаріїв поведінки користувача.

У третьому розділі «Методи синтезу та навчання моделей систем “Розумний будинок” на основі моделей Петрі-Маркова та доповнених функціональними компонентами» розроблено метод синтезу моделей систем РБ на основі мереж Петрі-Маркова (МПМ), доповнених функціональними компонентами, а також метод їх навчання для реалізації адаптивності цих систем. Обґрунтовано вибір МПМ як формалізму, що дозволяє відображати як динаміку сценаріїв роботи, так і ймовірнісну природу взаємодії системи з користувачем. Запропонована модель забезпечує можливість багаторівневої декомпозиції системи — від високорівневого формулювання сценаріїв до рівня фізичних компонентів - сенсорів та актуаторів. Такий підхід дає змогу

моделювати динаміку взаємодії користувача з системою, враховувати стохастичний характер подій, а також верифікувати ступінь детермінованості початкових і кінцевих станів сценаріїв. Запропоновано метод навчання моделей, який базується на зборі статистичних даних про поведінку користувача та подальшому коригуванні ймовірнісних параметрів моделі на основі зібраної в процесі роботи системи, інформації. Розглянуто три режими збору статистичних даних: ручний, автоматизований та автоматичний. Проведено аналіз кожного з них. Сформульовано базовий алгоритм навчання, що включає побудову матриць інцидентності, перехідних ймовірностей та адаптацію розподілу ймовірностей у сценаріях. Описано механізм реалізації «дії на випередження» як здатності системи ініціювати події завчасно, спираючись на історичні патерни поведінки користувача. Розроблено структуру зберігання моделі, що дозволяє оптимізувати обробку сценаріїв, здійснити перехід від громіздкої багатовимірної до одновимірної структури з уніфікованою нумерацією та єдиним репозиторієм реляцій. З цією метою реалізовано алгоритми конвертації між лінійним та ієрархічним представленням моделей. Запропоновані методи та алгоритми становлять основу для створення адаптивної системи розумного будинку, яка здатна пристосовувати свою поведінку до реальних умов експлуатації, навчаючись на основі накопиченого досвіду.

У четвертому розділі «Розроблення засобів інформаційної технології управління адаптивною системою “Розумний будинок” на основі моделей Петрі-Маркова і доповнених функціональними компонентами та результати дослідження» представлено практичну реалізацію розробленої інформаційної технології управління адаптивною системою розумного будинку. Запропоновано повноцінну архітектуру програмно-апаратного комплексу разом з алгоритмами функціонування, технічними рішеннями та інструментами моніторингу й адміністрування. Основою цієї інформаційної технології є модуль адаптивного реагування, що забезпечує обробку вхідних даних, їх перетворення у внутрішні сигнали, формування сценаріїв на основі моделей і виконання відповідних дій. Ключовою складовою комплексу виступає спеціалізований мікроконтролер,

який реалізує моніторинг подій, аналіз сценаріїв взаємодії з користувачем та запуск виконавчих пристроїв. Для цього розроблено алгоритмом імовірнісного прогнозування зміни станів, що дає змогу реалізовувати випереджувальне реагування на дії користувача. Для реалізації фізичної частини системи, запропоновано низьковартісне та енергоефективне технічне рішення на основі мікроконтролерів STM8 та ESP8266. Розроблено структурну схему контролера адаптивної системи РБ, схеми підключення сенсорів та актуаторів, а також алгоритми реагування на події. Для забезпечення безперервного функціонування та навчання системи імплементовано модуль статистики, який збирає дані про події, послідовність їх виникнення та результати дії виконавчих елементів. У рамках роботи реалізовано спеціалізоване програмне забезпечення - автоматизовану програмну систему адміністрування та моніторингу (АПСАМ). Вона виконує імпорт конфігураційних файлів, візуалізацію просторової структури будівлі, динамічне оновлення сценаріїв та журналізацію подій. Система забезпечує зв'язок з мікроконтролером та дозволяє адмініструвати компоненти з можливістю тестування у змодельованому середовищі. У розділі також наведено успішні результати проведеного тестування системи відповідно до стандартів, описаних у ISTQB СТ-AI, які базуються на обчисленні основних метрик.

Основні наукові результати дисертації опубліковано в 25-ти працях, зокрема: дев'ять статей – у наукових фахових періодичних виданнях України; дві статті – у наукових періодичних виданнях інших держав; чотирнадцять публікацій – у матеріалах міжнародних науково-технічних конференцій.

Ключові слова: інформаційна технологія, адаптивна система, управління системою, адаптивність, розумний будинок, мережі Петрі, моделі Маркова, розумні технології, моделі Петрі-Маркова, ймовірнісне прогнозування, спеціалізований мікроконтролер, автоматизована програмна система моніторингу, навчання моделей.

SUMMARY

Beregovska K.V. Information Technology for Managing an Adaptive Smart Home System. — Qualification research manuscript. Dissertation for the degree of Doctor of Philosophy in specialty 122 "Computer Sciences". — Lviv Polytechnic National University, Lviv, 2025.

Abstract content. The dissertation is devoted to the development of information technology for managing an adaptive Smart Home (SH) system capable of proactive operation by implementing anticipatory control based on training using statistical data on user behavior. The developed models and methods combine graph-theoretical and stochastic approaches based on Petri-Markov Networks (PMN), supplemented with functional components (sensors, actuators), which ensures a new quality of control, event response, and personalization in smart home systems.

The first chapter, "Analysis of Models, Methods, and Tools of Information Technologies for Smart Home System Management", provides a comprehensive overview of existing approaches to modeling and implementation of smart systems, their functions and properties. The advantages and disadvantages of various methods — classical structural (SADT, IDEF0, SSADM), as well as simulation, analytical, and intelligent ones — are discussed. It is substantiated that existing methods do not provide full behavioral adaptation and dynamic self-learning, which limits the ability of smart home systems to operate proactively. Software limitations are identified, as most solutions only offer static scenario-based control. The chapter also analyzes hardware platforms (microcontrollers, PLCs) used in home automation and evaluates their potential for adaptive applications. The need to develop a new information technology based on formal stochastic models — particularly Petri-Markov Networks — is emphasized, given their ability to integrate event logic, probabilistic transitions, and learning mechanisms.

The second chapter, "Models of an Adaptive Smart Home System Based on Petri-Markov Networks Supplemented with Functional Components", presents the theoretical foundations of the mathematical model aimed at improving system

responsiveness. The proposed modeling methodology integrates structural-dynamic features of Petri Nets with the probabilistic nature of Markov Chains. User interaction scenarios with the system are modeled, considering the random nature of events, temporal dependencies, and environmental factors. Emphasis is placed on enabling the system to activate components and switch states *before* a user action is fully completed. This is achieved through transition probability adjustment mechanisms, multi-level decomposition, and integration of sensors and actuators. Reachability graphs are used to validate the consistency and completeness of the models and demonstrate their adaptability to various user scenarios.

The third chapter, "Methods for Synthesis and Learning of Smart Home System Models Based on Petri-Markov Networks and Functional Components", develops methods for synthesizing and training models of an adaptive smart home system. PMNs are justified as a suitable formalism for modeling both scenario dynamics and probabilistic interaction. The model supports multilevel decomposition — from high-level user behavior to the physical level of sensors and actuators. A learning method based on the accumulation of user interaction statistics is proposed, enabling the model to adjust its probabilistic parameters over time. Three data collection modes are explored: manual, semi-automated, and fully automated. The basic learning algorithm includes the construction of incidence and transition probability matrices and the adaptation of probability distributions within scenarios. A specific mechanism of anticipatory action initiation is described. A compact storage structure is developed, optimizing performance by transitioning from a multidimensional to a linear unified structure. Conversion algorithms between hierarchical and linear representations are implemented. These tools provide a basis for creating an adaptive system that evolves based on real-world usage.

The fourth chapter, "Development of Tools for Adaptive Smart Home Control Information Technology Based on Petri-Markov Networks and Functional Components, and Research Results", presents the practical implementation of the developed adaptive SH management system. The complete architecture of the hardware-software complex is presented, including algorithms, monitoring tools, and

administration systems. A central component is a specialized microcontroller that monitors events, analyzes user interaction scenarios, and triggers actuators. A probabilistic forecasting algorithm is implemented, enabling the system to respond in advance of user actions. A low-cost, energy-efficient technical solution based on STM8 and ESP8266 microcontrollers is proposed. Detailed circuit diagrams, event response algorithms, and control structures are described. A statistical module ensures continuous learning by collecting and processing data about system operation and user activity.

An automated software system for administration and monitoring (APSAM) is developed. It supports configuration file import, building layout visualization, dynamic scenario updates, and event logging. The system communicates with the microcontroller and allows simulation-based testing of system components. Testing results, based on ISTQB CT-AI standards, confirm the system's effectiveness. Key performance metrics indicate high levels of prediction accuracy and responsiveness.

The main scientific results of the dissertation have been published in 25 works, including: nine articles in specialized scientific journals of Ukraine, two articles in international journals, and fourteen publications in proceedings of international scientific and technical conferences.

Keywords: information technology, adaptive system, system management, adaptability, smart home, Petri nets, Markov models, smart technologies, Petri-Markov networks, probabilistic forecasting, specialized microcontroller, automated monitoring system, model learning.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті в наукових фахових виданнях України:

1. Береговська Х., Теслюк В. Метод синтезу моделей для адаптивних систем розумного будинку з використанням мереж Петрі-Маркова. *Комп'ютерні технології друкарства*. 2024. Т.51, №1. С. 61–77. doi:[10.32403/2411-9210-2024-1-51-61-77](https://doi.org/10.32403/2411-9210-2024-1-51-61-77)
2. Береговська Х., Теслюк В. Моделі адаптивної системи розумного будинку на базі мереж Петрі та моделей Маркова. *Науковий вісник НЛТУ України*. 2024. Т. 34, № 6. С. 115–124. doi: [10.36930/40340616](https://doi.org/10.36930/40340616)
3. Teslyuk V., Beregovska K., Zerbino D., Teslyuk T., Seneta M. Universal controller for the distributed management in the adaptive smart home systems. *Український журнал інформаційних технологій*. 2024. Вип. 6, т. 2. С. 64–73. doi: [10.23939/ujit2024.02.064](https://doi.org/10.23939/ujit2024.02.064)
4. Береговська Х. В., Машевська М. В., Теслюк В. М. Розроблення моделей систем "Інтелектуальний будинок", побудованих на базі моделей Петрі-Маркова та доповнених функціональними компонентами. *Моделювання та інформаційні технології*. 2017. Вип. 78. С. 175–185. http://nbuv.gov.ua/UJRN/Mtit_2017_78_27
5. Береговська Х. В., Машевська М. В., Зелінський А. Я., Теслюк В. М. Розроблення методу побудови моделей систем "Інтелектуальний будинок", створених на базі моделей Петрі-Маркова та доповнених функціональними компонентами. *Моделювання та інформаційні технології*. 2017. Вип.79. С. 176–182. http://nbuv.gov.ua/UJRN/Mtit_2017_79_26
6. Теслюк В. М., Борецько О. Ю., Сидор А. Р., Береговська Х. В. Модель телекомунікаційної моделі мережі «інтелектуального будинку». *Науковий вісник НЛТУ України*. 2016. Т. 26, № 1. С. 351–357. doi: [10.15421/40260154](https://doi.org/10.15421/40260154)
7. Береговська Х. В., Коваль В. Я., Теслюк В. М. Розроблення спеціалізованого мікроконтролера моніторингу та імовірнісного прогнозування зміни станів систем «Розумний будинок» // *Моделювання*

- та інформаційні технології : збірник наукових праць. – Вип. 80. – Київ : Інститут моделювання в енергетиці ім. Г. Є. Пухова НАН України, 2017. – С. 121–129. <http://jnas.nbu.gov.ua/article/UJRN-0000855902>
8. Береговська Х. В., Теслюк В. М. Баран М. М. Розроблення автоматизованої програмної системи адміністрування та моніторингу систем "Розумний будинок". *Моделювання та інформаційні технології*. 2017. Вип. 81. С. 114–122. http://nbuv.gov.ua/UJRN/Mtit_2017_81_18
9. Теслюк В. М., Береговська Х. В. Угрин Л. Є. Розроблення методу навчання моделей систем "розумного будинку" на базі моделей Петрі-Маркова та доповнених функціональними компонентами. *Науковий вісник НЛТУ України*. 2017. Т. 27, № 6. С. 139–144. [doi: 10.15421/40270628](https://doi.org/10.15421/40270628)

Статті у наукових виданнях інших держав:

10. Teslyuk V., Beregovska K., Holovatyy A. Analysis of the current state of progress and development of "smart" house systems. *Innovation Technologies in Economy and Society* : series of monographs, monogr. 20. Katowice : Wydawn. Wyższej Szkoły Technicznej w Katowicach, 2018. P. 88–105. <http://www.wydawnictwo.wst.pl/uploads/files/10ce27dda4eab6e1ad0c1e55ca02e9cd.pdf>
11. Kazarian A., Beregovska K., Teslyuk V. Data analysis model and forms of cloud analytical functions for "smart" house systems. *Information and innovation technologies in economics and administration* : monograph / ed. by O. Chukurna, M. Gawron-Łapuszek. Katowice: Wydawnictwo Wyższej Szkoły Technicznej w Katowicach, 2019. P. 17–27. <http://www.wydawnictwo.wst.pl/uploads/files/2979ee6e305fe7fcd478539e48bb1fcd.pdf>

Матеріали конференцій:

12. Beregovska K., Teslyuk V., Beregovskyi V., Kazymyra I., Fabri L. Model and tools of adaptive control of a smart home system. *CEUR Workshop Proceedings*. 2021. Vol. 2917 : Modern machine learning technologies and data science

- workshop: proc. of the 3rd Intern. workshop MoMLeT&DS 2021, Lviv, Shatsk, Ukraine, June 5-6, 2021. Vol. I: main conf. P. 263–272 (Scopus) <https://ceur-ws.org/Vol-2917/paper23.pdf>
13. Teslyuk V., Denysyuk P., Kryvinska N., Beregovska K., Teslyuk T. Neural controller for smart house security subsystem. *Procedia Computer Science*. 2019. Vol. 160 : 10th Intern. conf. on emerging ubiquitous systems and pervasive networks EUSPN-2019, 9th Intern. conf. on current and future trends of information and communication technologies in healthcare ICTH-2019)/ Affiliated workshops. P. 394–401 (Scopus, Web of Science) doi: 10.1016/j.procs.2019.11.075
 14. Teslyuk V., Beregovska K., Denysyuk P. Decomposition of models of smart-home systems. *Perspective technologies and methods in MEMS design (MEMSTECH)* : proc. of XIIIth Intern. conf., Polyana, Ukraine, Apr. 20–23, 2017. 2017. P. 22–24. (Scopus, Web of Science) [doi: 10.1109/MEMSTECH.2017.7937524](https://doi.org/10.1109/MEMSTECH.2017.7937524)
 15. Teslyuk V., Denysyuk P., Beregovska K., Mashevskaya M. Method of development of smart-house-systems models based on Petri-Markov nets and extended by functional components. *Computer Science and Information Technologies* : proc. of the XII Intern. sci. and techn. conf. CSIT'2017, Lviv, Ukraine, Sept. 5–8, 2017. 2017. P. 352–355. (Scopus, Web of Science) [doi: 10.1109/STC-CSIT.2017.8098803](https://doi.org/10.1109/STC-CSIT.2017.8098803)
 16. Boreiko O., Teslyuk V., Beregovska K., Mykhailiuk A. Model of telecommunication networks for intelligent building. *The Experience of Designing and Application of CAD Systems in Microelectronics* : proc. of 13th Intern. conf, CADSM 2015, Lviv-Polyana, Ukraine, 24-27 Febr. 2015. P. 349–351. (Scopus, Web of Science) [doi: 10.1109/CADSM.2015.7230873](https://doi.org/10.1109/CADSM.2015.7230873)
 17. Береговська Х. В. Автоматизований програмно-апаратний комплекс для систем «розумний будинок» на основі моделей Петрі-Маркова, доповнених функціональними компонентами // Інтелектуальні комп'ютерні

- системи та мережі : збірник праць IX Науково-практичної конференції молодих вчених і студентів (Тернопіль, 21 травня 2024 р.). – 2024. – С. 51
18. Береговська Х. В., Рутецький Ю. О. Моделі та засоби адаптивної системи розумного будинку // Інтелектуальні комп'ютерні системи та мережі : збірник праць VIII Науково-практичної конференції молодих вчених і студентів, Тернопіль, 5 грудня, 2023. – С. 52–53.
https://ki.wunu.edu.ua/conference/archive/2023_2.pdf
 19. Х. Береговська, В. Теслюк Інтеграція сенсорів і актюаторів в моделі систем «інтелектуальний будинок» // Проблеми та перспективи розвитку економіки і підприємництва та комп'ютерних технологій в Україні : XIII наук.-практ. конф., Львів, 27–31 березня 2017 р. – С. 19–21.
 20. Береговська Х., Теслюк В. Розроблення моделей систем «Інтелектуальний будинок» на базі моделей Маркова // Інформаційне суспільство: тенденції регіонального розвитку : міжнар. наук.-практ. конф. ISRDT-2016, Львів, 20–21 вересня 2016. – С. 8–9.
 21. Ю. Вербіцький, Х. Береговська Система безпеки інтелектуального будинку // V Всеукр. школа-семінар «Сучасні комп'ютерні інформаційні технології», АСІТ'2015, Тернопіль, 22–23 травня 2015 р.
 22. Boreiko O. Video Monitoring System's Module for Smart Home / Boreiko Oleg, Teslyuk Vasyl, Berezhsky Oleg, Beregovska Cristina // Proceeding of the 10th International Conference "Perspective Technologies and Methods in MEMS Design" (MEMSTECH'2014), June 22-24, 2014. – Lviv : Lviv Polytechnic Publishing House, 2014. – P. 126-128. <https://orcid.org/0000-0002-1556-8753>
 23. Боре́йко, О. Ю. Модель комп'ютерної мережі інтелектуального будинку з використанням одноплатних комп'ютерів Raspberry Pi / О. Ю. Боре́йко, Х. В. Береговська, В. М. Теслюк // Сучасні комп'ютерні інформаційні технології : матеріали IV Всеукр. шк.-семінару молодих вчен. і студ. АСІТ'2014 - Тернопіль, 16-17 трав. 2014 р. / відп. за вип. М. П. Дивак. - Тернопіль : ТНЕУ, 2014. - С. 48-51.

<http://dspace.tneu.edu.ua/handle/316497/6719>

24. В. Теслюк, П. Денисюк, Х. Береговська Використання технологій розумного будинку для поліпшення енергетичної ситуації в Україні // XIII міжнар. наук. семінар «Сучасні проблеми інформатики в управлінні, економіці, освіті». – Київ, 2014. – С. 215–219.
<https://nam.kyiv.ua/files/publications/978-966-8406-88-1-konf-compressed.pdf>
25. В. Теслюк, Н. Дудкіна, Х. Береговська Розроблення структури, алгоритму та моделі роботи підсистеми моніторингу інтелектуального будинку // IV Всеукр. школа-семінар АСІТ'2014. – Тернопіль : ТНЕУ
<http://dspace.tneu.edu.ua/handle/316497/6697>

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	17
ВСТУП.....	18
РОЗДІЛ 1 АНАЛІЗ МОДЕЛЕЙ, МЕТОДІВ І ЗАСОБІВ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ УПРАВЛІННЯ СИСТЕМАМИ «РОЗУМНИЙ БУДИНОК»	27
1.1 Аналіз моделей та методів розроблення систем «Розумний будинок»...	28
1.2 Аналіз програмного забезпечення для роботи з системами «Розумний будинок»	41
1.3 Аналіз апаратного забезпечення систем «Розумний будинок»	47
1.4 Аналіз підходів до визначення активностей в системах «Розумний будинок»	49
1.5 Формулювання задач дисертаційного дослідження	50
Висновки до розділу 1	51
РОЗДІЛ 2. МОДЕЛІ АДАПТИВНОЇ СИСТЕМИ «РОЗУМНИЙ БУДИНОК» НА ОСНОВІ МЕРЕЖ ПЕТРІ-МАРКОВА ТА ДОПОВНЕНИХ ФУНКЦІОНАЛЬНИМИ КОМПОНЕНТАМИ.....	53
2.1. Модель взаємодії власника та адаптивної системи «Розумний будинок»	53
2.2. Моделі системи РБ для підвищення оперативності реагування системи «Розумний будинок», побудованої на основі моделі Маркова.....	55
2.3 Моделі системи РБ з врахуванням функціональних компонентів системи на основі мереж Петрі та моделей Маркова	63
2.4. Побудова графа досяжності станів для розроблених моделей	73
Висновки до розділу 2.....	76

РОЗДІЛ 3. МЕТОДИ СИНТЕЗУ ТА НАВЧАННЯ МОДЕЛЕЙ СИСТЕМ
«РОЗУМНИЙ БУДИНОК» НА ОСНОВІ МОДЕЛЕЙ ПЕТРІ-МАРКОВА ТА
ДОПОВНЕНИХ ФУНКЦІОНАЛЬНИМИ КОМПОНЕНТАМИ..... 78

3.1. Розроблення методу синтезу моделей систем РБ, створених на основі моделей Петрі-Маркова та доповнених функціональними компонентами	78
3.2. Розроблення методу навчання моделей систем РБ, створених на основі моделей Петрі-Маркова та доповнених функціональними компонентами	86
3.3. Розроблення алгоритму реалізації методу навчання моделей систем РБ, створених на основі МПМ та доповнених функціональними компонентами	98
3.4. Оптимізація алгоритму навчання моделей систем РБ на основі МПМ та доповнених функціональними компонентами	101
Висновки до розділу 3	108

РОЗДІЛ 4 РОЗРОБЛЕННЯ ЗАСОБІВ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ
УПРАВЛІННЯ АДАПТИВНОЮ СИСТЕМОЮ «РОЗУМНИЙ БУДИНОК» НА
ОСНОВІ МОДЕЛЕЙ ПЕТРІ-МАРКОВА І ДОПОВНЕНИХ
ФУНКЦІОНАЛЬНИМИ КОМПОНЕНТАМИ
ТА РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Розроблення структури інформаційної технології управління адаптивною системою «Розумний будинок» на основі моделей Петрі- Маркова та доповнених функціональними компонентами	110
4.2 Загальна структура автоматизованого програмно-апаратного комплексу	111
4.3 Алгоритм функціонування та структура спеціалізованого мікроконтролера	116

4.4 Алгоритм імовірнісного прогнозування зміни станів для адаптивної системи РБ.....	119
4.5 Особливості розроблення технічного забезпечення адаптивної системи РБ.....	123
4.5.1 Загальна структура технічного забезпечення адаптивної системи РБ.....	123
4.5.2 Особливості реалізації технічного забезпечення контролера.....	125
4.5.3 Особливості розроблення спеціалізованого програмного забезпечення контролера адаптивної системи РБ.....	134
4.6 Розроблення автоматизованої програмної системи адміністрування та моніторингу адаптивної системи РБ.....	136
4.7 Результати моделювання	144
4.8. Тестування системи.....	151
Висновки до розділу 4.....	155
ЗАГАЛЬНІ ВИСНОВКИ ДО РОБОТИ	157
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	160
ДОДАТКИ	173

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

АПСАМ	- Автоматизована програмна система адміністрування та моніторингу
БДС	- База даних статистики
ЕОМ	- Електронно-обчислювальна машина
ІТ	– Інформаційні технології
КАСРБ	- Контролер адаптивної системи розумного будинку
ММ	- Мережі Маркова
МП	- Мережі Петрі
МПМ	- Мережі Петрі-Маркова
ОС	- Операційна система
ПЗ	- Програмне забезпечення
ПЛК	- Програмовані логічні комплекси
РБ	– Розумний будинок
ШІ	- Штучний інтелект
ШНМ	– Штучні нейронні мережі
DFM	- Data Flow Modeling
IDEF0	- Integrated Definition for Function Modeling
IoT	- Internet of Things
ISTQB CT-AI	- International Software Testing Qualifications Board Certified Tester - AI Testing
LDM	- Logical Data Modeli
SADT	- Structured analysis and design technique
SSADM	- Structured Systems Analysis and Design
UI	- User Interface
UML	– Unified Modeling Language

ВСТУП

Актуальність теми. Однією з визначальних рис сучасного суспільства є стрімке зростання темпу життя. З одного боку, люди прагнуть максимально ефективно використовувати свій робочий час, з іншого — вивільнити більше часу для особистих потреб, відпочинку, самореалізації та дозвілля. Одним із дієвих засобів досягнення цієї мети є автоматизація процесів, як виробничих, так і побутових, які тісно пов'язані з повсякденною людською діяльністю.

Реалізацією ідеї автоматизації побуту є системи «розумний будинок» (РБ), які покликані не лише полегшити виконання рутинних дій, а й підвищити рівень енергоефективності, безпеки та загального комфорту користувача. У рамках цієї дисертаційної роботи основна увага зосереджена на реалізації функції вивільнення продуктивного часу користувача за рахунок інтелектуалізації процесів керування побутовими пристроями.

На ранніх етапах розвитку системи РБ були орієнтовані переважно на дистанційне або автоматизоване виконання окремих дій — таких як керування освітленням, опаленням чи побутовими електроприладами. Їх реалізація базувалася на сценаріях, жорстко закладених у програмну логіку. Проте зі зростанням вимог до комфорту, індивідуалізації та адаптивності постала потреба у створенні нових типів РБ — таких, що здатні до самонавчання, урахування поведінкових патернів користувачів і прийняття рішень на основі накопиченого досвіду.

Хоч ідея розумного будинку виникла ще у 1950-х роках у США, вона залишається актуальною й сьогодні. Це пояснюється складністю таких систем, необхідністю їх гнучкої адаптації до конкретного середовища та користувача, постійною потребою у переналаштуванні та вдосконаленні алгоритмів управління. Саме ці фактори зумовлюють високі вимоги до проєктування, реалізації та обслуговування систем РБ.

Сучасні комерційні рішення, хоча й демонструють значні досягнення в автоматизації житлового простору, як правило, не забезпечують повноцінної

поведінкової адаптації. Більшість з них реалізують сценарне управління без врахування ймовірнісної природи змін поведінки користувача. Це істотно обмежує функціональні можливості РБ, зокрема в аспектах передбачення дій користувача та динамічного пристосування до них.

Незважаючи на активний розвиток технологій «розумного будинку», на сьогодні відсутні усталені методи та засоби моделювання таких систем, які б дозволяли формалізовано та кількісно описувати взаємодію з користувачем, ураховуючи ймовірнісний характер подій. Особливо актуальним є завдання розроблення моделей, здатних ініціювати дії ще до фактичного настання відповідних подій — тобто діяти проактивно. Реалізація таких підходів є передумовою створення систем нового класу — адаптивних, навчальних, здатних ефективно співпрацювати з користувачем і діяти на випередження.

Значний внесок в галузі дослідження процесів управління адаптивними системами розумного будинку зробили: Ioan Doré Landau — французький науковець, один із засновників теорії адаптивного управління та ідентифікації систем; MengChu Zhou — професор кафедри електротехніки та комп'ютерної інженерії Технологічного інституту Нью-Джерсі (NJIT), роботи якого охоплюють моделювання, аналіз та синтез автоматизованих систем, включаючи розумні будинки. Зокрема, він розробив методи для запобігання тупикам у складних системах, є одним із найвідоміших дослідників у галузі мереж Петрі та дискретних подієвих систем; Sajal K. Das — професор Технологічного інституту Міссурі, що спеціалізується на бездротових сенсорних мережах, мобільних обчисленнях та розумних середовищах. Його дослідження охоплюють різні аспекти розумних будинків, включаючи сенсорні мережі та адаптивні системи; Franco Zambonelli — італійський вчений, професор комп'ютерних наук в Університеті Модени та Реджо-Емілії (Італія). Він відомий своїм внеском у галузі розподілених систем, багатоагентних технологій, самонавчальних та самоналаштовуваних систем, а також інтернету речей (IoT) і розумних середовищ; Mariagrazia Dotoli — італійська вчена,

професорка в галузі систем і управління, відома своїми дослідженнями в автоматизації, оптимізації енергетичних систем, логістиці, розумних містах та управлінні розподіленими енергетичними ресурсами. Серед українських науковців, що працюють в цій сфері варто згадати Валерія Семенця – українського вченого у галузі автоматизованого проєктування технічних систем, зокрема з використанням марковських процесів; Вячеслава Хареченка – українського вченого у галузі надійності та кібербезпеки хмарних та IoT-систем. Його дослідження охоплюють комбінування марковських та напівмарковських моделей для оцінки доступності та кібербезпеки систем; Ярославу Бойко - експертку та координаторку в галузі цифрових технологій та міських інновацій, яка відіграла ключову роль у розвитку проєктів Kyiv Smart City з 2015 по 2020 роки. Вона сприяла впровадженню концепції «розумного міста» в столиці України; Леоніда Файнзильберга - професора кафедри біомедичної кібернетики Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», який досліджує методи та системи штучного інтелекту, зокрема в контексті обробки сигналів та інтелектуальних інформаційних технологій.

Таким чином, розроблення інформаційної технології управління адаптивною системою розумного будинку на основі формалізованих стохастичних моделей, зокрема мереж Петрі-Маркова, є актуальною науковою проблемою, вирішення якої дозволить підвищити якість та функціональність сучасних автоматизованих побутових систем.

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

Дисертація виконувалася відповідно до пріоритетних напрямів науково-дослідних робіт Національного університету «Львівська політехніка» та відповідно до координаційних планів Міністерства освіти й науки України. Зокрема, наукові дослідження виконувалися в рамках держбюджетних наукових тем кафедри автоматизованих систем управління: «Інтелектуальні інформаційні технології багаторівневого управління енергоефективністю

регіону» (номер державної реєстрації 0117U004450, роки виконання: 2017-2018); «Методи та засоби нейронечіткого управління групою мобільних робототехнічних платформ» (номер державної реєстрації: 0123U101688, роки виконання: 2023-2024); «Методи та засоби інтелектуального вимірювання параметрів руху та визначення просторової орієнтації наземних мобільних робототехнічних платформ» (номер державної реєстрації: 0124U000822, роки виконання: 2024-2026). Документи, що підтверджують впровадження результатів дисертаційного дослідження надано у додатку 3.

Метою дисертаційної роботи є розроблення методів та засобів розроблення якісно нових моделей систем «Розумний будинок», які б дали змогу реалізувати механізм адаптації системи до особливостей характеру та поведінки конкретних користувачів, забезпечивши тим самим високий рівень комфорту та досягнення поставлених перед даними системами цілей.

Для розв'язання поставленої наукової задачі необхідно виконати такі завдання:

1. Здійснити аналіз методів, моделей і засобів інформаційних технологій управління системами «Розумний будинок».

2. Розробити моделі для системи розумного будинку, які ґрунтуються на використанні теорії мереж Петрі-Маркова та доповнені функціональними компонентами, що дає змогу врахувати стани та ймовірнісні процеси, що виникають в результаті взаємодії системи управління розумним будинком і користувачем.

3. Розробити метод навчання моделей систем РБ, який би забезпечив механізм адаптації проектованої системи управління до особливостей поведінки користувачів системи РБ.

4. Розробити метод автоматизованого синтезу моделей для системи розумного будинку, які використовують теорію мереж Петрі-Маркова та доповнені функціональними компонентами.

5. Розробити модель управління системою розумного будинку, яка

використовує стохастичні мережі Петрі та забезпечує врахування випадкових процесів.

6. Розробити інформаційну технологію управління адаптивною системою розумного будинку на основі використання теорії мереж Петрі-Маркова.

7. Розробити автоматизований програмно-апаратний комплекс дослідження, адміністрування та моніторингу проєктованих систем РБ, а також реалізації наведених вище концепцій кількісної оцінки ймовірнісних процесів, що виникають в результаті взаємодії системи з користувачем та механізму адаптації проєктованої системи РБ до особливостей поведінки користувачів системи.

Об’єкт дослідження – процеси управління адаптивною системою РБ.

Предмет дослідження – методи та засоби управління адаптивною системою розумного будинку.

Методи дослідження: для розробки методів, моделей, засобів і алгоритмів – теорія системного аналізу, теорія мереж Петрі, теорія моделей Маркова, теорія моделей Петрі-Маркова; при розробці програмних моделей, програмного та апаратного забезпечення – принципи структурного програмування та методологія структурного аналізу та проєктування.

Наукова новизна одержаних результатів:

вперше розроблено:

– моделі для системи управління розумним будинком, які використовують теорію мереж Петрі-Маркова та доповнених функціональними компонентами, що дає змогу реалізувати сценарії функціонування та функцію адаптації системи управління розумним будинком до поведінки та вимог його власника;

– метод навчання моделей для системи управління розумним будинком, який використовує розроблені режими збору даних (ручний, автоматизований, автоматичний) та етапи опрацювання статистичної інформації та подальше

налаштування моделей, що дає змогу здійснити адаптацію розроблених моделей до поведінки користувача;

вдосконалено метод синтезу моделей для системи управління розумним будинком, який використовує теорію мереж Петрі-Маркова та розроблені алгоритми, що дає змогу автоматизувати процес побудови адаптивних моделей;

отримала подальший розвиток модель для системи управління, яка, на відміну від існуючих, використовує теорію стохастичних мереж Петрі та дає змогу врахувати випадкові процеси та стани складових розумного будинку.

Практичне значення одержаних результатів.

Розроблено інформаційну технологію управління адаптивною системою РБ, побудовану на основі моделей Петрі-Маркова, доповнених функціональними компонентами. Запропонована структура забезпечує опрацювання вхідних даних від сенсорів, оцінювання ймовірностей переходів між станами та ініціалізацію керуючих дій, базуючись на адаптивному сценарному підході.

Розроблений алгоритм функціонування та структура спеціалізованого мікроконтролера моніторингу та ймовірнісного прогнозування зміни станів систем «Розумний будинок», що забезпечує автоматичний моніторинг проектованої системи РБ, зчитування вхідної інформації від сенсорів а також автоматичне надсилання керуючих сигналів до актуаторів, з паралельною обробкою та збереженням всієї цієї інформації в базі даних статистики в якості вхідних даних для розробленої автоматизованої програмної системи адміністрування та моніторингу систем «Розумний будинок».

Розроблене технічне забезпечення контролера адаптивної системи РБ, яке базується на використанні мікроконтролера STM8, що забезпечує низьку ціну пристрою та можливість зміни функціональності мікроконтролера.

Розроблене програмне забезпечення адаптивної системи РБ, яке реалізовано на мові Assembler мікроконтролера STM8, що забезпечує високу швидкодію функціонування пристрою.

Розроблена автоматизована програмна система адміністрування та моніторингу систем РБ, призначена для зчитування вхідної інформації від мікроконтролера моніторингу та ймовірнісного прогнозування зміни станів систем РБ або з зовнішнього архівного файлу для моделювання конкретних сценаріїв, або ж вхідного потоку в режимі онлайн, здійснюючи таким чином моніторинг проектованої системи РБ. В розробленій системі реалізована можливість адміністрування основних параметрів системи.

Результати дослідження, а саме: оцінка якості адаптивної класифікаційної моделі відповідно до метрик, описаних у ISTQB CT-AI; параметри моделювання, які демонструють здатність системи до випереджального реагування та графі досяжності станів розроблених моделей, які демонструють скінченність і досяжність усіх станів та дають змогу дослідити динаміку функціонування моделей.

Результати дисертаційних досліджень впроваджені в навчальний процес кафедри автоматизованих систем управління Національного університету «Львівська політехніка» для викладання наступних дисциплін: «Основи смарт-технологій і систем» та «Моделювання процесів і смарт-систем» (підтверджено актом впровадження у додатку 3).

Особистий внесок здобувача.

Усі наукові результати дисертаційної роботи отримані автором самостійно. Особисто здобувачеві належать такі наукові результати: розроблено метод синтезу моделей для адаптивних систем розумного будинку з використанням мереж Петрі-Маркова [1]; розроблено моделі адаптивної системи розумного будинку на основі мереж Петрі та моделей Маркова [2]; розроблено контролер для розподіленого керування в адаптивних системах розумного будинку [3]; розроблено моделі систем «Інтелектуальний будинок»

побудованих на основі моделей Петрі-Маркова та доповнених функціональними компонентами [4], [15], [19]; розроблено методу побудови моделей систем «Інтелектуальний будинок», створених на основі моделей Петрі-Маркова та доповнених функціональними компонентами [5]; розроблено структурну модель на основі теорії мереж Петрі [6]; розроблено спеціалізований мікроконтролер моніторингу та імовірнісного прогнозування зміни станів систем «Розумний будинок» [7]; розроблено автоматизовану програмну систему адміністрування та моніторингу систем «Розумний будинок» [8], [17]; розроблено метод навчання моделей систем розумного будинку на основі моделей Петрі-Маркова та доповнених функціональними компонентами [9]; проведено аналіз сучасного стану розвитку систем РБ, аналіз методів та моделей автоматизації їх проєктування [10], [12], [18], [24]; розроблено алгоритм формування шаблонів для системи правил, пов'язаних з керуванням пристроями [11]; розроблено математичну модель підсистеми безпеки розумного будинку [13], [21]; розроблено механізм декомпозиції моделей проєктованих систем РБ [14]; розроблено математичну модель телекомунікаційної мережі для розумної будівлі [16]; розроблено модель системи «Інтелектуальний будинок» на основі моделі Маркова [20]; розроблено математичний алгоритм для модуля моніторингу в розумній будівлі [22], [25]; розроблено структурну модель на основі мереж Петрі [23]

Апробація результатів дисертації.

Основні наукові та практичні результати роботи доповідались та обговорювались на міжнародних наукових та науково-технічних конференціях:

9-й науково-практичній конференції «Інтелектуальні комп'ютерні системи та мережі» (2024), 8-й науково-практичній конференції «Інтелектуальні комп'ютерні системи та мережі» (2023), 3-й міжнародній конференції «Modern Machine Learning Technologies and Data Science Workshop» (MoMLet&DS 2021), 10-й міжнародній конференції «Emerging

Ubiquitous Systems and Pervasive Networks» (EUSPN-2019), 9-й міжнародній конференції «Current and Future Trends of Information and Communication Technologies in Healthcare» (ICTH-2019), 13-й міжнародній конференції «Perspective Technologies and Methods in MEMS Design» (MEMSTECH 2017), 12-й міжнародній науково-технічній конференції «Computer Science and Information Technologies» (CSIT'2017), 13-й міжнародній конференції «The Experience of Designing and Application of CAD Systems in Microelectronics» (CADSM'2015), 5-й Всеукраїнській школі-семінарі «Сучасні комп'ютерні інформаційні технології» (ACIT'2015), 4-й Всеукраїнській школі-семінарі молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології» (ACIT'2014), 13-му міжнародному науковому семінарі «Сучасні проблеми інформатики в управлінні, економіці, освіті» (2014), міжнародній науково-практичній конференції «Інформаційне суспільство: тенденції регіонального розвитку» (ISRDT-2016), XIII науково-практичній конференції «Проблеми та перспективи розвитку економіки і підприємництва та комп'ютерних технологій в Україні» (2017), 10-й міжнародній конференції «Perspective Technologies and Methods in MEMS Design» (MEMSTECH'2014).

Матеріали дисертації неодноразово доповідалися та обговорювалися на наукових семінарах кафедри автоматизованих систем управління Національного університету «Львівська політехніка».

Структура та обсяг дисертації.

Дисертаційна робота складається із вступу, чотирьох розділів, висновків, списку літератури з 101 найменувань та 3 додатків. Загальний обсяг дисертації становить 180 сторінок, з них 159 сторінок основного тексту, який містить 54 рисунки та 8 таблиць.

РОЗДІЛ 1 АНАЛІЗ МОДЕЛЕЙ, МЕТОДІВ І ЗАСОБІВ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ УПРАВЛІННЯ СИСТЕМАМИ «РОЗУМНИЙ БУДИНОК»

Технологія «Розумного будинку» (РБ) бере свій початок у США в 1950-х роках і вже сімдесят років перебуває у стадії активного розвитку [26]. На сьогодні вона викликає все більшу цікавість в професійному середовищі науковців, інженерів та представників бізнесу. Це пояснюється стрімким розвитком інформаційних технологій, а також індивідуалізованим підходом до проектування систем РБ, які орієнтовані на унікальні потреби користувача.

Сучасні вимоги до функціональності таких систем значно перевершують можливості перших реалізацій. Якщо на початкових етапах автоматизація обмежувалася дистанційним керуванням електроприладами чи базовою оптимізацією витрат енергоресурсів (наприклад, увімкненням котла за таймером), то сьогодні очікування користувачів зміщуються у бік створення гнучких, інтелектуальних систем, здатних адаптуватися до звичок і вподобань мешканців [26], [27]. Ідеальна система РБ має функціонувати автономно, навчатися на основі поведінки власників і передбачати їхні потреби, мінімізуючи потребу в ручному втручанні.

Досягнення такого рівня інтелектуальності й автономності вимагає як вдосконалення наявних, так і розроблення нових інформаційних технологій управління. Ключовим аспектом є впровадження механізмів машинного навчання, які дають змогу системі адаптуватися до індивідуальних особливостей користувача шляхом аналізу його поведінки та уподобань. Це своєю чергою ставить завдання розробки ефективних методів навчання й налаштування моделей функціонування РБ, здатних забезпечити комфортну, непомітну та інтегровану присутність системи в повсякденному житті.

1.1 Аналіз моделей та методів розроблення систем «Розумний будинок»

Системи «Розумний будинок» [26], [27] мають багаторічну історію розвитку і на сьогодні є однією з ключових технологій у сфері автоматизації побуту. Незважаючи на стрімке поширення, потенціал таких систем досі не реалізовано в повному обсязі через зростаючі вимоги користувачів, індивідуалізацію запитів та швидкий розвиток інформаційних технологій. Кожне нове покоління РБ демонструє поглиблення інтеграції інтелектуальних алгоритмів керування, аналітики та елементів машинного навчання, що значно ускладнює структуру систем.

В таблиці 1.1 [10] показано еволюцію систем РБ протягом часу їх існування, що охоплює основні етапи розвитку від електромеханічних пристроїв до інтегрованих рішень з використанням IoT, ШІ та предиктивної аналітики [28], [29].

У результаті проведеного аналізу було ідентифіковано сукупність ключових факторів, які сприяють широкому впровадженню, розвитку та адаптації систем РБ у сучасних умовах (рис. 1.1) [10].

Ці фактори визначають функціональні, технічні та експлуатаційні вимоги до інтелектуальних систем управління та формують основу для розроблення адаптивних інформаційних технологій, здатних гнучко реагувати на динамічні зміни зовнішнього середовища та поведінкові особливості користувачів.

Усі вказані фактори є критично важливими для побудови ефективної системи управління, в якій інформаційна технологія не лише виконує роль зв'язувальної ланки між підсистемами, але й виступає засобом персоналізації, предиктивного аналізу та динамічного прийняття рішень [28], [29].

Таблиця 1.1

Етапи розвитку систем «Розумний будинок»

№	Етап розвитку	Характеристика	Ключові особливості	Основні технології	Основні обмеження
1	Ранній етап (1950–1970-ті роки)	Перші спроби автоматизації побутових процесів	Електромеханічні системи, автоматичне вмикання/вимикання освітлення	Таймери, реле, термостати	Відсутність централізованого управління, висока вартість
2	Програмовані системи (1980-ті – середина 1990-х)	Поява мікропроцесорів та перших контролерів	Використання PLC-контролерів, дротові системи управління	Siemens PLC, Honeywell, кабельні системи	Відсутність взаємодії між пристроями різних виробників, складність монтажу
3	Інтегровані системи (кінець 1990-х – 2010-ті роки)	Інтеграція комп'ютерних мереж та інтернету в системи керування	Використання бездротових протоколів, мобільні застосунки, відеоспостереження	Wi-Fi, ZigBee, Z-Wave, KNX	Обмежена сумісність, ризики кібербезпеки
4	Системи зі штучним інтелектом та IoT (2010-ті – сьогодення)	Адаптація до поведінки користувачів за допомогою алгоритмів ШІ	Голосові асистенти, енергоефективність, прогнозування дій користувача	Google Assistant, Amazon Alexa, Apple Siri, IoT-давачі	Залежність від інтернету, загроза витоку персональних даних
5	Майбутні тенденції (2025+)	Повна автономія, інтеграція з «розумними містами»	Використання предиктивної аналітики, автономних систем керування	5G/6G, квантові обчислення, AR/VR, біометричні системи	Висока вартість впровадження, потреба в глобальній інфраструктурі

Аналіз розвитку РБ дає змогу виокремити основні фактори, які сприяли поширенню цих систем, зокрема: зменшення вартості мікроконтролерів, поява стандартів бездротової передачі даних, розвиток хмарних сервісів та зростання попиту на енергоефективність[24], безпеку [21] та захист інформації [30].

Однією з визначальних рис систем РБ є їхня структурна та функціональна варіативність, що зумовлена відсутністю жорстких обмежень щодо архітектури, принципів реалізації та технологій синтезу. Конфігурація таких систем може суттєво відрізнятися залежно від індивідуальних потреб користувачів, обраних

технічних рішень, рівня інтелектуалізації та цільових сценаріїв використання [18].



Рис. 1.1 – Ключові фактори систем РБ

З одного боку, гнучкість проєктування дає змогу створювати персоналізовані рішення, які відповідають специфічним вимогам користувачів. З іншого — вона істотно ускладнює процеси стандартизації, уніфікації та інтеграції компонентів, особливо при побудові масштабованих або міжплатформних систем [30], [31].

Враховуючи вищезазначене, метою даного розділу є систематизація наявних методів синтезу та розроблення систем РБ з подальшим формуванням узагальненої структури адаптивної інформаційної технології управління, що

враховує сучасні IT-рішення, інструменти моделювання, засоби автоматизації та потребу в персоналізованому керуванні на основі поведінкових особливостей користувача [12].

На основі проведеного аналізу визначено ключові етапи, що є критично важливими для проєктування адаптивної системи управління типу РБ. Послідовність цих етапів забезпечує системний підхід до створення високофункціональних, масштабованих і інтелектуально адаптивних рішень [32]:

1. Формулювання функціональних вимог – визначення цілей та завдань, які повинна реалізовувати система відповідно до запитів користувача та сценаріїв експлуатації.

2. Визначення рівня інтелектуалізації – встановлення ступеня автоматизації, адаптивності та можливості самонавчання системи на основі збору, аналізу та обробки даних.

3. Класифікація архітектурних рішень – вибір моделі реалізації системи з урахуванням особливостей топології (провідна, бездротова), структури управління (централізована чи децентралізована), а також типу використовуваних протоколів (відкриті або закриті) [33].

4. Розроблення загальної архітектури системи – визначення основних функціональних модулів, способів взаємодії між компонентами, методів синтезу та технічних засобів реалізації (сенсори, актуатори, мережеві вузли тощо) [33].

5. Проєктування архітектурної та інженерної моделі – використання спеціалізованих CAD-інструментів для створення структурних, електричних та логічних моделей системи.

6. Формування математичної або програмної моделі [33] – моделювання функціональних процесів з метою аналізу взаємодії між компонентами, оцінки динаміки, навантажень і реакцій на зовнішні події.

7. Функціональне тестування та верифікація – перевірка працездатності системи, виявлення критичних ділянок, помилок проектування та оцінка відповідності заданим технічним вимогам.

8. Реалізація та впровадження – інтеграція системи в реальні умови експлуатації з подальшим налаштуванням, запуском та оцінкою її ефективності.

Загальна структура типової системи «Розумний будинок», сформована на основі проведеного аналізу, представлена на рис. 1.2 [10].

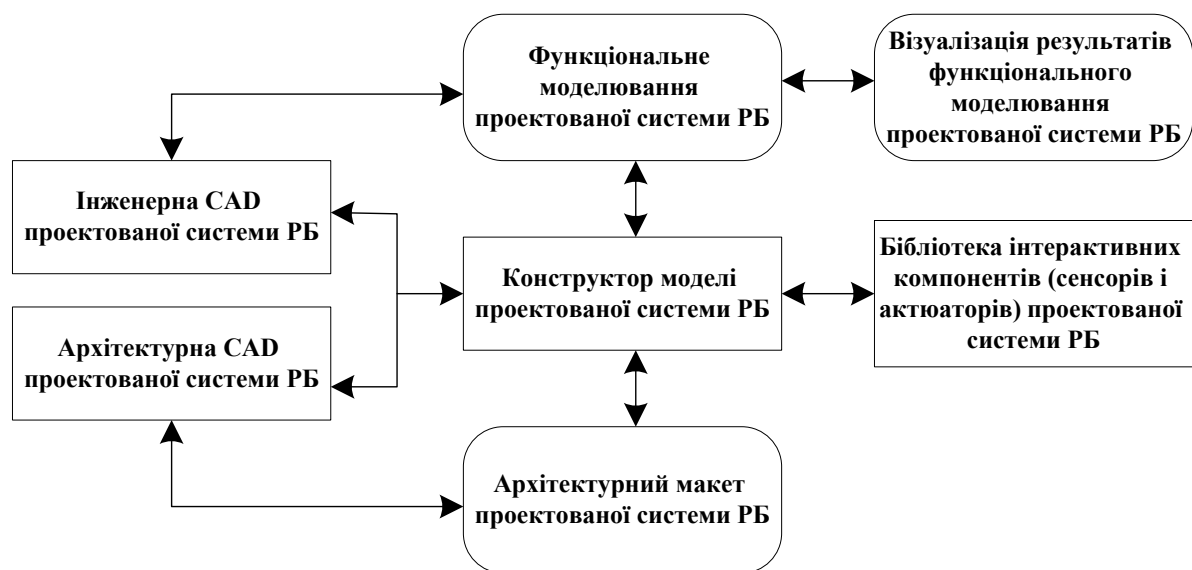


Рис. 1.2 – Загальна структура проекрованої системи РБ

Вибір типу системи за архітектурними ознаками та ключовими характеристиками є одним із початкових і водночас найвідповідальніших етапів розроблення. Саме на цьому етапі закладаються принципові відмінності в реалізації, які зумовлюють архітектуру системи, рівень її гнучкості, адаптивності та здатності до масштабування.

Кожен з типів систем РБ має унікальні функціональні та технологічні особливості, які визначають його переваги, обмеження та доцільність застосування в конкретному середовищі. Узагальнене порівняння таких типів наведено в таблиці 1.2.

Таблиця 1.2

Класифікація систем «Розумний будинок» за архітектурними ознаками

Тип системи	Опис	Ключові переваги	Особливості та обмеження
Провідні	Компоненти підключені до єдиної дротової інформаційної шини	1. Висока надійність і швидкість реакції 2. Стійкість до радіоперешкод 3. Широка сумісність 4. Тривалий термін служби	1. Необхідність прокладки кабелю 2. Висока вартість монтажу 3. Обмежена гнучкість
Бездротові	Взаємодія компонентів через бездротові протоколи	1. Простота встановлення 2. Гнучкість масштабування 3. Відносно невисока вартість впровадження	1. Чутливість до радіоперешкод 2. Потреба в заміні батарей 3. Нижчий рівень безпеки
Централізовані	Управління здійснюється з єдиного головного контролера	1. Уніфікований інтерфейс 2. Можливість реалізації складних сценаріїв 3. Зручність інтеграції	1. Висока вартість 2. Вразливість через єдиний вузол 3. Залежність від постачальника
Децентралізовані	Компоненти мають автономну логіку та пам'ять	1. Надійність 2. Гнучкість налаштування 3. Можливість унікальних сценаріїв	1. Складність синхронізації 2. Навантаження на мережу 3. Складніше налаштування
З відкритим протоколом	Використання відкритих стандартів (KNX, MQTT, BACnet)	1. Широкий вибір пристроїв 2. Ктивна спільнота 3. Незалежність від одного виробника	1. Складність налаштування 2. Можлива несумісність 3. Вища вартість обладнання
З закритим протоколом	Використання власних (пропріетарних) протоколів виробника	1. Оптимізована інтеграція 2. Простота налаштування 3. Підвищена безпека в межах екосистеми	1. Залежність від постачальника 2. Обмежена сумісність 3. Ризик припинення підтримки

Вибір методів синтезу та аналізу для систем РБ є надзвичайно широким та залежить від складності, цільового функціоналу, вимог до адаптивності, самонавчання і взаємодії з користувачем.

Розглянемо основні групи методів і методологій, які застосовуються на різних етапах проектування РБ [33] [34].

1. Методологія SADT [34] (акронім від англ. Structured Analysis and Design Technique) є сукупністю методів, правил і процедур, призначених для побудови функціональної моделі об'єкта будь-якої предметної області. Основна мета методології полягає у формалізованому представленні функціональної структури системи, зокрема її дій та взаємозв'язків між цими діями.

Результатом застосування SADT є модель, що складається з набору діаграм, фрагментів текстів і глосарію, які взаємопов'язані між собою. Діаграми є ключовим компонентом моделі: усі функції системи та її інтерфейси відображаються у вигляді блоків і дуг.

Однією з ключових особливостей SADT є акцент на чіткому визначенні меж моделі. Кожна SADT-модель фокусує увагу лише на одному об'єкті моделювання, що забезпечує точність і сфокусованість аналізу. Такий підхід дає змогу уникнути включення сторонніх елементів у модель і підвищує її цілісність та узгодженість.

Завдяки ієрархічній структурі SADT забезпечує можливість поетапної деталізації системи, починаючи від загального огляду до глибокої декомпозиції функцій. Однак недоліком методології є обмежена здатність до опису динамічних процесів, часових залежностей, стохастичних подій або адаптивної поведінки системи, що робить її малоефективною для розроблення інтелектуальних чи самонавчальних систем.

2. Методологія IDEF0 (Integration DEFinition language 0) є розвитком методу структурного аналізу та проектування SADT [34]. У своїй базовій формі вона поєднує два основних компоненти:

- визначення формальної мови графічного моделювання (синтаксису та семантики);
- опис повної методології побудови моделей.

IDEF0 застосовується для побудови функціональної моделі системи, яка являє собою структуроване відображення функцій, взаємодій або процесів у межах заданої предметної області. Така модель дає змогу отримати формалізоване представлення функціональної архітектури системи, а також її взаємодії з внутрішнім і зовнішнім середовищем.

Моделювання в IDEF0 завжди починається зі створення контекстної діаграми, яка подає систему як єдину функціональну сутність (один блок), пов'язану з зовнішніми елементами за допомогою інтерфейсних дуг. Далі відбувається ієрархічна декомпозиція цієї функції на підфункції, які відображаються на нижчих рівнях діаграм.

Перевагами IDEF0 є формалізованість, модульність, наочність, підтримка багаторівневої декомпозиції та можливість стандартизації моделей. Недоліками є обмежена здатність відображати часову динаміку, асинхронність або стохастичні процеси, а також відсутність вбудованих механізмів адаптивності.

3. Метод структурного системного аналізу і проєктування SSADM (Structured Systems Analysis and Design Method) [35] є комплексною методологією, що застосовується для структурованого аналізу та проєктування інформаційних систем. Вона поєднує три взаємопов'язані підходи до моделювання, кожен з яких дає змогу розглядати систему з різних точок зору:

- логічне моделювання даних (Logical Data Modeling, LDM) — це процес ідентифікації, структуризації та документування даних, необхідних для функціонування системи;
- моделювання потоків даних (Data Flow Modeling, DFM) — процес, спрямований на ідентифікацію, візуалізацію та документування потоків інформації в межах проєктованої системи;

- моделювання поведінки сутностей (Entity Behavior Modeling) — метод, який дозволяє описати, які події впливають на кожну сутність, та у якій послідовності ці події відбуваються. Такий підхід є важливим для формалізації сценаріїв взаємодії з об'єктами системи.

Кожна з цих трьох методик забезпечує окрему, унікальну перспективу на модель системи, що, у сукупності, дає змогу сформувати повне й цілісне уявлення про логіку, структуру та поведінку проєктованої системи [36]. Їхня взаємодія забезпечує узгодженість і точність побудови моделі, що особливо важливо на етапах планування, валідації та реалізації.

Метод SSADM добре підходить для великих і складних проєктів, де потрібна чітка документація, формалізація вимог та контроль якості моделювання. Проте його недоліками є відносна складність впровадження, недостатня підтримка гнучких або адаптивних архітектур, а також обмежена здатність до моделювання динаміки поведінки користувача в реальному часі [36].

4. Аналітичні методи проєктування [37] передбачають побудову математичних моделей, які у явному вигляді відображають взаємозв'язок між характеристиками функціонування системи та її структурно-функціональними й навантажувальними параметрами. Такі методи дають змогу отримати числові залежності, що описують поведінку системи за різних умов експлуатації.

На основі побудованих моделей формуються графічні залежності, які візуалізують вплив окремих параметрів на результати функціонування системи. Це дає змогу виявити критичні властивості та визначити фактори, які мають найбільший вплив на ефективність роботи системи. Отримані аналітичні висновки часто використовуються для оптимізації конфігурації системи, зменшення кількості альтернативних варіантів реалізації та формування рекомендацій щодо раціонального проєктування.

Перевагами методу є:

- висока точність опису процесів у системі;

- можливість отримання формалізованих залежностей;
- зручність у математичному аналізі та оптимізації.

Проте даний метод має і ряд недоліків основними з яких є:

- необхідність формулювання низки припущень у процесі моделювання;
- складність побудови моделей для складних або багатofакторних систем;
- неможливість, у ряді випадків, отримати аналітичне рішення через нерозв'язність рівнянь, відсутність аналітичних форм первісних функцій тощо.

Аналітичні методи є доцільними переважно на етапах попереднього аналізу або в системах зі строго визначеною структурою [37], [38]. Проте для адаптивних, нелінійних або стохастичних систем, зокрема таких як інтелектуальні системи управління в межах РБ, ці методи доцільно доповнювати чисельними, імітаційними чи інтелектуальними підходами.

5. Методи імітаційного моделювання [39], [40], [41] призначені для аналізу функціональних характеристик процесів, що протікають усередині синтезованої системи, з акцентом на оцінку ефективності автоматизованих операцій. Імітаційне моделювання полягає у створенні динамічних моделей, які враховують часові характеристики виконання операцій, взаємодію між підсистемами та зміну параметрів у реальному часі.

У тих випадках, коли аналіз математичної моделі аналітичними методами є надто складним, малоефективним або неможливим через трудомісткість і чутливість до похибок апроксимації та заокруглення, доцільно будувати імітаційну модель. Вона описує процеси як послідовність операцій над числами, що представляють значення входів і виходів відповідних елементів.

Імітаційні моделі дають змогу описувати не лише логіку функціонування системи, а й потоки даних, комунікацію між компонентами, а також кількісні метрики — такі як час відгуку, продуктивність, навантаження на ресурси. Однією з важливих особливостей є можливість асоціювати зміни станів системи з подіями, що дозволяє моделювати подієво-орієнтовану динаміку [41].

Такий підхід дає змогу точно відтворити сценарії функціонування системи в умовах, наближених до реальних, та отримати статистичні дані щодо її поведінки.

Імітаційні методи реалізуються на ЕОМ у вигляді моделей, які, зокрема, можуть поєднувати стохастичні мережі Петрі — інструмент, що дає змогу за властивостями окремих елементів оцінити загальну ефективність системи [42]. Широко використовується також метод статистичних випробувань (метод Монте-Карло), який забезпечує статистичне моделювання, тобто отримання оціночних даних про властивості моделі за допомогою багаторазових експериментів з випадковими параметрами.

Серед недоліків імітаційного моделювання варто відзначити потребу у значних обчислювальних ресурсах, складність побудови, верифікації моделей, а також те, що результати, як правило, мають дискретний характер і не дають змогу аналітично виявити залежності характеристик функціонування системи від її структурно-функціональних параметрів. Це особливо ускладнює процес оптимального синтезу систем РБ, що характеризуються великою розмірністю та високим ступенем взаємозалежності елементів [41], [42].

6. Інтелектуальні методи [43], [44], [45]: нейронні мережі, машинне навчання, нечітка логіка.

Інтелектуальні методи забезпечують здатність систем РБ до навчання на основі історичних або емпіричних даних. Зокрема, нейронні мережі ефективно справляються з розпізнаванням шаблонів поведінки користувача, даючи змогу системі адаптувати свої дії до індивідуальних сценаріїв [44]. Нечітка логіка, у свою чергу, дає змогу приймати рішення в умовах невизначеності — коли вихідні дані є неповними, неточними або частково визначеними, що часто має місце у побутовому середовищі.

У прикладному сенсі нейронні мережі дають змогу знайти розв'язання в тих випадках, коли неможливо або складно побудувати аналітичну функціональну залежність між вхідними параметрами та очікуваними

результатами. У таких ситуаціях процес навчання (або самонавчання) мережі стає ефективною альтернативою традиційному програмуванню, значно скорочуючи час розробки системи [44], [45].

Найбільшу ефективність інтелектуальні методи демонструють при побудові персоналізованих, адаптивних систем керування, які здатні змінювати поведінку на основі накопиченого досвіду. Проте існують і певні обмеження. Насамперед — це висока залежність від якісних та репрезентативних навчальних вибірок. Якщо таких даних недостатньо або вони є нерелевантними, точність прийняття рішень суттєво знижується [45].

Крім того, на практиці трапляються випадки, коли система, заснована на нейронній мережі, приймає некоректні рішення [44]. Для виправлення таких ситуацій важливо передбачити механізми фіксації та повторного навчання мережі на проблемних прикладах. У разі значних змін у поведінці користувачів або умов експлуатації системи може виникнути потреба у повному перенавчанні моделі, що створює додаткові труднощі як для розробників, так і для кінцевих користувачів. Також недоліками є складність інтерпретації рішень, прийнятих інтелектуальними алгоритмами, і вимоги до значних обчислювальних ресурсів у разі використання складних моделей машинного навчання.

7. Методи параметрично-орієнтованого проєктування [46]. Ці методи передбачають декомпозицію проєктованої системи РБ на сукупність окремих компонентів (підсистем, програмних модулів тощо), з подальшим вибором та придбанням типових проєктних рішень, доступних на ринку, для реалізації кожного з компонентів. Наступним етапом є налаштування, доопрацювання або модифікація обраних рішень відповідно до специфіки конкретної системи. Такий підхід дає змогу скоротити час і вартість проєктування, забезпечуючи водночас гнучкість адаптації.

Основними недоліками параметрично-орієнтованого проєктування є:

- обмежена індивідуалізація: типові рішення не завжди повністю відповідають унікальним вимогам користувача, що обмежує рівень персоналізації системи;
- залежність від сторонніх рішень: функціональні можливості обмежені наявними на ринку компонентами, що знижує контроль над архітектурою;
- можливі проблеми сумісності: компоненти різних виробників можуть мати труднощі з інтеграцією, особливо при використанні закритих протоколів;
- складність масштабування: адаптація системи до нових умов чи її розширення іноді вимагає суттєвої перебудови вже наявної структури [44].

8. Методи модельно-орієнтованого проєктування (Model-Based Design) передбачають побудову цифрових моделей систем ще на етапі концептуального проєктування [47], [48] — до фізичної реалізації. Цей підхід полягає в адаптації структури та характеристик типової системи до конкретної моделі об'єкта автоматизації, тобто дає змогу формувати індивідуалізовані рішення на основі типових конфігурацій.

Перевагою є можливість створення віртуальної копії системи для симуляції, перевірки, тестування та виявлення помилок на ранніх етапах. Такий підхід забезпечує точне уявлення про поведінку майбутньої системи в різних умовах експлуатації, дає змогу коригувати технічні параметри, визначати вузькі місця та зменшити ризики проєктних рішень [48].

Реалізація модельно-орієнтованого підходу зазвичай здійснюється за допомогою спеціалізованого програмного забезпечення, такого як MATLAB/Simulink, LabVIEW, Stateflow тощо [48]. Ці інструменти підтримують як моделювання структури типової системи, так і налаштування моделі під індивідуальні вимоги конкретного РБ.

Недоліками такого методу є:

- складність підтримки та масштабування моделей у разі ускладнення системи або додавання нових функцій;

- потреба у високій кваліфікації персоналу, що працює з відповідним програмним забезпеченням;
- значні часові й обчислювальні ресурси на етапі побудови моделей та їх перевірки.

Проте незважаючи на вказані обмеження, модельно-орієнтоване проєктування є ефективним інструментом для побудови адаптивних та надійних систем управління, зокрема у сфері РБ, де велике значення має перевірка сценаріїв взаємодії між компонентами до впровадження [49].

Попри широкий спектр існуючих методів і підходів до моделювання та аналізу систем РБ, жоден із них не забезпечує повного охоплення вимог, що висуваються до побудови адаптивних, персоналізованих та передбачуваних систем управління. Зокрема, сучасні рішення виявляються недостатніми для реалізації таких функцій, як самоналаштування під змінну поведінку користувача, динамічне прогнозування подій та своєчасна реакція системи на них [48], [49]. Це обґрунтовує необхідність розробки нової інформаційної технології, яка б поєднувала формалізованість, стохастичність та адаптивність.

1.2 Аналіз програмного забезпечення для роботи з системами «Розумний будинок»

На сучасному етапі розвитку ІТ-технологій існує значна кількість різноманітного програмного забезпечення (ПЗ), що використовується у системах РБ. Це як ПЗ для синтезу та проєктування таких систем, так і програмні засоби для подальшого управління, моніторингу, адміністрування, налагодження та аналізу вже реалізованих рішень.

Часто програмне забезпечення постачається разом із мікроконтролерами або іншим апаратним обладнанням, яке застосовується в системах РБ. У зв'язку з цим поділ на апаратну та програмну складові є умовним і не завжди доцільним. У межах цього та наступного підрозділів таке розмежування

подекуди ігноруватиметься — у випадках, коли доцільно розглядати програмно-апаратний комплекс як цілісну інтегровану систему.

Основною метою цього підрозділу є узагальнення спектра програмного забезпечення, представленого на сучасному ІТ-ринку, що використовується в системах РБ, з акцентом на виявлення ключових функціональних характеристик та обмежень, які необхідно враховувати при розробленні ефективного, адаптивного й безпечного ПЗ для систем РБ.

Умовно ПЗ для систем РБ поділяється на три основні категорії [50]:

1. Настільне програмне забезпечення перевагами якого є:

- доступність обчислювального вузла для реалізації інтелектуального управління;
- порівняно низька вартість модулів введення/виведення;
- відсутність жорстких обмежень на кількість підключених пристроїв;
- можливість створення комплексних систем із широким функціоналом;
- зручне адміністрування охоронних комплексів, зокрема з доступом до відеокамер і сенсорів [22].

В той же час воно має ряд недоліків, а саме:

- залежність від апаратного вузла з надійною ОС;
- уразливість до механічних пошкоджень та збоїв живлення;
- потреба в самостійному плануванні та реалізації системних схем;
- необхідність встановлення додаткових драйверів;
- залежність продуктивності від апаратних характеристик ПК;
- втрата керованості при виході ПК з ладу.

2. Web-орієнтоване програмне забезпечення з такими перевагами [50]:

- віддалений доступ з будь-якої точки світу;
- відсутність потреби в інсталяції на клієнтський пристрій;
- кросплатформеність;
- підтримка доступу з ПК та мобільних пристроїв;
- кращий рівень захисту від локального шкідливого ПЗ;

- висока надійність завдяки серверному резервуванню.

Проте і воно має недоліки:

- залежність від постійного доступу до інтернету;
- ризики перехоплення даних, складність виявлення джерел несанкціонованого доступу;
- психологічний дискомфорт через зберігання персональних даних на віддалених серверах;
- обмеження у впровадженні індивідуальних рішень без відповідної серверної інфраструктури;
- залежність швидкодії від навантаження на сервер.

3. Мобільне програмне забезпечення з такими перевагами як [50]:

- висока мобільність та оперативність реагування;
- простота користування (дружній інтерфейс);
- відсутність потреби в драйверах чи додатковому ПЗ;
- вартісна ефективність при масштабуванні на групу користувачів.

І все ж є недоліки:

- вразливість до фізичної втрати пристрою;
- обмежена сумісність з деякими ОС чи версіями;
- менші обчислювальні ресурси порівняно з ПК;
- обмежений розмір дисплея;
- нижчий рівень антивірусного захисту.

Нижче наведено приклади сучасного програмного забезпечення, що використовується для управління системами РБ, з коротким аналізом їхніх основних характеристик.

MajorDoMo [51] — це гібридне ПЗ (настільне з web доступом) безкоштовне програмне забезпечення з відкритою архітектурою, яке забезпечує централізоване управління усіма підсистемами РБ, зокрема електроприладами, сигналізацією, освітленням та температурою. Завдяки невибагливості до ресурсів та сумісності з Windows і Linux, MajorDoMo можна використовувати

навіть на ПК середнього класу. Утім, інтерфейс програми потребує технічної підготовки користувача, а налаштування системи часто вимагає глибшого втручання у структуру ПЗ, що ускладнює його використання для недосвідчених осіб.

Home Sapiens [52] – web застосунок з елементами мобільного доступу, що відзначається зручністю голосового управління та підтримкою універсального пульта для часто використовуваних функцій. Завдяки віддаленому доступу з комп'ютера або смартфона, програму можна використовувати незалежно від місцезнаходження. Проте функціонал Home Sapiens у базовій версії є доволі обмеженим, а повноцінне використання усіх можливостей передбачає додаткові витрати.

Z-Wave Fibaro [53] — це комплексне рішення, що побудоване навколо власного протоколу бездротового зв'язку, яке поєднує апаратні й програмні компоненти. Платформа орієнтована на користувачів без навичок програмування, пропонуючи простий спосіб розширення системи за допомогою модульних елементів. Завдяки шифруванню з обміном ключами між компонентами, забезпечено високий рівень безпеки. Водночас, обмеження на сумісність із стороннім обладнанням і необхідність придбання усіх компонентів від одного виробника можуть обмежувати гнучкість проєктування.

Xiaomi Smart Home [54] реалізує концепцію інтегрованого програмно-апаратного комплексу, де керування усіма пристроями відбувається через мобільний застосунок. Це зручно з погляду мобільності та швидкої установки. Проте залежність від закритої екосистеми та обмежена кросплатформеність можуть спричинити труднощі у налаштуванні складніших сценаріїв або у використанні сторонніх компонентів.

Crestron – Android R2 [55] є мобільним застосунком, що дає змогу управляти всіма аспектами системи РБ через смартфон. Перевагою є автоматичне виявлення нових пристроїв і підтримка індивідуальних сценаріїв з можливістю запису таймерів. Однак програма вимагає стабільного підключення

до Інтернету та не передбачає повноцінного локального режиму, що може бути критичним у разі збоїв мережі.

EasyHome [56] вирізняється гнучкістю — програмне забезпечення сумісне з Windows, Android та iOS, і працює з широким переліком промислових контролерів через протокол ModBus TCP. Його функціонал охоплює управління освітленням, кліматом, охоронними системами, лічильниками ресурсів і навіть аварійними ситуаціями. Завдяки можливості інтеграції із зовнішніми IP-камерами та налаштування сценаріїв, EasyHome підходить для комплексної автоматизації. Однак повноцінне розгортання потребує ретельної конфігурації та, у деяких випадках, технічної підтримки.

Таким чином, хоча представлений спектр ПЗ охоплює різні функціональні потреби користувача, більшість рішень залишаються обмеженими з точки зору відкритості архітектури, гнучкості масштабування та підтримки адаптивних функцій, що є критичними для реалізації дійсно інтелектуальних, самоналаштовуваних систем. Саме ці аспекти потребують подальших досліджень і розроблення нових інформаційних технологій, зокрема на основі стохастичних моделей і мереж Петрі-Маркова. В таблиці 1.3 представлено аналіз підтримки адаптивності в ПЗ для систем РБ.

У результаті аналізу сучасного програмного забезпечення для систем РБ було виявлено широкий спектр інструментів, що охоплюють різні функціональні аспекти — від базового управління до комплексної автоматизації. Однак більшість таких рішень демонструють лише умовну адаптивність, яка реалізується переважно через сценарне управління, систему тригерів або логіку «умова—дія».

Серед досліджених рішень найвищу (умовну) адаптивність демонструють MajorDoMo, Z-Wave Fibaro, EasyHome та Xiaomi Smart Home — завдяки підтримці автоматизованих сценаріїв та певних умовних реакцій. Натомість програмне забезпечення на кшталт Home Sapiens і Crestron R2 мають лише

обмежену можливість до ручного налаштування або створення простих сценаріїв [57].

Таблиця 1.3

Підтримка адаптивності систем РБ в ПЗ

Назва ПЗ	Підтримка адаптивності	Коментар
MajorDoMo	Так, частково (сценарне управління, правила)	Можна створювати логіку на основі подій, значень сенсорів, часу тощо. Підтримується правилове управління, але відсутнє самонавчання.
Home Sapiens	Обмежена (ручне налаштування сценаріїв)	Голосове управління та базові сценарії, без контекстної адаптації чи навчання.
Z-Wave Fibaro	Так, обмежено (автоматизація, сценарії, тригери)	Є запрограмовані сценарії, тригери. Відсутнє машинне навчання.
Xiaomi Smart Home	Так, частково (правила, сцени, умовна логіка)	Правила типу 'якщо..., тоді...', що імітують адаптивність. Немає самонавчання.
Crestron – Android R2	Обмежено (сценарії, таймери)	Прості сценарії за розкладом або подією, адаптивність не є ключовою функцією.
EasyHome	Так (розгалужені сценарії, умови, аварійні реакції)	Реакція на події, зміни параметрів, умовне управління без самонавчання.

Жодна з розглянутих систем не реалізує повноцінну адаптацію на основі самонавчання чи аналізу поведінки користувача. Це свідчить про відсутність у сучасних ПЗ гнучких механізмів, які б давали змогу системі розвиватися,

підлаштовуватись під зміну користувацьких звичок чи адаптувати поведінку у динамічному середовищі.

Таким чином, представлене ПЗ має суттєві обмеження щодо відкритості, гнучкості масштабування та адаптивності, що критично важливо для побудови інтелектуальних керувальних систем. Це підтверджує актуальність розроблення нових програмних підходів — зокрема, на основі формальних стохастичних моделей, таких як мережі Петрі-Маркова, які здатні враховувати подієвість, ймовірнісну природу подій і історію поведінки користувача для реалізації адаптивного, прогнозованого управління у системах РБ.

1.3 Аналіз апаратного забезпечення систем «Розумний будинок»

На сучасному ринку представлено велику кількість мікроконтролерів і програмованих логічних пристроїв, які можуть використовуватись у системах РБ. Навіть якщо розглядати виключно ті мікроконтролери, які використовуються для побудови систем автоматизації побутового середовища, їх перелік налічує сотні, а подекуди й тисячі моделей. Така різноманітність зумовлена кількома факторами: широкою доступністю елементної бази, унікальністю функціональних вимог до кожної конкретної системи РБ, а також індивідуальними підходами до проектування відповідно до потреб замовника.

Кожна система РБ має свій набір задач, функцій, алгоритмів і сценаріїв поведінки, що, відповідно, вимагає гнучкого вибору апаратної бази. Сучасні мікроконтролери відрізняються між собою за розміром, потужністю, набором інтерфейсів, програмованістю, енергоспоживанням, масштабованістю та підтримкою комунікаційних протоколів.

Нижче подано короткий огляд найбільш популярних апаратних рішень для систем РБ з характеристиками, що визначають їх придатність до побудови адаптивних, масштабованих і прогнозованих систем.

Arduino — універсальна платформа з відкритим кодом, яка підтримує широкий набір цифрових та аналогових інтерфейсів. Arduino легко інтегрується

з давачами та виконавчими механізмами, проте має обмежені ресурси обробки даних, що знижує її ефективність у складних адаптивних сценаріях [67].

Raspberry Pi — міні-комп'ютер із повноцінною ОС (наприклад, Linux), що має значно більші обчислювальні можливості, ніж Arduino. Це робить його придатним для реалізації складніших моделей поведінки, включаючи модулі самонавчання [58], [59].

Siemens LOGO — програмований логічний контролер для базових систем автоматизації. Має готову бібліотеку функцій, вбудований Ethernet і сумісність з доповнюючими модулями, однак обмежений у масштабованості та інтеграції складних моделей адаптації [60].

Beckhoff — потужні програмовані контролери, що охоплюють лінійку від простих вузлів до повноцінних ПК. Програмуються в середовищі TwinCAT і є одними з найкращих для промислових систем. Підтримують інтеграцію з системами реального часу, що важливо для реалізації мереж Петрі-Маркова [61].

WAGO — підтримує кілька мов програмування (LD, FBD, SFC) і має хорошу гнучкість. Відзначається стабільністю та можливістю роботи в широкому температурному діапазоні, але обмежений у можливостях навчання [62].

Velocio — компактні програмовані логічні комплекси (ПЛК), що підтримують графічне програмування. Підходять для компактних систем, але не розраховані на складні обчислення або масштабовану поведінку [63].

Vera — контролери з підтримкою Z-Wave Plus, орієнтовані на побутові задачі автоматизації. Оснащені інтуїтивним UI та базовою логікою сценаріїв, проте мають закриту екосистему [64].

PWU12 «RABBIT» — мікроконтролер із вбудованими Web/FTP-серверами та підтримкою широкого спектра протоколів. Завдяки цьому має гарні можливості для мережевої інтеграції, але обмежений апаратно [65].

ОВЕН — український виробник з широким модельним рядом ПЛК. Підтримує протоколи Modbus, має можливість роботи з нестандартними пристроями, а також підтримку реального часу. Програмується в CODESYS, що забезпечує гнучкість, у тому числі для інтеграції з адаптивними системами [66].

Таким чином, попри широке різноманіття доступних мікроконтролерів, більшість із них орієнтовані на реалізацію певного обмеженого класу задач і не мають вбудованих механізмів підтримки адаптивної поведінки. Проте, на відміну від програмного забезпечення, апаратна складова дає змогу здійснити часткову або повну реконфігурацію через перепрограмування або заміну прошивки. Це відкриває потенціал для впровадження інтелектуальних моделей управління, зокрема на основі мереж Петрі-Маркова, які потребують як подієвої логіки, так і можливості обробки статистичних і часових залежностей.

1.4 Аналіз підходів до визначення активностей в системах

«Розумний будинок»

Одним з ключових аспектів підвищення ефективності функціонування систем РБ є здатність до розпізнавання та адаптації до активностей користувача. Визначення людської активності відіграє важливу роль у багатьох галузях, зокрема в медицині — для моніторингу стану пацієнтів під час реабілітації, підтримки осіб похилого віку [66], [68], [69], а також для відстеження динаміки захворювань [70]. Подібні підходи застосовуються і в спортивній аналітиці [71].

В контексті РБ визначення активностей дозволяє реалізовувати індивідуалізовані сценарії управління, забезпечувати комфорт, безпеку й енергоефективність. Одним із поширених підходів до цього завдання є використання моделей на основі штучних нейронних мереж (ШНМ) [72]. У ряді робіт запропоновано нейроконтролери, що функціонують як підсистеми розумного будинку, здатні обробляти нечіткі та неструктуровані вхідні дані [73] та демонструвати високу швидкодію в разі апаратної реалізації [74], [75].

Однак незважаючи на переваги, такі моделі мають низку обмежень. Зокрема, складність реалізації перенавчання нейронної мережі в реальному часі унеможливорює повноцінну адаптацію до змін поведінки користувача. Крім того, побудова таких моделей вимагає значних обсягів статистичних даних для навчання [76], [77], [78], та потужної обчислювальної інфраструктури, що обмежує їхнє застосування в побутових умовах. До того ж, масштабованість і підтримка складних сценаріїв адаптації в цих моделях залишаються проблематичними [79], [80], [81].

Альтернативні підходи передбачають використання систем давачів, що дозволяють безпосередньо визначати присутність, рух, тип дії користувача, або його розташування у просторі [82], [83]. Такі рішення є перспективними в контексті побудови контекстно-залежних адаптивних алгоритмів, зокрема на основі подієво-орієнтованих моделей [84], [85], [86].

З урахуванням вказаних обмежень, виникає потреба у створенні нових підходів до моделювання активностей, які поєднували б здатність до адаптації, подієвість, ймовірнісне оцінювання та підтримку роботи в реальному часі. У цьому контексті формальні моделі на основі мереж Петрі-Маркова відкривають нові можливості — зокрема, для інтеграції подій, станів і стохастичної природи поведінки користувача в межах єдиної адаптивної інформаційної технології.

1.5 Формулювання задач дисертаційного дослідження

На основі проведеного аналізу моделей, методів, підходів та засобів інформаційних технологій системи РБ сформульовано наступні задачі дисертаційного дослідження:

1. Здійснити аналіз методів, моделей і засобів інформаційних технологій управління системами РБ.

2. Розробити метод синтезу моделей систем РБ, які б дали змогу реалізовувати кількісну оцінку ймовірнісних процесів, що виникають в результаті взаємодії проєктованої системи з користувачем.

3. Розробити моделі для адаптивної системи РБ, які ґрунтуються на мережах Петрі-Маркова.

4. Розробити метод навчання моделей систем РБ, який би забезпечив механізм адаптації проєктованої системи РБ до особливостей характеру та поведінки конкретних користувачів системи.

5. Розробити автоматизований програмно-апаратний комплекс дослідження, адміністрування та моніторингу проєктованих систем РБ, а також реалізації наведених вище моделей та методів кількісної оцінки імовірнісних процесів, що виникають в результаті взаємодії системи з користувачами, та механізму адаптації проєктованої системи РБ до особливостей поведінки користувачів системи.

Висновки до розділу 1

У першому розділі дисертаційної роботи проведено комплексний аналіз сучасного стану методів, моделей, програмного й апаратного забезпечення, що використовуються у проєктуванні та реалізації систем РБ. Результати цього аналізу дозволили сформулювати ключові вимоги до майбутньої інформаційної технології управління адаптивними системами РБ, а саме:

У результаті аналізу моделей, методів та підходів до проєктування систем РБ було виявлено, що наявні методології (SADT, IDEF0, SSADM), аналітичні та імітаційні підходи, інтелектуальні та модельно-орієнтовані методи, хоча й забезпечують окремі аспекти моделювання та синтезу, не дозволяють комплексно реалізувати повноцінну адаптивність, самонавчання та поведінкове прогнозування в системах РБ. Жоден із розглянутих підходів не забезпечує необхідної формалізованої підтримки адаптації систем до змінних моделей поведінки користувача в реальному часі. Це підкреслює потребу у створенні нового підходу, зокрема на основі стохастичних моделей і мереж Петрі-Маркова.

Аналіз сучасного програмного забезпечення для систем РБ дав змогу виокремити три основні категорії ПЗ: настільне, веб-орієнтоване та мобільне.

Для кожного з них було сформульовано типові переваги та обмеження. Проведено поглиблений огляд найбільш поширених програмних продуктів, що використовуються у практиці управління системами РБ. Особливу увагу приділено їх функціональним можливостям, відкритості, підтримці адаптивності та залежності від апаратної платформи. Як показав аналіз, жодна з існуючих програмних платформ не реалізує повноцінну поведінкову адаптацію або механізми прогнозування, що підтверджує доцільність розробки нового ПЗ з відповідними інтелектуальними можливостями.

Проведений огляд апаратного забезпечення — зокрема, програмованих мікроконтролерів і логічних контролерів — продемонстрував широке різноманіття рішень, які використовуються в сучасних системах РБ. У межах аналізу представлено ключові технічні характеристики 9 типових мікроконтролерних платформ (Arduino, Raspberry Pi, Siemens LOGO, Beckhoff, WAGO, Velocio, Vera, PWU12 Rabbit, ОВЕН), які відрізняються за рівнем масштабованості, типом програмування, комунікаційними можливостями та цільовим призначенням. На відміну від програмного забезпечення, ці пристрої в більшості випадків дають змогу реалізувати адаптивну поведінку через відповідне програмування, однак вимагають ручного налаштування та не підтримують самонавчання.

Загалом результати розділу підтверджують, що незважаючи на значний прогрес у розвитку складових систем РБ, наявні методи, програмні та апаратні засоби не задовольняють у повному обсязі сучасні вимоги до адаптивності, прогнозування та гнучкого налаштування під індивідуальні моделі поведінки користувача. Це обґрунтовує необхідність розроблення нової інформаційної технології управління адаптивними системами РБ, заснованої на використанні формалізованих стохастичних моделей, зокрема мереж Петрі-Маркова.

РОЗДІЛ 2. МОДЕЛІ АДАПТИВНОЇ СИСТЕМИ «РОЗУМНИЙ БУДИНОК» НА ОСНОВІ МЕРЕЖ ПЕТРІ-МАРКОВА ТА ДОПОВНЕНИХ ФУНКЦІОНАЛЬНИМИ КОМПОНЕНТАМИ

Основною вимогою до сучасних систем «Розумний будинок» є підвищена оперативність реагування та здатність до самонавчання. Інакше кажучи, система має «працювати на випередження», враховуючи індивідуальні особливості та побажання користувачів, яких вона обслуговує. Важливою особливістю таких систем є їхня персоніфікованість: кожен РБ повинен бути максимально адаптованим до конкретного користувача або групи користувачів, що робить кожен систему унікальною за структурою, поведінкою та функціональністю.

Серед найбільш ефективних інструментів для синтезу складних адаптивних систем доцільно виокремити мережі Петрі [87], [88], [89] які забезпечують можливість аналізу динаміки та надійності на системному рівні проектування. Водночас моделі Маркова [90], [91] дають змогу враховувати ймовірнісну природу змін станів системи, що є надзвичайно важливим для прогнозування її поведінки у широкому спектрі сценаріїв.

Однак навіть поєднання мереж Петрі та моделей Маркова не забезпечує повного задоволення вимог до сучасних інтелектуальних систем РБ [92], [93], [94]. Це зумовлює необхідність розроблення нових методів, моделей і засобів для синтезу, моделювання та функціонування таких адаптивних систем, які б поєднували структурно-динамічні та ймовірнісні аспекти в єдиній моделі.

2.1. Модель взаємодії власника та адаптивної системи «Розумний будинок»

Приклад схеми взаємодії між основними складовими системи РБ подано на рис. 2.1. Основними учасниками цієї взаємодії є власник, система РБ та зовнішнє середовище [2].

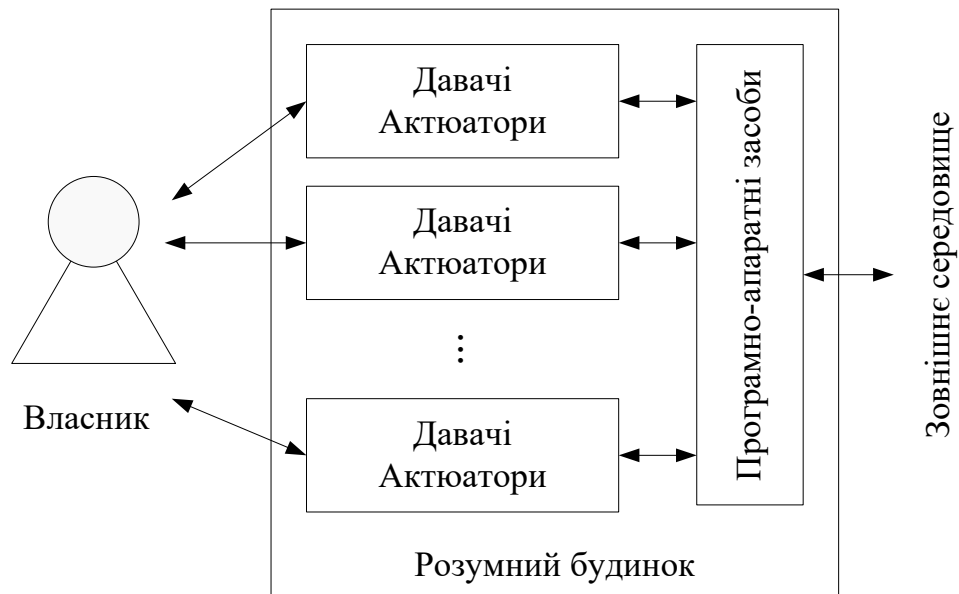


Рис.2.1 - Схема взаємодії між системою РБ, користувачем та зовнішнім середовищем

У загальному випадку структуру, подану на рис. 2.1, можна формалізувати за допомогою кортежу:

$$\text{Str} = \langle \text{Str}_{\text{owner}}, \text{Str}_{\text{SH}}, \text{Str}_{\text{env}} \rangle, \quad (2.1)$$

де:

- $\text{Str}_{\text{owner}}$ — множина осіб, які є власниками системи РБ;
- Str_{SH} — множина, що включає приміщення та програмно-апаратні засоби системи РБ;
- Str_{env} — множина складових зовнішнього середовища, таких як комунальні служби, системи енергопостачання тощо.

Для опису функціонування системи РБ доцільно використовувати теорію мереж Петрі, яка дає змогу формалізувати процеси переходів і визначати стани, в яких перебувають елементи системи на кожному етапі її роботи [87], [95].

У найпростішому випадку застосовуються класичні (прості) мережі Петрі, які дають змогу змодельовати базову логіку взаємодії між компонентами. Для моделювання взаємодії групи користувачів (власників) доцільно використовувати кольорові мережі Петрі, які забезпечують ідентифікацію окремих дій або користувачів у межах одного місця мережі. Для опису поведінки системи в умовах випадкових змін та часових затримок

застосовують стохастичні та функціональні мережі Петрі, які розширюють можливості класичної моделі для потреб аналізу реальних сценаріїв експлуатації системи РБ [2], [88], [89].

2.2. Моделі системи РБ для підвищення оперативності реагування системи «Розумний будинок», побудованої на основі моделі Маркова

Одним зі шляхів реалізації функціонування системи РБ на випередження є застосування стохастичних моделей, що враховують випадковий характер процесів. Такий підхід дає змогу здійснити деталізований аналіз і обчислення ймовірнісних характеристик процесів, а надалі – розробити відповідну інтелектуальну підсистему прийняття рішень [88].

Одним з основних представників класу стохастичних моделей є моделі (ланцюги та мережі) Маркова (ММ), які оптимально підходять для моделювання процесів і станів, що виникають у процесі комунікації між користувачем та системою РБ. Основні переваги застосування моделей Маркова полягають у наступному [1], [90]:

- ММ можна легко синтезувати на основі експериментальних даних;
- алгоритми опрацювання ММ не потребують значних затрат машинного часу;
- результати моделювання (як і самі ММ) можуть бути представлені у зручній, інформативній та зрозумілій графічній формі.

Для демонстрації доцільності застосування моделей Маркова розглянемо конкретний приклад [2]. Припустимо, що користувач зранку прокидається, готується до роботи та виходить з дому. На рис. 2.2 представлено схемну модель для відслідковування переміщення користувача в межах будівлі з урахуванням відправної точки («спальня») та фінальної точки («вулиця»).

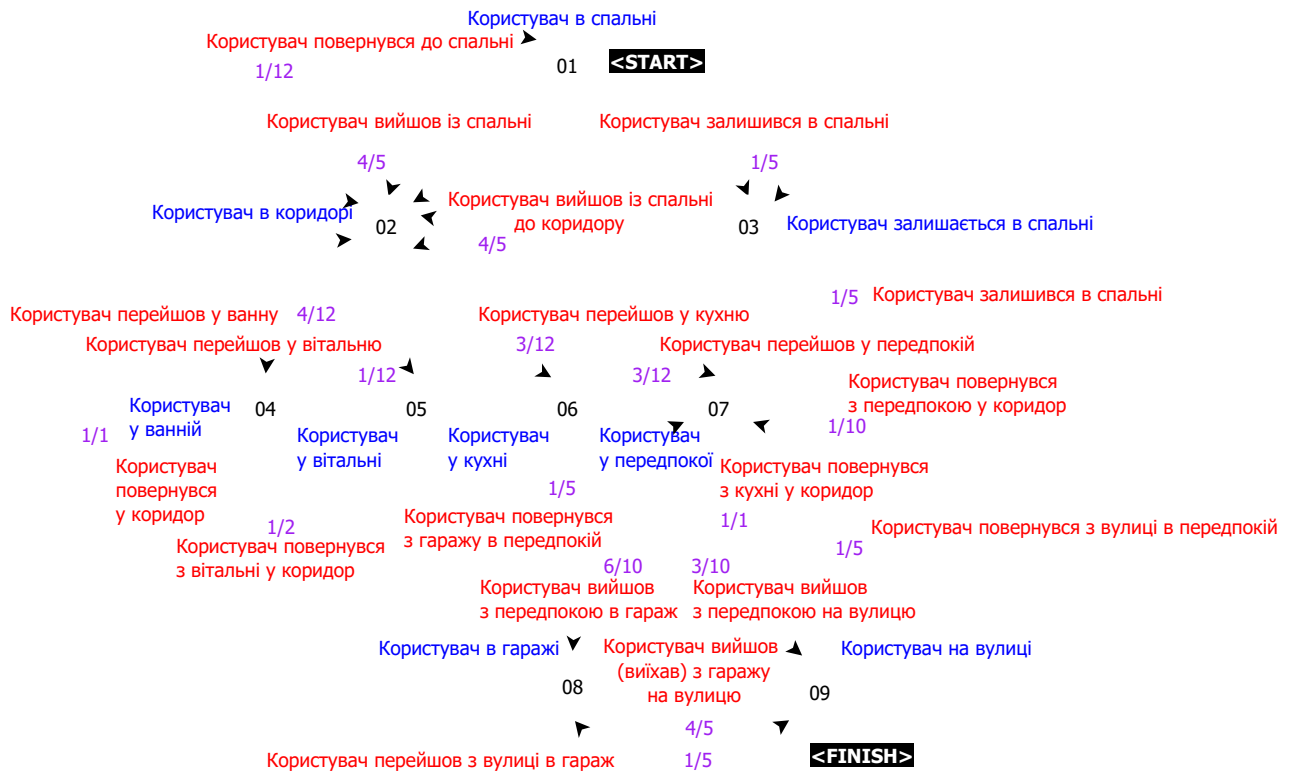


Рис. 2.2. Схемна модель відстеження переміщення користувача в будинку: стани (позначені номерами від 01 до 09) та ймовірнісні переходи між ними

Структуру базової моделі можна математично описати за допомогою теорії кортежів наступним чином [88]:

$$\text{Str} = \langle S, F, J \rangle, \quad (2.2)$$

де:

- S – множина станів;
- F – множина дуг (переходів між станами);
- J – множина ймовірностей досягнення станів (значення, прив'язані до дуг).

Слід відзначити, що для інших сценаріїв (наприклад, коли користувач ввечері повертається з роботи або перебуває у певній кімнаті в різні періоди доби) чи різні дні тижня, початкові та кінцеві точки моделювання можуть змінюватися, що вимагатиме побудови відповідних моделей Маркова з відкоректованими параметрами [2].

Також варто зазначити, що стан «03» у даній моделі, по суті, є надлишковим. З метою оптимізації розроблюваних моделей подібні стани доцільно виключати, хоча в класичному представленні моделей Маркова вони часто присутні. Представлена базова модель є загальною та відображає лише основні аспекти одного із сценаріїв функціонування системи РБ.

Для подальшого формалізування моделі (2.2) та її схематичного представлення (див. рис. 2.2), доцільно застосувати теорію стохастичних мереж Петрі [88].

Отже, стан в будь-який момент можна визначити:

$$M = f(S, T, F, J), \quad (2.3)$$

де: T – множина переходів.

Для поданої на рис. 2.2 моделі маємо 9 станів, тобто: $S = \{s_1, s_2, \dots, s_9\}$. Кількість дуг подвоюється, оскільки кожній дузі на рис.2.2 буде відповідати перехід, що має свою ймовірність спрацювання, а кожному такому переходу відповідатиме одна вхідна та одна вихідна дуга.

Відповідно, модель на основі стохастичної мережі Петрі (2.3) буде вдосконалена та отримає такий вигляд [2], [88]:

$$M = f(S, T, Fin, Fout, J), \quad (2.4)$$

де:

- Fin – множина вхідних дуг (щодо переходів);
- $Fout$ – множина вихідних дуг;
- функція f – визначає умови спрацювання переходів [89].

На рис.2.3 кількість переходів між станами в одному напрямку рівна 1. Аналогічна кількість вхідних та вихідних дуг, а початкове маркування $M_0(1, 0, 0, 0, 0, 0, 0, 0, 0)$.

Для більш детального опису сценарію функціонування РБ доцільно модифікувати модель на рис. 2.3 і ввести такі додаткові коригувальні компоненти [2]:

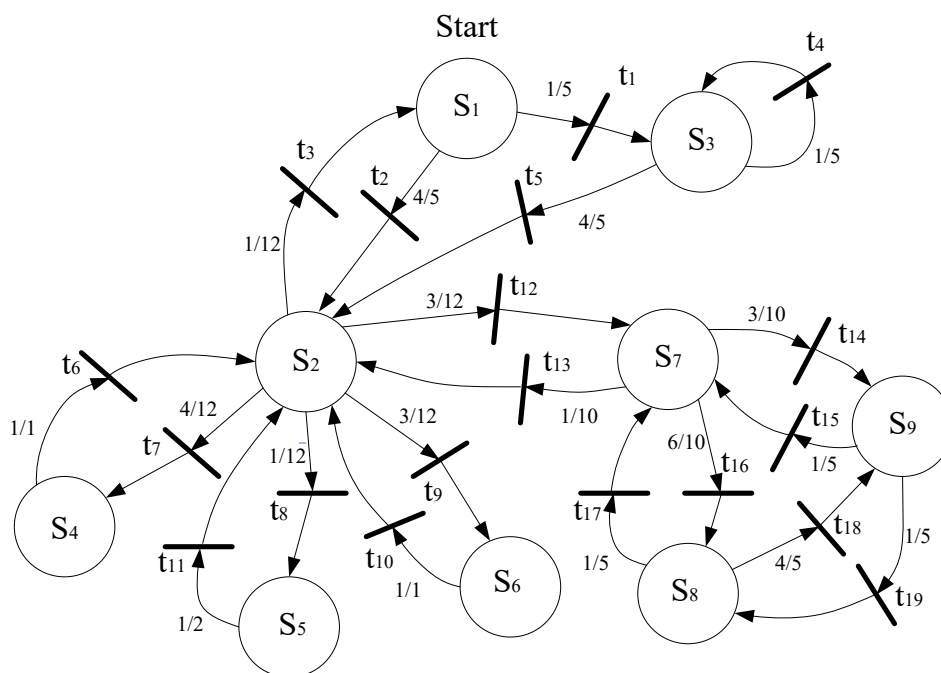


Рис.2.3 - Схемна форма моделі на основі стохастичних мереж Петрі

- необхідно врахувати дані від усіх наявних в розроблюваній системі РБ сенсорів збору та опрацювання вхідних даних (сенсори руху, звуку, вологості, освітлення, температури, а також дані про поточний час та дату). Важливо також забезпечити кожного із вхідних сенсорів своїм персональним «слухачем», який періодично опитує сенсор або реагує на зміну його стану. Дані збираються у відповідному блоці пам'яті, звідки їх може зчитувати контролер блоку управління;

- необхідно забезпечити можливість корекції коефіцієнтів ймовірностей матриці переходів системи, зважаючи на поведінку користувача. Наприклад, якщо користувач зранку виходить зі спальні та відвідує ванну кімнату, ймовірність наступного переходу в кухню буде вища, ніж перехід у передпокій, але після повторного потрапляння в коридор співвідношення цих ймовірностей зміниться. При цьому значення коефіцієнтів встановлюють також у відповідності до значень, отриманих від вхідних сенсорів системи. Якщо користувач прокидається вночі, ймовірність відвідування кухні зростає (наприклад, щоб випити склянку води), тоді як інші переходи стають менш вірогідними. Підґрунтям цьому виступають також статистичні дані, отримані в

період навчання системи. Тут доцільно додатково ввести також лічильник ітерацій переходу системи в той чи інший стан і досягти, тим самим, кращої динаміки процесу корекції коефіцієнтів ймовірностей матриці переходів системи;

- до моделі необхідно включити компоненти - «виконавці», які активують актуатори вихідних процесів (наприклад, ввімкнення/вимкнення світла, витяжки, нагрівання двигуна автомобіля, роботу побутових приладів). А також задати «порогові значення» ймовірностей, при досягненні яких ці «виконавці» запускатимуть відповідні актуатори (дії).

«Слухачі» (Listeners в мові програмування JAVA) – це компоненти, які або періодично опитують стан сенсора, або реагують на кожну зміну стану сенсора, за яким вони закріплені. Дані від усіх сенсорів передаються у відповідний блок пам'яті, звідки можуть бути опрацьовані відповідними контролерами блоку управління.

Корекція коефіцієнтів ймовірностей матриці переходів системи здійснюється при кожному переході між станами системи.

«Виконавці» запускають в дію відповідні актуатори при досягненні порогових значень ймовірностей переходу або при отриманні певних вхідних значень від сенсорів. Ситуація, коли «виконавці» запускають в дію відповідні актуатори при досягненні «порогових значень» коефіцієнтів ймовірності дає змогу реалізувати підвищення оперативності реагування системи. У цьому випадку система кожного разу після здійснення корекції коефіцієнтів ймовірності матриці переходів розраховує імовірність досягнення усіх станів s_i-s_n (окрім s_p) з поточного стану s_p , і якщо імовірність досягнення деякого конкретного стану s_j із поточного стану s_p більша за порогове значення для стану s_j , то система діє на випередження і прискорює досягнення стану s_j (відбувається так зване «прогнозоване» досягнення стану системи).

Наприклад, у робочий день взимку, коли користувач прокидається, виходить зі спальні в коридор, а потім переходить до ванної кімнати (стан

системи «04»), коефіцієнти ймовірностей можуть змінитися таким чином, що ймовірність переходу з стану «04» до стану «08» досягла порогового значення. У результаті запускається механізм прискорення, який дає змогу системі ще до фактичного досягнення стану «08» ініціювати відповідні операції, наприклад, дистанційно увімкнути витяжку в гаражі та запустити двигун автомобіля для попереднього прогрівання. Такий підхід дає змогу економити час користувача, забезпечуючи більш оперативне реагування системи.

Таким чином, ввівши додатково в побудовану первинну модель Маркова [90] усі перелічені вище компоненти, ми отримаємо модифіковану ММ абсолютно нового якісного та функціонального рівня, здатну з високою точністю відображати поведінку користувача в системі РБ та значно підвищувати оперативність її реагування. Замість поліпшення лише окремих елементів системи, така модель забезпечує комплексний підхід до управління РБ.

Отже, в результаті ми отримаємо вдосконалену ММ (рис. 2.4), доповнену додатковими компонентами (наявні сенсори, «виконавці», а також можливість корекції коефіцієнтів ймовірностей матриці переходів системи), для моделювання даного сценарію як одного із можливих погоджених сценаріїв [2].

У математичній формі, схемну модель на рис. 2.4 можна записати наступним чином [90]:

$$IMS = \langle M_0, J_{\max} \rangle, \quad (2.5)$$

де $J_{\max}$ – множина граничних значень ймовірностей, які призводять до спрощення функціонування пристрою.

Для того, щоб не перевантажувати візуально подану вище модель (рис. 2.5), можливість корекції коефіцієнтів ймовірностей матриці переходів системи представлено окремо у вигляді таблиць, які розміщено у додатку 1.

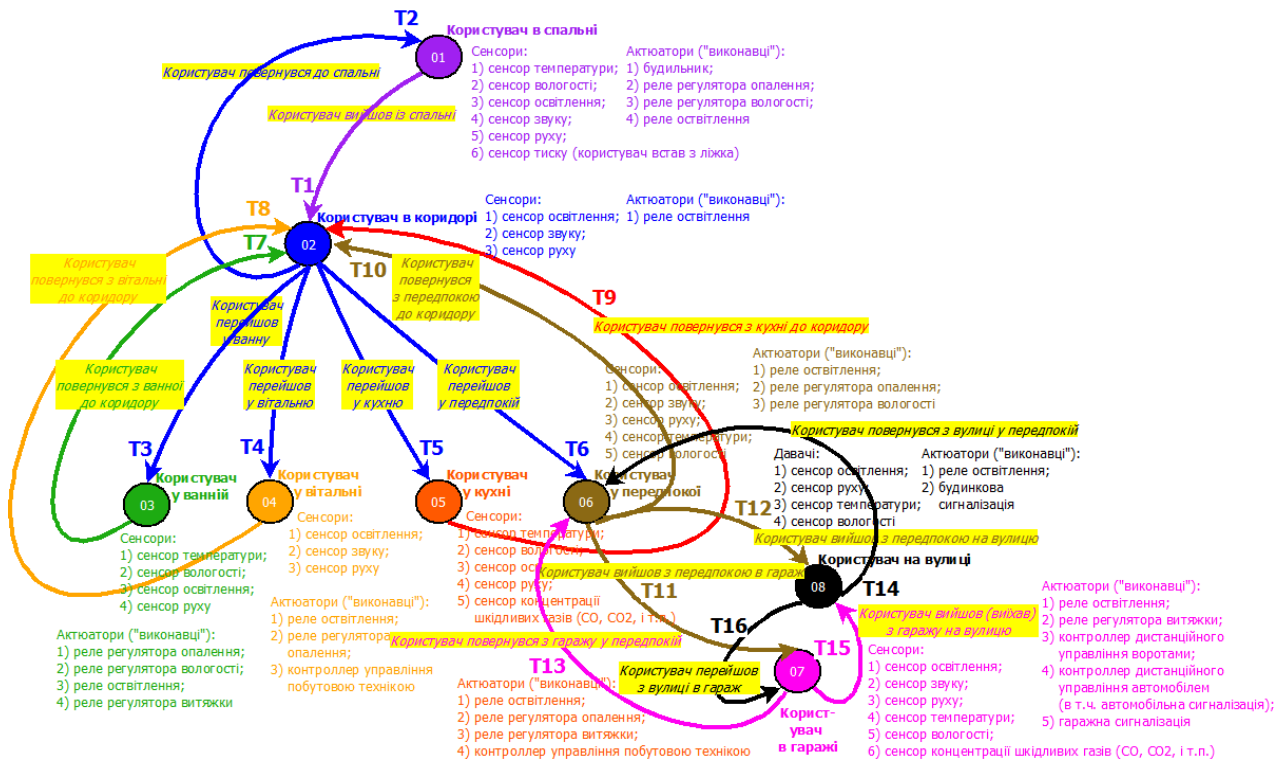


Рис. 2.4 - Вдосконалена схемна модель підвищення оперативності реагування системи РБ на основі ММ

Варіант використання лічильника ітерацій досягнення станів моделі для реалізації механізму корекції коефіцієнтів ймовірностей матриці переходів системи є надзвичайно гнучким підходом оскільки дає змогу з великою точністю враховувати найдрібніші нюанси поведінки користувача (звички, настрої, самопочуття тощо). Проте цей метод вимагає значних часових затрат на навчання системи, і будь-яке суттєве відхилення від «класичного» рутинного щоденного сценарію може вимагати додаткових коригувань функціональності.

Тому можливий також альтернативний варіант реалізації механізму корекції коефіцієнтів матриці переходів системи: без використання лічильника ітерацій, шляхом зміни значень ймовірностей переходів позначених як «Т1» – «Т16» (рис. 2.4), при зміні станів системи (додаток 1). У цьому випадку аналізується не ймовірність досягнення стану, а ймовірність спрацювання конкретних переходів.

Оскільки обидва підходи мають як переваги, так і недоліки, їх слід розглядати як альтернативні методи, а в деяких випадках – комбінувати,

оскільки вхідна таблиця переходу станів в обох випадках залишається ідентичною (однаковий набір стовпців і рядків).

Сенсори та актуатори, введені в модель, дають змогу за необхідності представивши дану модель в розгорнутій формі та розширити можливості моделі. Так, наприклад, розгорнута схемна модель (рис. 2.5) дає змогу детально аналізувати функціонування кожного окремого сенсора та актуатора, визначаючи рівень навантаження на кожен окремий функціональний елемент системи. Сенсори в такому разі доцільно програмувати не на безперервне опитування, а на зміну дискретного значення вимірюваної ними величини, оскільки для нас важливим є не так неперервне потактове опитування сенсора, як реагування виключно на зміну дискретного значення (наприклад, температури, вологості, тиску тощо), що забезпечує отримання корисної вхідної інформації.

Фрагмент розгорнутої схемної моделі (рис. 2.5) на рівні функціональних компонентів демонструє один із загальних станів системи, коли користувач перебуває в одній із кімнат. Аналогічно можна представити й усі інші стани моделі (стани «01»–«08» - рис. 2.5), подаючи для кожного з них розширений опис локальних процесів (активація/дія сенсорів і виконавців) [2].

Представлена розгорнута схемна модель дає змогу детально дослідити взаємодію між сенсорами та актуаторами в межах одного стану (наприклад, «користувач у спальні»), не впливаючи на решту частин моделі. Застосування принципу декомпозиції [14] дає змогу розробникам системи РБ послідовно удосконалювати окремі підмоделі без ризику порушення загальної структури. Це забезпечує можливість ефективного аналізу навантаження на конкретні давачі та виконавці, передбачати та тестувати роботу системи для різних сценаріїв експлуатації (наприклад, зміна погоди, різні добові ритми користувача, режим економії електроенергії тощо).

Розгорнутий вигляд моделі, як показано на рисунках 2.5 та 2.6 наочно відображає процеси, що відбуваються у межах стану «користувач у кімнаті» та

їхній вплив на коригування коефіцієнтів ймовірностей переходів, підвищуючи загальну оперативність реагування системи РБ.

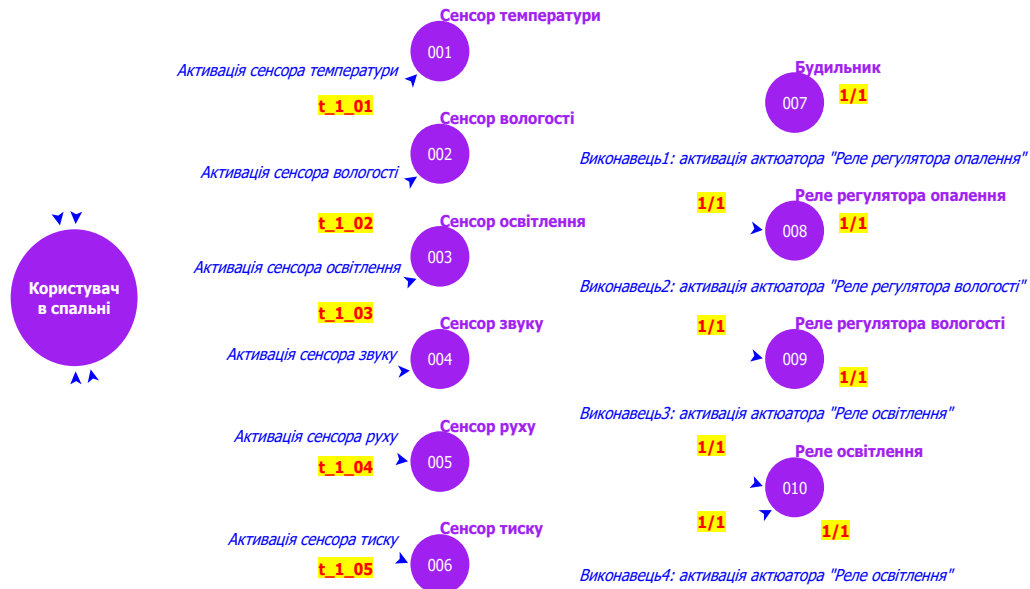


Рис. 2.5 - Фрагмент розгорнутої схемної моделі у формі графа підвищення оперативності реагування системи РБ

2.3 Моделі системи РБ з врахуванням функціональних компонентів системи на основі мереж Петрі та моделей Маркова

Перші системи РБ були надзвичайно простими - їх основне завдання полягало в наданні користувачу можливості дистанційного ввімкнення/вимкнення побутових приладів за допомогою пульта дистанційного керування [26]. Тоді складність таких систем полягала виключно в апаратній реалізації її виконавчих компонентів. Згодом основна функція систем РБ зводилася до раціонального розподілу ресурсів при підтриманні комфортних умов проживання (наприклад, енергозбереження, ефективне використання газу та водопостачання) [26]. На сьогоднішній день просте голосове керування – типу «дім, хочу послухати релаксуючу музику» або «дім, замов мою улюблену

піцу» – вже недостатньо. Користувачі прагнуть, щоб система РБ не просто реагувала на команди, а завчасно задовольняла їхні потреби, мінімізуючи участь людини у рутинних побутових процесах [26].

Система повинна навчитися «працювати на випередження», знаючи на кілька кроків вперед [2] куди переміститься користувач, його подальші дії та побажання, надавати відповідні сервіси до того, як користувач їх вимагатиме/

Наприклад, коли користувач повертається з роботи (наблизився до дому ближче, ніж на 5 км) система повинна:

- проаналізувати наявні в холодильнику продукти, сформувати та надіслати користувачу на мобільний телефон список продуктів, які потрібно купити, враховуючи гастрономічні побажання користувача та базуючись на статистичних даних про історію покупок;

- врахувати зовнішні погодні умови і налаштувати клімат-контроль для забезпечення оптимального мікроклімату всередині будинку;

- враховуючи поточний день тижня та персональний розклад користувача, приготувати користувачу каву, або протеїновий коктейль;

- увімкнути бойлер і підігріти теплу підлогу, забезпечивши комфорт користувачу при першому вході;

- підготувати для користувача вечірню вибірку найактуальніших новин та подій, що відбулися в світі, тощо.

Приклад, наведений вище, є одним із багатьох можливих сценаріїв, які потрібно передбачити та реалізувати в системі РБ. Оскільки кожна система РБ є абсолютно унікальною та повинна враховувати особливості свого конкретного користувача чи групи користувачів, то існує безліч варіантів сценаріїв, які може реалізовувати персональний РБ [2].

Саме тому сучасні системи РБ стають все складнішими. Тепер основна складність розроблення таких систем полягає вже не у апаратній реалізації функціональних компонентів, елементів та модулів системи, а у грамотному проектуванні, декомпозиції та можливості комплексного дослідження як всієї

системи, так і її окремих складових на більш ранніх етапах проєктування [14]. Раннє виявлення потенційних проблемних ділянок дає змогу значно зменшити витрати на їх виправлення, що надзвичайно важливо для систем, тісно пов'язаних із динамікою людських потреб та настроїв.

Одним із найбільш ефективних та широко використовуваних інструментів для аналізу складних систем на ранньому етапі проєктування є мережі Петрі. В арсеналі існує близько восьми основних типів МП для моделювання різноманітних процесів. Це - часові, стохастичні, функціональні, кольорові, інгібіторні, ієрархічні, потокові та мереж з пріоритетами [88], [89].

Основна ідея МП полягає у визначенні чіткої множини позицій P та переходів T мережі. При цьому, задаються стартові позиції мережі, в які поміщаються маркери, які через відповідні переходи можуть потрапляти в інші позиції, аж доки не досягнуть деяких фінішних позицій, що сигналізуватиме про завершення повного функціонального циклу мережі (моделі).

В загальному випадку, математична модель на основі теорії МП включає такі основні складові [2], [88]:

$$MPN = \{P_s, T_s, A_s, S_s, T_{ps}, P_s T_{ps}, A_s T_{ps}, C_s\}, \quad (2.6)$$

де:

- $P_s = \{p_1, p_2, \dots, p_n\}$ - множина позицій;
- $T_s = \{t_1, t_2, \dots, t_m\}$ - множина переходів;
- A_s - множина дуг, що зв'язують позиції з переходами (вхідні та вихідні дуги щодо кожного переходу);
- S_s - множина (вектор) початкового маркування мережі Петрі;
- T_{ps} - множина типів, які можуть бути призначені позиціям, переходам чи навіть маркерам залежно від конкретної розширеної концепції. Це дозволяє деталізувати, який вид інформації чи дія дозволена або можлива в кожному місці мережі або які маркери можуть «проходити» через відповідні переходи;

- $P_s T_{ps}$ – множина, яка вказує, які саме типи можуть бути пов’язані з певними позиціями. Це уточнює, яким чином «розширювати» класичні позиції, щоб диференціювати їх за функціями чи властивостями;
- $A_s T_{ps}$ - множина типів маркерів, що збуджують перехід;
- C_s - множина умов збудження переходів.

МП, які включають множини P_s , T_s , A_s , та S_s – відносять до простих МП, а МП, які включають додаткові множини – відносять до розширених МП [95].

Синтезуємо модель на основі звичайної МП [1], яка б дала змогу дослідити сценарій функціонування системи РБ, коли користувач ввечері повертається додому. Тут позиції представляють відповідні стани системи, а переходи – зміни між цими станами. Розроблена схемна модель зображена на рис. 2.6.

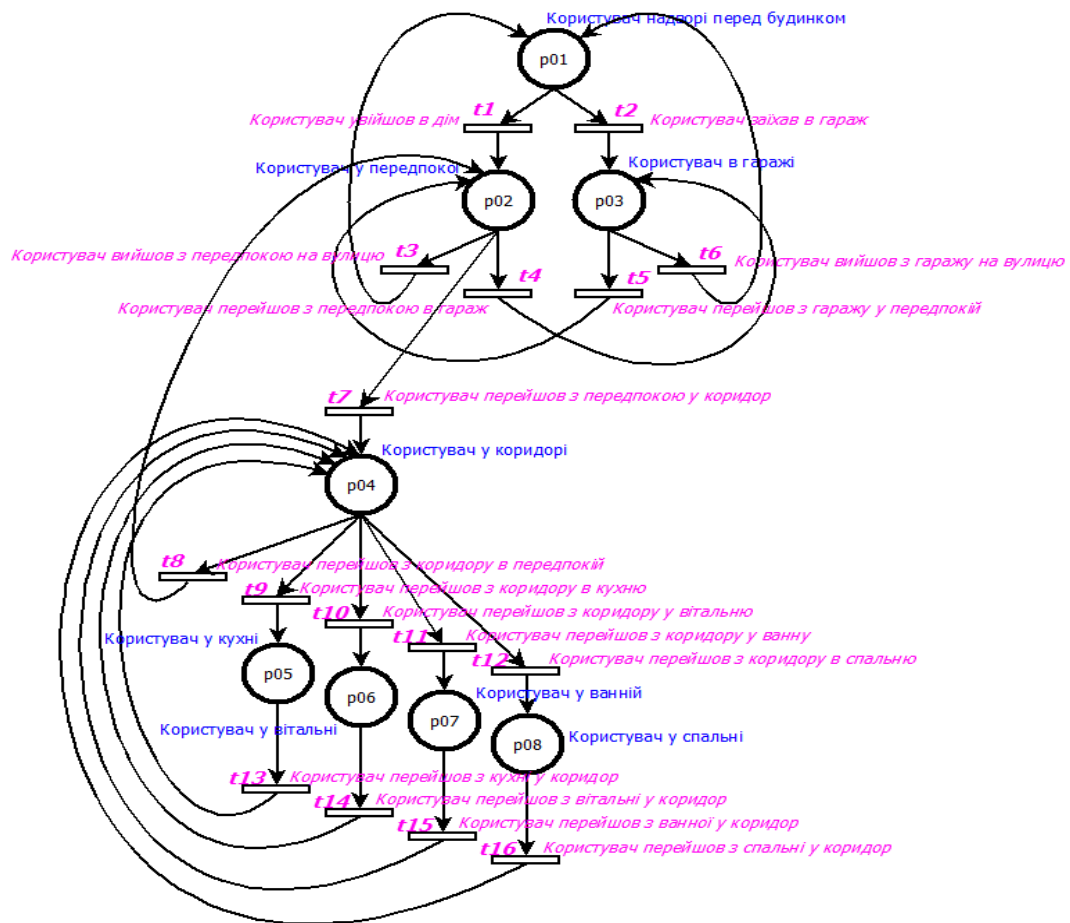


Рис. 2.6 - Схемна модель на основі простої МП для дослідження одного із сценаріїв функціонування системи РБ

Особливістю розробленої моделі на основі МП є відсутність чіткої фінальної (тупикової) позиції, що означає, що модель може працювати нескінченно. Це обумовлено її узагальненим характером. Згідно з конкретним сценарієм, у модель можна додати додаткові фінальні позиції, при потраплянні маркера в які, відбувалося б завершення повного функціонального циклу моделі [2].

Розроблена модель розпочинає свою роботу із поміщення маркера в стартову позицію p_{01} . Далі можливе спрацювання переходу t_1 (маркер переходить в позицію p_{02}) або спрацювання переходу t_2 (маркер потрапляє в позицію p_{03}). З позиції p_{03} маркер може повернутися в p_{01} через перехід t_6 , або перейти в позицію p_{02} через t_5 . З p_{02} маркер може повернутися в позицію p_{01} через t_3 , перейти в p_{03} через перехід t_4 , чи перейти в позицію p_{04} через t_7 . Після потрапляння маркера в p_{04} ми стикаємося з ситуацією, коли можливе спрацювання п'яти різних переходів (t_8-t_{12}).

Який саме з переходів спрацює – невідомо, оскільки в простій МП відсутня можливість задавання пріоритетності переходів. Навіть якби така можливість існувала, модель не задовольняла б вимоги системи РБ, оскільки сценарії функціонування тут не спираються на жорстко прописані пріоритети послідовності подій, а залежать від ймовірності настання конкретних подій, що спричиняють перехід системи з одного стану до іншого.

Отже, з'являється необхідність у використанні альтернативного підходу, який дає змогу уникнути поняття пріоритетності спрацювання переходів і вводить поняття ймовірності спрацювання. Одним із таких підходів є застосування моделей Маркова, що належать до класу стохастичних моделей і дають змогу враховувати випадкові процеси. ММ можна представити за допомогою наступного співвідношення [92]:

$$P_{s+1}^{(j)} = \sum_{k=0}^K I(< J S_{s-L+1} > = k) * q_{i \rightarrow j}^{(m_k)} , \quad (2.7)$$

де:

- $P_{s+1}^{(j)}$ - імовірність перебування (чи переходу) у стан j на кроці $s+1$;
- $I(< J S_{s-L+1} = k)$ – індикаторна (характеристична) функція, що дорівнює 1, коли $< J S_{s-L+1} >$ належить класу (типу) k , і дорівнює 0 в інших випадках;
- $q_{i \rightarrow j}^{(m_k)}$ – елемент матриці ймовірностей переходів $Q^{(m_k)}$ (марковська матриця) для «моди» m_k . Індекси i та j мають відповідати станам (елементам множини R), між якими розглядається перехід;
- $k = 0, \dots, K$ – індекс, який пробігає різні класи системи або сценарії, за якими формується набір марковських матриць переходів $Q^{(m_k)}$;
- $R = \{r_1, r_2, \dots, r_n\}$ – множина можливих станів системи

Окрім того, поєднання МП та ММ є досить злагодженим і не вимагає жодного штучного приведення МП до ММ чи навпаки [93]. Симбіоз МП та ММ здається очевидним, проте у літературі представлено обмежену кількість робіт, де було б подано детальний опис функціональності мереж Петрі–Маркова (МПМ) [93]. Отримати вдосконалену модель для дослідження сценаріїв функціонування системи РБ на основі МПМ можна шляхом додавання до моделі лише поняття ймовірності спрацювання переходів. Попри незначні модифікації, така модель (рис. 2.7) набуває якісно нового функціонального рівня, що дає змогу усунути неоднозначності, які виникають при спрацюванні розгалужених переходів та врахувати ймовірнісні процеси, властиві системам РБ.

В математичній формі модель рис. 2.7 можна записати у вигляді [93]:

$$\Pi = \{A, Z, I_A(Z), Q_A(Z)\} \text{ або } \Pi = \{A, Z, I_Z(A), Q_Z(A)\}, \quad (2.8)$$

де:

- A – скінченна множина позицій;
- Z – скінченна множина переходів;
- $I_A(Z), Q_A(Z)$ - вхідна та вихідна функції переходів;

- $I_z(A), Q_z(A)$ - вхідна та вихідна функції позицій.

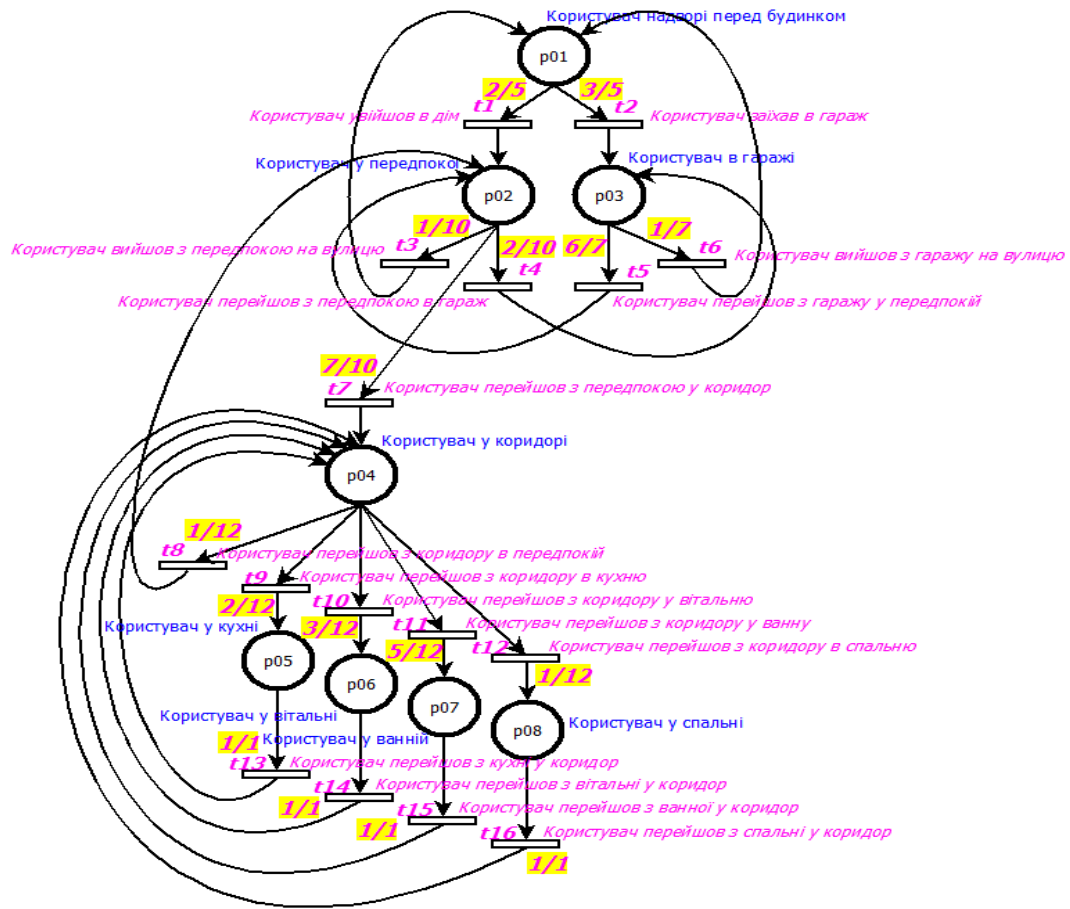


Рис. 2.7 - Модель на основі МПМ для дослідження одного із сценаріїв функціонування системи

Проте, розроблена модель на основі МПМ має суттєві недоліки – зокрема, ймовірності спрацювання переходів між станами визначені статично, що є неприйнятним для дослідженні систем типу РБ [2]. Тому необхідно забезпечити можливість динамічної корекції цих ймовірностей спрацювання переходів, або, в термінах класичних ММ, коригування коефіцієнтів матриці переходів моделі (табл.2.1).

В загальному випадку така матриця переходів має наступний вигляд:

$$Probability = \begin{bmatrix} p_1^1 & p_1^2 & \dots & p_1^m \\ p_2^1 & p_2^2 & \dots & p_2^m \\ \dots & \dots & \dots & \dots \\ p_n^1 & p_n^2 & \dots & p_n^m \end{bmatrix}, \quad (2.9)$$

де P_i^j - імовірність спрацювання переходу з позиції i в позицію j .

Таблиця 2.1

Таблиця переходу станів моделі на основі МПМ

	p01	p02	p03	p04	p05	p06	p07	p08
p01	—	2/5	3/5	—	—	—	—	—
p02	1/10	—	2/10	7/10	—	—	—	—
p03	1/7	6/7	—	—	—	—	—	—
p04	—	1/12	—	—	2/12	3/12	5/12	1/12
p05	—	—	—	1/1	—	—	—	—
p06	—	—	—	1/1	—	—	—	—
p07	—	—	—	1/1	—	—	—	—
p08	—	—	—	1/1	—	—	—	—

Реалізувати дану можливість корекції коефіцієнтів можна двома способами: з застосуванням лічильника ітерацій досягнення станів моделі та без застосування лічильника ітерацій досягнення станів моделі. І в одному, і в другому випадку ми отримаємо таблицю переходу станів моделі, в якій замість статичних даних у вигляді конкретних цифр як на рис. 2.7, ми матимемо:

- при першому способі реалізації встановлення значення коефіцієнта в залежності від поточного значення лічильника ітерацій (табл. 2.2);
- при другому способі реалізації при кожному переході моделі із одного стану в інший будуть задаватися нові значення коефіцієнтів для відповідної множини переходів (табл. 2.3).

Перший варіант дає змогу описати велику кількість імовірних шляхів розвитку різних ділянок сценарію, особливо коли модель неодноразово повертається до певних станів, проте вимагає значних часових затрат на розробку, впровадження та навчання системи у режимі відлагодження.

Другий метод є простішим як для реалізації, так і для навчання, і підходить для ситуацій, коли сценарій проходить за більш-менш класичною траєкторією з мінімальними відхиленнями; проте він не надає можливості

описати різноманітні імовірні варіанти розвитку одного й того самого місця сценарію. Тож в якості отримання оптимального рішення найкращим варіантом буде поєднання (комбінація) двох описаних способів.

Таблиця 2.2

Таблиця переходу станів моделі на основі МПМ з застосуванням лічильника ітерацій досягнення станів моделі

	p01	p02	p03	p04	p05	p06	p07	p08
p01	—	case iteration of: 1: 2/5 2: 4/5	case iteration of: 1: 3/5 2: 1/5	—	—	—	—	—
p02	case iteration of: 1: 1/10 2: 4/10 3: 1/10	—	case iteration of: 1: 2/10 2: 5/10 3: 1/10	case iteration of: 1: 7/10 2: 1/10 3: 8/10	—	—	—	—
p03	case iteration of: 1: 1/7 2: 1/7 3: 5/7	case iteration of: 1: 6/7 2: 6/7 3: 2/7	—	—	—	—	—	—
p04	—	case iteration of: 1: 1/12 2: 1/12 3: 1/12 4: 1/12 5: 3/12	—	—	case iteration of: 1: 2/12 2: 4/12 3: 4/12 4: 3/12 5: 2/12	case iteration of: 1: 3/12 2: 4/12 3: 4/12 4: 6/12 5: 4/12	case iteration of: 1: 5/12 2: 1/12 3: 2/12 4: 1/12 5: 1/12	case iteration of: 1: 1/12 2: 2/12 3: 1/12 4: 1/12 5: 2/12
p05	—	—	—	1/1	—	—	—	—
p06	—	—	—	1/1	—	—	—	—
p07	—	—	—	1/1	—	—	—	—
p08	—	—	—	1/1	—	—	—	—

Таблиця 2.3

Таблиця переходу станів моделі на основі МПМ без застосуванням лічильника
ітерацій досягнення станів моделі

	p01	p02	p03	p04	p05	p06	p07	p08
p01	–	t3:=1/10 t4:=2/10 t7:=7/10	t5:=6/7 t6:=1/7	–	–	–	–	–
p02	t1:=4/5 t2:=1/5	–	t5:=6/7 t6:=1/7	t8:=1/12 t9:=2/12 t10:=3/12 t11:=5/12 t12:=1/12	–	–	–	–
p03	t1:=1/10 t2:=9/10	t3:=1/10 t4:=1/10 t7:=8/10	–	–	–	–	–	–
p04	–	t3:=1/10 t4:=5/10 t7:=4/10	–	–	t13:=1/1 t8:=1/12 t9:=1/12 t10:=6/12 t11:=2/12 t12:=2/12	t14:=1/1 t8:=1/12 t9:=5/12 t10:=1/12 t11:=4/12 t12:=1/12	t15:=1/1 t8:=1/12 t9:=5/12 t10:=3/12 t11:=1/12 t12:=2/12	t16:=1/1 t8:=1/12 t9:=4/12 t10:=2/1 2 t11:=4/1 2 t12:=1/1 2
p05	–	–	–	t8:=1/12 t9:=1/12 t10:=6/12 t11:=2/12 t12:=2/12	–	–	–	–
p06	–	–	–	t8:=1/12 t9:=5/12 t10:=1/12 t11:=4/12 t12:=1/12	–	–	–	–
p07	–	–	–	t8:=1/12 t9:=5/12 t10:=3/12 t11:=1/12 t12:=2/12	–	–	–	–
p08	–	–	–	t8:=1/12 t9:=4/12 t10:=2/12 t11:=4/12 t12:=1/12	–	–	–	–

2.4. Побудова графа досяжності станів для розроблених моделей

В загальному випадку, граф досяжності станів системи РБ позначимо наступною формулою [96]:

$$G=(V, E, P), \quad (2.10)$$

де:

- V – множина вершин;
- E – множина ребер;
- P - множина (або функція) ймовірностей, прив'язаних до ребер.

На рис.2.8 подано граф досяжності станів розробленої моделі, який демонструє скінченність та досяжність усіх станів моделі з урахуванням ймовірностей переходів, так як дана модель побудована на основі МПМ. Як видно із графа досяжності станів на рисунку 2.8 у моделі відсутні термінальні (фінальні) стани. Це обумовлено узагальненим характером сценарію, який моделюється. Для того щоб отримати термінальні стани, сценарій необхідно доповнити відповідними фінальними положеннями сценаріїв.

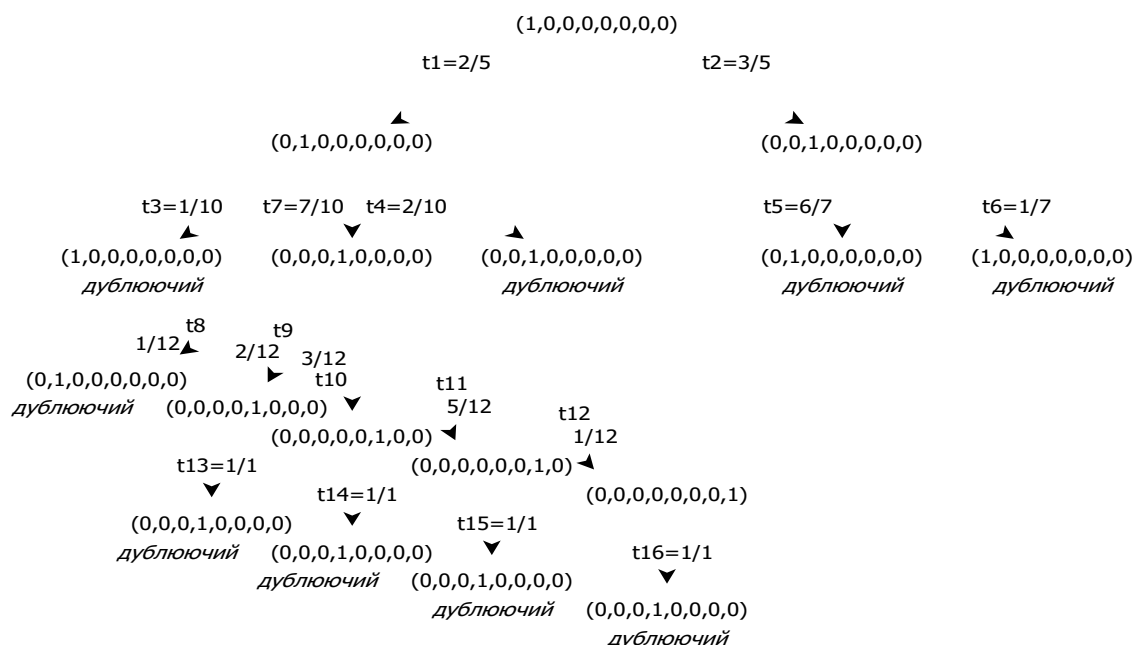


Рис. 2.8 - Граф досяжності станів розробленої моделі на основі МПМ

Наприклад, розглянемо сценарій, за яким користувач ввечері повертається з роботи додому. Термінальним (фінальним) етапом у такому

сценарії може бути ситуація, коли після відвідин ванної кімнати та кухні користувач переходить у вітальню та зручно розташовується на дивані перед телевізором. Для опису такого сценарію до моделі необхідно ввести додаткові стани, наприклад, стан p_{09} — "Користувач вмовстився на диван перед телевізором".

Проте існує інше рішення, яке, крім того, дає змогу ще більше розширити функціональні можливості розроблюваної моделі. Рішення полягає у додатковому введенні в розроблену на основі МПМ модель функціональних компонентів проєктованої системи РБ - сенсорів та актуаторів. Це забезпечить можливість гнучкої декомпозиції моделі, що є надзвичайно потрібним при розробленні складних систем, до яких відносяться системи РБ [14]. Таким чином, ми отримаємо модель на основі МПМ, доповнену функціональними компонентами системи - сенсорами та актуаторами (рис.2.9).

Запропонований підхід побудови моделей на основі МПМ, інтегрованих із функціональними компонентами системи РБ, дає змогу здійснювати декомпозицію на будь-якому рівні деталізації — що є безсумнівною перевагою цього методу [14]. Крім того, за потреби в модель можна завжди додатково включати нові стани та переходи як для відображення функціональних компонентів системи, так і для більш ґрунтовного аналізу поведінки системи у конкретних сценаріях розвитку подій.

Приклад математичної моделі графа досяжності станів матиме вигляд [2]:

$$G := \left(\begin{array}{c} Gp_1^1 = \begin{bmatrix} p_1^1 & p_1^2 & \dots & p_1^m \\ p_2^1 & p_2^2 & \dots & p_1^m \\ \dots & \dots & \dots & \dots \\ p_n^1 & p_n^2 & \dots & p_n^m \end{bmatrix}, Gp_1^2 = \begin{bmatrix} p_1^1 & p_1^2 & \dots & p_1^m \\ p_2^1 & p_2^2 & \dots & p_1^m \\ \dots & \dots & \dots & \dots \\ p_n^1 & p_n^2 & \dots & p_n^m \end{bmatrix}, \dots, Gp_1^y = \begin{bmatrix} p_1^1 & p_1^2 & \dots & p_1^m \\ p_2^1 & p_2^2 & \dots & p_1^m \\ \dots & \dots & \dots & \dots \\ p_n^1 & p_n^2 & \dots & p_n^m \end{bmatrix}, \\ Gp_2^1 = \begin{bmatrix} p_1^1 & p_1^2 & \dots & p_1^m \\ p_2^1 & p_2^2 & \dots & p_1^m \\ \dots & \dots & \dots & \dots \\ p_n^1 & p_n^2 & \dots & p_n^m \end{bmatrix}, Gp_2^2 = \begin{bmatrix} p_1^1 & p_1^2 & \dots & p_1^m \\ p_2^1 & p_2^2 & \dots & p_1^m \\ \dots & \dots & \dots & \dots \\ p_n^1 & p_n^2 & \dots & p_n^m \end{bmatrix}, \dots, Gp_2^y = \begin{bmatrix} p_1^1 & p_1^2 & \dots & p_1^m \\ p_2^1 & p_2^2 & \dots & p_1^m \\ \dots & \dots & \dots & \dots \\ p_n^1 & p_n^2 & \dots & p_n^m \end{bmatrix}, (0,1) \quad (0,1) \quad \dots \quad (0,1) \\ \dots & \dots & \dots & \dots \\ (0,1) \quad (0,1) \quad \dots \quad (0,1) \end{array} \right), \quad (2.11)$$

де Gp_{zz}^z - множина станів моделі поточного відгалуження дерева досяжності.

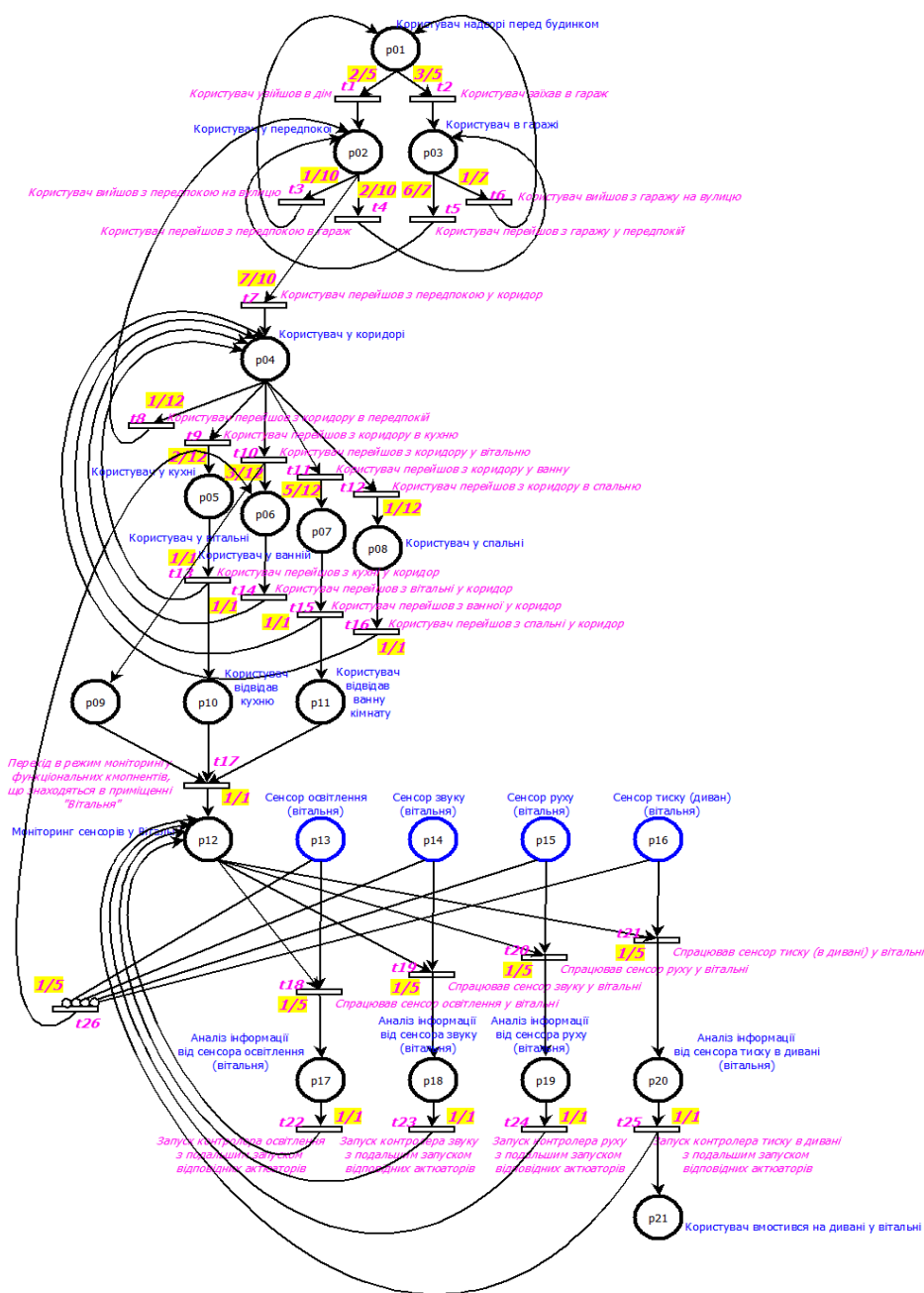


Рис. 2.9 - Модель системи на основі МПМ, доповнена функціональними компонентами системи

Представлена на рис.2.9 модель зображає лише основні принципи побудови моделей на основі МПМ, доповнених функціональними компонентами і дає розробнику широке поле для діяльності [2]. Також модель

The diagram illustrates a game tree for a 10-player game. The root node is a chance node with two branches: $t_1=2/5$ and $t_2=3/5$. The tree branches out through various nodes, many of which are labeled "дублирующий" (redundant). The tree ends at a terminal node labeled "терминальный". A blue line highlights a specific path through the tree, starting from the root and ending at the terminal node.

Висновки до розділу 2

- 76

додаткових компонентів значною мірою розширює функціональні можливості моделі. Удосконалена модель дає змогу здійснювати декомпозицію системи на довільному рівні (в тому числі на рівні функціональних елементів системи – сенсорів та актуаторів). Вона також передбачає реалізацію двох можливих варіантів механізму корекції коефіцієнтів ймовірностей матриці переходів системи, забезпечуючи можливість реалізації різноманітних шляхів розвитку конкретних сценаріїв. Окрім цього, модель сприяє підвищенню оперативності реагування системи шляхом досягнення стану системи до того, як система фактично опиниться в даному стані у випадку коли імовірність досягнення даного стану перевищила деяке порогове значення. Таким чином, діючи на випередження.

Отже, розроблена модель володіє рядом необхідних додаткових характеристик, які дають змогу здійснити якісне підвищення оперативності реагування систем РБ.

Розроблено модель для системи РБ на основі теорії мереж Петрі-Маркова, доповнену функціональними компонентами (сенсорами та актуаторами). Запропонована модель дає змогу провести комплексний аналіз різноманітних сценаріїв розвитку подій у системах РБ; дослідити динаміку та надійність як всієї системи так і окремих її складових ще на рівні системного проєктування; виключити неоднозначні ситуації при спрацюванні розгалужених переходів; врахувати імовірнісні процеси, що притаманні системі РБ; здійснити декомпозицію довільного рівня [14].

Найважливішою перевагою даного підходу є оптимальне інтегрування функціональних компонентів (сенсорів та актуаторів) у модель на основі мереж Петрі-Маркова, що дозволяє досягти якості нового функціонально-методологічного рівня, з надзвичайно гнучкими можливостями декомпозиції.

Побудовано графи досяжності станів розроблених моделей, які демонструють скінченність і досяжність усіх станів та дають змогу дослідити динаміку функціонування моделей.

РОЗДІЛ 3. МЕТОДИ СИНТЕЗУ ТА НАВЧАННЯ МОДЕЛЕЙ СИСТЕМ «РОЗУМНИЙ БУДИНОК» НА ОСНОВІ МОДЕЛЕЙ ПЕТРІ-МАРКОВА ТА ДОПОВНЕНИХ ФУНКЦІОНАЛЬНИМИ КОМПОНЕНТАМИ

3.1. Розроблення методу синтезу моделей систем РБ, створених на основі моделей Петрі-Маркова та доповнених функціональними компонентами

При розробленні систем РБ необхідно враховувати ряд ключових факторів, таких як: динаміка системи, надійність її роботи, а також випадкові процеси в ході функціонування системи та її взаємодії з користувачем. Беручи до уваги високу вартість таких систем, особливої актуальності набуває потреба в ідентифікації потенційно проблемних ділянок ще на ранніх етапах проєктування. Складність архітектури систем РБ вимагає застосування багаторівневої декомпозиції для забезпечення ефективного моделювання та управління такими системами. Беручи до уваги перелічені вище фактори, одним із найбільш вдалих інструментів для їх реалізації є моделі Петрі-Маркова. Вони дають змогу досліджувати динаміку та надійність системи ще на системному рівні проєктування, враховувати ймовірності випадкових процесів, що відбуваються всередині системи в ході її функціонування, а також з легкістю дають змогу реалізовувати розгалужені схеми в найрізноманітніших сценаріях розвитку подій в процесі функціонування розроблюваної системи РБ.

Проте, використання базових моделей Петрі-Маркова обмежує можливість багаторівневої декомпозиції, так само як і можливість динамічної корекції коефіцієнтів ймовірностей матриці переходів системи (усталено всі ці коефіцієнти ймовірностей задаються статично). Для подолання цих обмежень запропоновано модифікацію МПМ, в результаті чого отримано моделі якісно нового функціонально-методологічного рівня. Ці вдосконалені моделі

додатково доповнені можливістю динамічної корекції коефіцієнтів ймовірностей матриці переходів системи, а також детальним та чітким механізмом багаторівневої декомпозиції моделі від найбільш глобального (узагальнюючого) рівня – з точки зору користувача до найбільш деталізованого рівня – рівня функціональних компонентів системи (сенсорів та актуаторів), які представляють апаратну складову системи. Це дає змогу реалізовувати можливість вдалого поєднання МПМ з апаратною складовою системи [20].

Розглянемо детальніше усі вище перелічені переваги розроблених моделей систем РБ, побудованих на основі МПМ та доповнених функціональними компонентами на конкретному прикладі моделювання сценарію, коли користувач ввечері, повернувшись з роботи додому вмищується на дивані у вітальні та готується до сну [4]. Початковим станом даного сценарію буде стан «Користувач на дивані», кінцевим станом – «Користувач в ліжку». Модель, побудована для цього сценарію, наведена на рисунку 3.1 [15].

У додатку 2 представлений граф досяжності станів розробленої моделі. Він демонструє скінченність та досяжність кожного із станів розробленої моделі [5].

В таблиці 3.1, наведено коефіцієнти ймовірностей матриці переходів системи. Тут реалізований механізм динамічної корекції значень даних коефіцієнтів із застосуванням лічильника ітерацій досягнення станів. Для переходів працюють лічильники спрацювань із кожного вихідного стану в кожен доступний результуючий стан, і в залежності від значення конкретного лічильника ми отримуємо значення відповідного коефіцієнта переходу [4], [15].

Таким чином, побудована модель, дає змогу досліджувати один із імовірних сценаріїв розвитку подій в ході функціонування розроблюваної системи РБ та її взаємодії із користувачем. У загальному випадку кількість можливих сценаріїв експлуатації системи може бути досить великою. Всі ці сценарії попередньо погоджуються із замовником (та/або користувачами, якщо вони не є замовниками) розроблюваної системи РБ [4], [20].

Таблиця 3.1

Таблиця переходу станів розробленої моделі

№	Із стану	В стан	Коефіцієнт ймовірності переходу	№	Із стану	В стан	Коефіцієнт ймовірності переходу	№	Із стану	В стан	Коефіцієнт ймовірності переходу
1	01	02	1/1	2	02	07	1/5	3	02	08	1/5
4	02	09	1/5	5	02	10	1/5	6	11	12	1/1
7	12	11	case iteration of: 1: 1/12 2: 3/12 3: 4/12 4: 3/12	8	12	13	case iteration of: 1: 1/12 2: 1/12 3: 1/12 4: 1/12	9	12	14	case iteration of: 1: 5/12 2: 3/12 3: 2/12 4: 1/12
10	12	15	case iteration of: 1: 4/12 2: 4/12 3: 2/12 4: 3/12	11	12	16 (35)	case iteration of: 1: 1/12 2: 1/12 3: 3/12 4: 4/12	12	13	11	2/5
13	13	17	1/5	14	13	18	2/5	15	17	13	4/5
16	17	18	1/5	17	18	13	4/5	18	18	17	1/5
19	35	29	1/6	20	35	30	1/6	21	35	31	1/6
22	35	32	1/6	23	35	33	1/6	24	35	34	1/6
25	29	35	1/1	26	30	35	1/1	27	31	35	1/1
28	32	35	1/2	29	33	35	1/5	30	34	35	1/2
31	22	36	1/1	32	37	38	1/1	33	38, 16	12	1/2
34	19, 20, 21	02	1/15	35	10	01, 02	1/1	36	32, 33, 34, 36	39	1/16

Саме в зв'язку з великою кількістю імовірних сценаріїв виникає необхідність розроблення універсального методу побудови моделей систем РБ,

створених на основі МПМ та доповнених функціональними компонентами, які б дали змогу описати всі ці сценарії функціонування системи та її взаємодії з користувачем. Як альтернатива - побудова однієї загальної моделі, яка б включала в себе всі задекларовані сценарії.

З точки зору швидкодії, з врахуванням обчислювальних потужностей сучасних програмно-апаратних засобів, обидва варіанти є прийнятними. Однак побудова узагальненої моделі є необхідною, оскільки лише така модель може вважатися повноцінною.

Множину моделей окремих сценаріїв можна сформувати на основі узагальненої моделі шляхом обходу дерева станів – від початкового до кінцевого стану кожного конкретного сценарію.

Хоч узагальнена модель є необхідною в процесі проєктування системи, використання саме множини похідних моделей сценаріїв, згенерованих із завершеної узагальненої моделі, є більш раціональним в процесі функціонування систем РБ [4].

Робота розробленого алгоритму (рис. 3.2) побудови моделей систем РБ розпочинається із визначення вхідних параметрів, зокрема [5], [15]:

- погодження з користувачем (замовником) наявних рівнів декомпозиції моделі;
- погодження наявних станів для кожного із узгоджених рівнів декомпозиції;
- погодження зв'язків (переходів) між станами;
- погодження основних сценаріїв, їх початкових та кінцевих станів.

Після узгодження вхідних параметрів розробнику необхідно перевірити такі умови:

- чи досягнутий та описаний найнижчий рівень декомпозиції моделі – рівень функціональних компонентів;

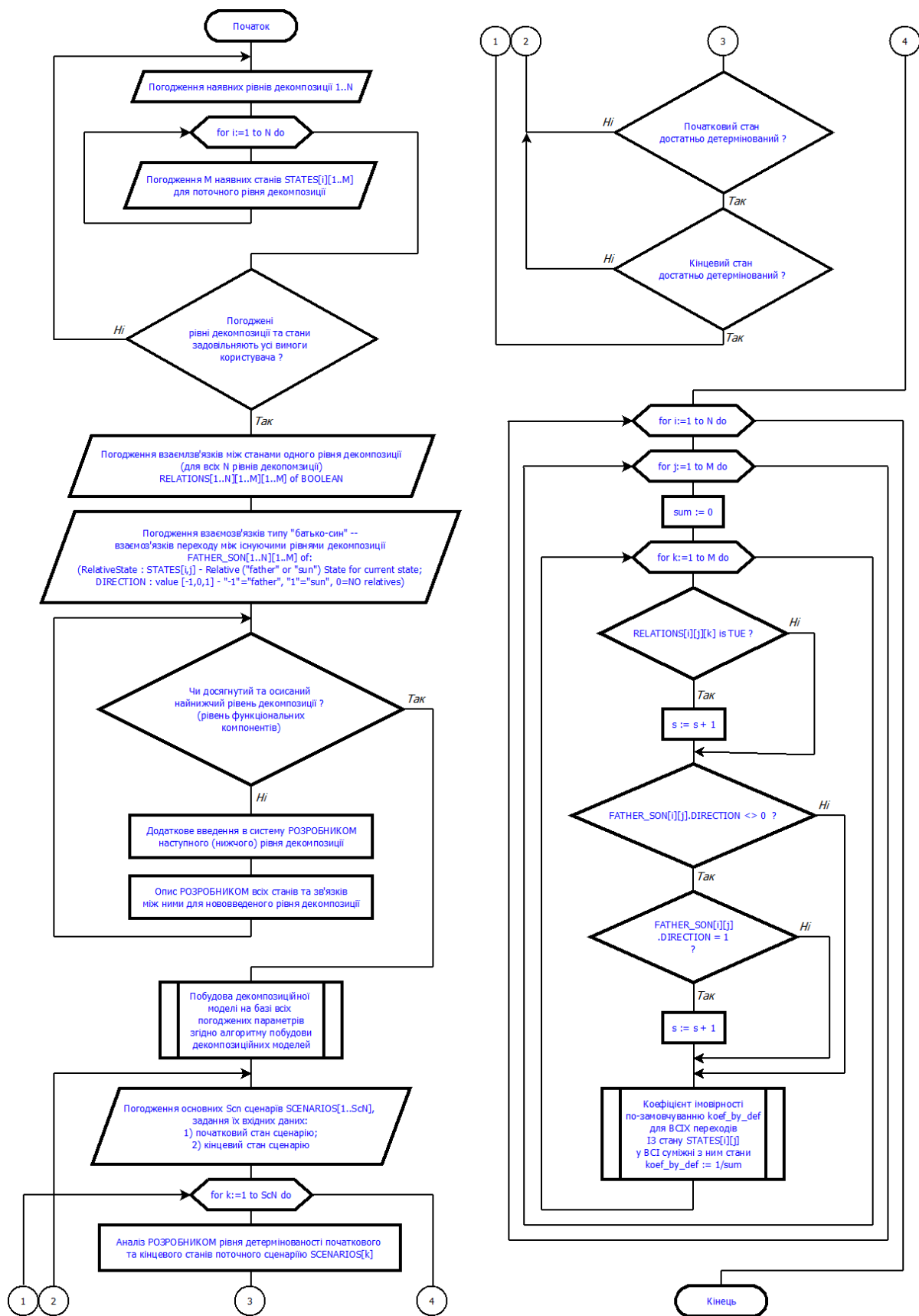


Рис. 3.2 - Блок-схема алгоритму побудови моделей систем РБ, створених на основі моделей Петрі-Маркова та доповнених функціональними компонентами

- чи достатньо детерміновані початкові стани кожного із погоджених сценаріїв;

- чи достатньо детерміновані кінцеві стани кожного із погоджених сценаріїв.

У випадку, якщо під час побудови моделі згідно з алгоритмом (рис. 3.2) не досягнутий найнижчий рівень декомпозиції (рівень функціональних компонентів), розробнику необхідно додатково внести в систему ще один (або кілька) нижчих рівнів разом із відповідними станами та переходами. аж поки не буде досягнутий найнижчий рівень декомпозиції. Тільки після цього відбувається запуск процедури побудови повної декомпозиційної моделі на основі всіх узгоджених параметрів відповідно до алгоритму, що базується на моделях Петрі-Маркова (МПМ), доповнених функціональними компонентами.

Після завершення побудови структури моделі необхідно перевірити детермінованість початкових та кінцевих станів кожного зі сценаріїв, попередньо погоджених із клієнтом (замовником). Стан вважається достатньо детермінованим, якщо його можливо технічно ідентифікувати з високою імовірністю, максимально наближеною до 100%. Зокрема, нижче на рисунку представлені приклади недостатньо детермінованого (рис.3.3, а) та достатньо детермінованого (рис.3.3, б) станів фрагменту моделі [4].

Недостатня детермінованість стану моделі може бути спричинена рядом факторів, таких як:

- невиправдана складність (або неможливість) технічної реалізації однозначної ідентифікації даного стану з імовірністю, близькою до 100%;

- даний стан належить до станів вищих узагальнених рівнів декомпозиції та не потребує деталізації з методологічної або логістичної точки зору для подальшої декомпозиції та розгалуження на вітки станів нижчих рівнів;

- отриманий стан належить до станів верхніх рівнів декомпозиції, і при цьому вже є описані нижчі рівні декомпозиції. В такому випадку достатньо

обрати (погодити з замовником) в якості кінцевого стану один із станів нижчих рівнів декомпозиції;

– недостатній рівень декомпозиції дерева станів у розглядуваній гілці моделі.

Наприклад, якщо початковим станом сценарію є "Користувач вийшов з автомобіля ввечері після роботи", а кінцевим — "Користувач заснув у ліжку", то ідентифікація факту сну з імовірністю, близькою до 100%, можлива лише за допомогою складних технічних засобів (наприклад, комп'ютерної томографії), вартість яких може перевищити бюджет системи.

Або ж погоджений стан "Користувач у будинку" належить до другого рівня декомпозиції та є надто загальним. Якщо вже існують розроблені гілки нижчих рівнів (наприклад, "у кухні", "у спальні"), раціонально вибрати один із них як кінцевий стан сценарію [15].

У випадку на прикладі, наведеному на рис.3.4, ми маємо справу із недостатнім рівнем декомпозиції гілки дерева станів для стану моделі p_{18} «Користувач в гаражі» (рис. 3.3, а), який був погоджений в якості кінцевого стану для деякого сценарію. Для однозначної ідентифікації розглядуваного стану необхідно додатково внести в модель ряд станів функціонального рівня декомпозиції (рис. 3.3, б) [4], [5].

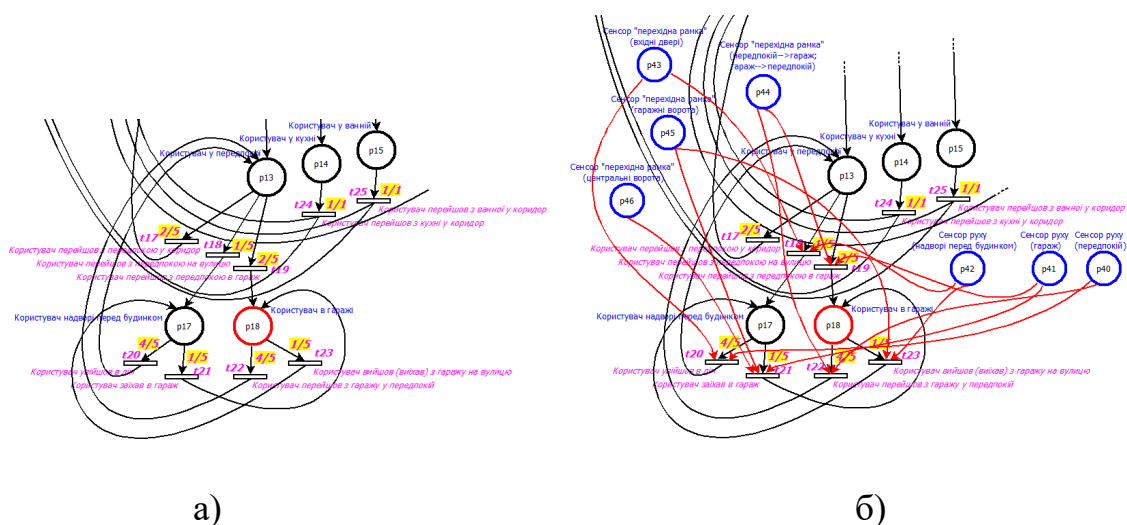


Рис.3.3 - Приклади недостатньо (а) та достатньо (б) детермінованих станів фрагменту моделі

Після верифікації детермінованості всіх початкових та кінцевих станів для всіх погоджених сценаріїв функціонування моделі, згідно алгоритму (рис. 3.2), відбувається перехід до фінальної стадії побудови моделі – встановлення усталених значень для всіх переходів моделі.

Розроблений алгоритм (рис. 3.2), а також модель (рис. 3.1) складають основу розробленого методу побудови моделей систем РБ, створених на основі МПМ та доповнених функціональними компонентами [5].

3.2. Розроблення методу навчання моделей систем РБ, створених на основі моделей Петрі-Маркова та доповнених функціональними компонентами

Однією з основних характеристик систем РБ є їх динаміка та імовірнісні процеси, що мають місце в процесі роботи таких систем. Тому, для моделювання роботи системи РБ за основу обрано мережі Петрі-Маркова, що дає можливість не лише ефективно здійснювати моделювання саме такого типу систем, а й досліджувати кількісні характеристики ймовірностей переходів між станами системи, базуючись на історії її роботи [9].

Мережі Петрі в даному контексті, служать саме для відображення динаміки, а ланцюги Маркова дають можливість дослідити імовірнісні процеси та стани, що виникають в процесі комунікації між користувачем та самою системою розумного будинку.

Розроблена в роботі модель управління адаптивною системою розумного будинку ґрунтується на мережах Петрі-Маркова, включає функціональні елементи і дає змогу врахувати передісторію подій та поведінку користувача, реалізується алгоритмом, що можна описати наступними кроками [9]:

Крок 1. Визначити множину умов для переходів системи розумного будинку з одних станів в інші: $|P| = m$; $P = S \cup A$, де S - множина сенсорів; A - множина активаторів.

Крок 2. Визначити множину подій, що можуть мати місце в системі розумного будинку: $|T| = d$ Ці події представляють собою переходи системи з одних станів в інші.

Крок 3. Визначити функції, що задають залежності між подіями, передумовами для цих подій та постподієвими станами системи розумного будинку:

$J : T \times P \rightarrow \{0,1\}$ - функція слідування.

$O : P \times T \rightarrow \{0,1\}$ - функція передування.

Крок 4. Задати множину порогових значень ймовірностей подій, що відповідають за спрацювання активаторів системи розумного будинку:

$|H| = h$ - функція слідування; $h \leq d$.

Крок 5. Задати початкове маркування: $M_0 = [M_0(p_1)M_0(p_2) \dots M_0(p_m)]$; $M(p_i) \in \mathbb{N}_0$; $M_0 : P \rightarrow \mathbb{N}$.

Крок 6. Визначити початковий розподіл ймовірностей:

$$P(X_0 = s) = q_0(s); \forall s \in T \quad (3.1)$$

- для першої ітерації роботи системи:

$s_1 = s_2 = \dots = s_j = \frac{1}{j}$, де s_i - ймовірність події i ; j – кількість конкуруючих подій;

- для наступних ітерацій роботи системи: зчитати значення з бази даних статистики, зібраної та обчисленої на основі попередніх прогонів роботи системи розумного будинку як:

$$s_i = \frac{RELATIONS.SUM}{GeneralSum}, \quad (3.2)$$

де $RELATIONS.SUM$ – фактична кількість активації даної події (спрацювань даного переходу) згідно з базою даних статистики; $GeneralSum$ – загальна кількість активацій конкуруючих подій (спрацювань усіх вихідних із даного стану переходів).

Крок 7. Визначити матрицю перехідних ймовірностей на основі розподілу ймовірностей:

$$P(X_{n+1} = s_{n+1} | X_n = s_n) = p(s_n, s_{n+1}), \forall (s_{n+1}, s_n) \in T \times T \quad (3.3)$$

Крок 8. Визначити матриці:

- преінцидентності: $A^- = [a_{ij}^-]$;
- постінцидентності: $A^+ = [a_{ij}^+]$;
- інцидентності: $A = A^+ - A^-$:

$$a_{ij}^- = \begin{cases} 1, \text{ якщо умова } j - \text{ вхідна умова для події } i \\ 0, \text{ якщо умова } j - \text{ не є вхідною умовою для події } i \end{cases}$$

$$a_{ij}^+ = \begin{cases} 1, \text{ якщо стан } j - \text{ вихідний стан події } i \\ 0, \text{ якщо стан } j - \text{ не є вихідним станом події } i \end{cases}$$

Крок 9. Визначити вектор керування для поточного маркування M_k :

$$v_{k,i} = \begin{cases} 1, \text{ якщо } t_i - \text{ активний в маркуванні } M_k, i = \overline{1, n} \\ 0, \text{ якщо } t_i - \text{ неактивний в маркуванні } M_k, i = \overline{1, n} \end{cases}$$

Умова активності події t_i в маркуванні M_k :

$$\forall M_{k,i} \geq a_{ij}^-, j = \overline{1, m} \quad (3.4)$$

Крок 10. Визначити маркування на наступному $(k+1)$ – му кроці згідно рівняння стану:

$$M_{k+1} = M_k + A^T v_k, \quad k = 0, 1, 2, \dots \quad (3.5)$$

$$M_n = M_0 + A^T \sum_{k=0}^{n-1} v_k, \quad (3.6)$$

де $\{v_0, v_1, \dots, v_{n-1}\}$ - послідовність векторів керування, що переводять стан системи з початкового M_0 у деякий M_n .

Крок 11. Визначити розподіл ймовірностей в момент $(n+1)$:

$$q_{n+1}(t_{n+1}) \stackrel{\text{def}}{=} P(X_{n+1} = t_{n+1}) = \sum_{t \in T} P(X_n = t) P(X_{n+1} = t_{n+1} | X_n = t) = \sum_{t \in T} q_n(t) p(t, t_{n+1}) \quad (3.7)$$

$$(q_0)_i = q_0(t_i) = P(X_0 = t_i), \quad (3.8)$$

$$p_{i,j} = p(t_i, t_j) = P(X_{n+1} = t_j | X_n = t_i), \quad (3.9)$$

$$(q_n)_i = q_n(t_i) = P(X_0 = t_i), \quad (3.10)$$

$$q_{n+1} = q_n p, \quad (3.11)$$

$$q_{n+2} = q_{n+1}p = (q_n p)p = q_n p^2, \quad (3.12)$$

.....

$$q_{n+m} = q_n p^m, \quad (3.13)$$

де m – кількість кроків (подій), що повинні відбутись для активації переходу $t_i \in T^*$

$T^* \in T$ – множина подій, що спричиняють спрацювання активаторів

$$M_{m,i} \geq a_{ij}^-, \quad m = \min(m) \quad (3.14)$$

Крок 12. Перевірити виконання умови реалізації дії на випередження (спрацювання t_j):

$$[q_{n+1}(t_{n+1})]_j \geq H_j, \quad t_j \in T^* \quad (3.15)$$

Реалізація сучасних систем РБ, які б відповідали актуальним вимогам, потребує розроблення нових або вдосконалення вже існуючих методів та засобів проектування, моделювання та дослідження [9]. У попередньому розділі дисертаційної роботи було представлено метод побудови моделей систем РБ, створених на основі моделей Петрі-Маркова та доповнених функціональними компонентами. Моделі, розроблені на основі даного методу, володіють рядом переваг, оскільки вони дають змогу:

- реалізовувати широкий спектр сценаріїв взаємодії користувачів з системою;
- досліджувати динаміку та надійність як всієї системи, так і окремих її компонентів ще на етапі системного проектування;
- описувати складні розгалужуючі схеми, виключивши неоднозначні ситуації, що можуть мати місце;
- надати кількісну оцінку імовірнісним процесам, які виникають в результаті взаємодії системи з користувачами, використовуючи можливість динамічної корекції значень коефіцієнтів матриці переходів системи;

– здійснити комплексну декомпозицію системи з органічним включенням апаратної частини (рівня функціональних компонентів) у стохастичну модель.

Таким чином, запропонований метод дає змогу побудувати максимально деталізовану модель системи РБ. Водночас це лише перший етап на шляху до створення повнофункціональної системи. Для повного функціонування така модель повинна додатково пройти етап навчання, в ході якого значення коефіцієнтів ймовірності переходів, задані за замовчуванням, замінюються на реальні значення, отримані на основі статистичних даних, зібраних під час дослідної експлуатації системи в умовах її взаємодії з користувачами. Таким чином, навчання моделі виступає в якості окремої фази розробки як самої моделі зокрема, так і системи вцілому, і потребує розроблення відповідного методу навчання моделей систем РБ, побудованих на основі МПМ, та доповнених функціональними компонентами.

Процес навчання моделі поділяється на два етапи: етап збору та опрацювання вхідної статистичної інформації та етап самого навчання моделі на основі отриманої результуючої статистичної інформації. Для збору та опрацювання вхідної статистичної інформації в процесі навчання розробленої моделі проєктованої системи РБ запропоновані три основні режими: 1) ручний; 2) автоматизований; 3) автоматичний [9].

В *ручному режимі* (рис. 3.4) основними засобами для збору інформації є портативні відеокамери, що розміщуються по всьому будинку з метою фіксації дій користувачів та їх взаємодії із системою [22]. Опрацювання отриманої інформації здійснює оператор, який аналізує відеозаписи, ідентифікує стан системи, визначає переходи між станами, що спрацьовують та формує відповідну статистику, яка надалі використовується для навчання моделі.

Серед переваг такого збору статистичної інформації можна виділити:

– ефективність при необхідності здійснити якомога швидший процес збору та опрацювання статистичної інформації за короткий проміжок часу;

- відносну фінансову економію у випадку швидкого процесу збору та опрацювання статистичної інформації;
- гнучке масштабування;

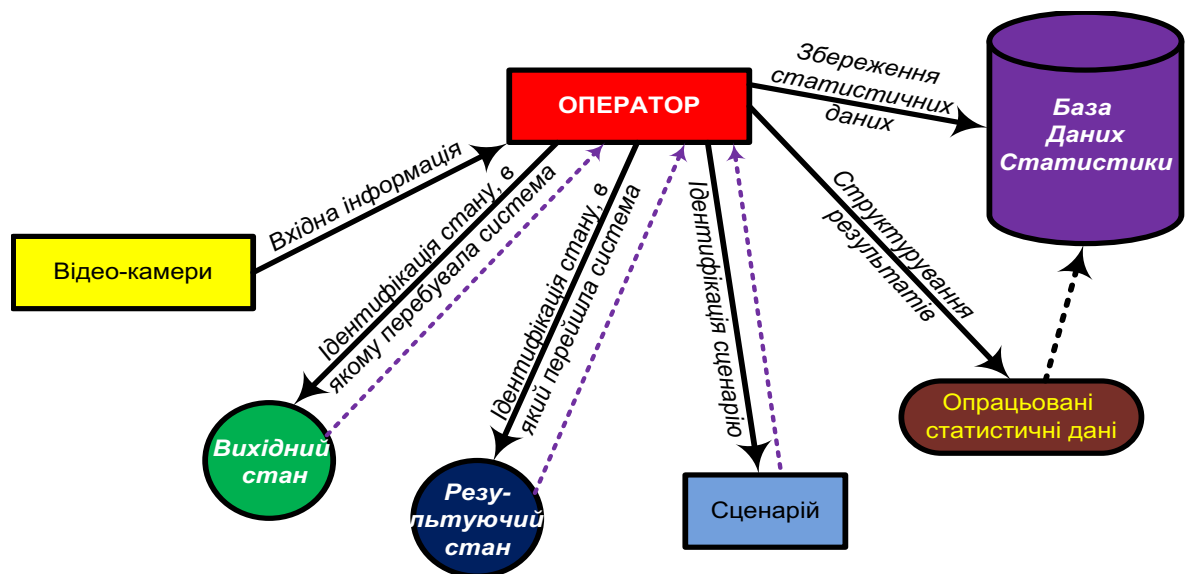


Рис. 3.4 - Структурна схема ручного режиму збору та опрацювання вхідної інформації

– можливість проведення паралельного збору та опрацювання статистичної інформації для довільної кількості користувачів системи. При цьому можливість технічної реалізації такого паралельного збору та опрацювання статистичної інформації при необхідності може бути реалізована в найкоротші строки та із залученням мінімальних фінансових витрат;

– можливість залучення довільної кількості «каналів» для збору та опрацювання однієї і тієї ж вхідної статистичної інформації. Це дає змогу підвищити точність збору та опрацювання цієї інформації (на практиці – це випадок, коли одного і того ж користувача системи паралельно курують два або більше операторів);

- підтримку офлайн-режиму аналізу відеозаписів;
- можливість повторного перегляду відео у разі спірних моментів;
- мінімальні вимоги до технічного оснащення та інсталяції;

- можливість більш точної ідентифікації оператором моменту початку сценарію в моделі;

- незалежність від формальних сценаріїв: їх зміна не вимагає технічної адаптації інструментів збору, оскільки оператор ідентифікує їх вручну.

Разом з тим, ручний режим має ряд суттєвих обмежень, що знижують його ефективність у довгостроковій перспективі:

- значні фінансові витрати при довготерміновому зборі та опрацюванні статистичної інформації через залученість персоналу;

- людський фактор – як джерело похибок, суб'єктивності та обмежень швидкодії;

- недосяжність рівня функціональних компонентів (сенсорів і актуаторів) для детального аналізу, оскільки оператор не може точно визначити час і послідовність їх спрацювання. Це в свою чергу унеможливорює аналіз та навчання моделі на цьому рівні;

- заповнення вихідної статистичної інформації в ручному режимі збору та опрацювання потребує розроблення додаткового механізму структурування та збереження результуючої статистичної інформації після її збору та опрацювання. Причому, чим більш жорсткі вимоги до часу опрацювання вхідної статистичної інформації, тим, відповідно, складнішими є механізми її структурування та збереження;

- психологічний дискомфорт користувача через наявність відеоспостереження, що може бути критичним фактором неприйняття системи з боку замовника [22].

Отже, враховуючи наведені вище переваги та недоліки ручного режиму збору та опрацювання вхідної статистичної інформації, даний режим найбільш доцільно використовувати при короткотерміновому зборі та опрацюванні вхідної статистичної інформації, або ж на початковому етапі довготермінового збору та опрацювання вхідної інформації для навчання моделі системи РБ.

Окремої уваги заслуговує останній із перелічених вище недоліків ручного режиму – психологічний фактор. Попри те, що він зазначений останнім, на практиці саме він може виявитися вирішальним, Якщо користувач (або замовник), який виступає фінансовим інвестором проєкту, категорично не сприймає відеоспостереження у приватному просторі, жодні технічні чи економічні переваги ручного режиму не зможуть компенсувати цей бар'єр [22]. Тому питання відеофіксації слід обговорювати з замовником на етапі початкового погодження технічного завдання [9].

Завдяки розвитку комп'ютерного зору, машинного навчання та IoT-сенсорики, сучасний ручний режим все частіше поєднується з напівавтоматизованими підходами: відеоаналіз доповнюється AI-системами розпізнавання дій, інтелектуальними камерами з попереднім маркуванням подій, а також аналітикою від сенсорів, синхронізованих із відеоспостереженням. Технології Edge AI дають змогу частково автоматизувати процес, зберігаючи приватність користувача. У цьому контексті роль оператора зміщується від ручного збору даних до аналітичної верифікації, що зменшує вплив людського фактора та підвищує точність.

В *автоматизованому режимі* (рис. 3.5) основними засобами для збору вхідної інформації є сенсори і давачі, а основними засобами опрацювання вхідної інформації з подальшим структуруванням та збереженням результуючої статистичної інформації є автоматизований програмно-апаратний комплекс (АзПАК), який працює під управлінням адміністратора.

На відміну від ручного режиму, де основну роль відіграє людина-оператор, в автоматизованому режимі до 90–95% обробки даних виконується системою автономно. Людина залучається лише для розв'язання вузьких завдань — наприклад, уточнення моменту початку сценарію або перевірки результатів. Основна функція адміністратора полягає в моніторингу процесу функціонування даного АзПАК та можливості його корекції (адміністрування) за необхідності.

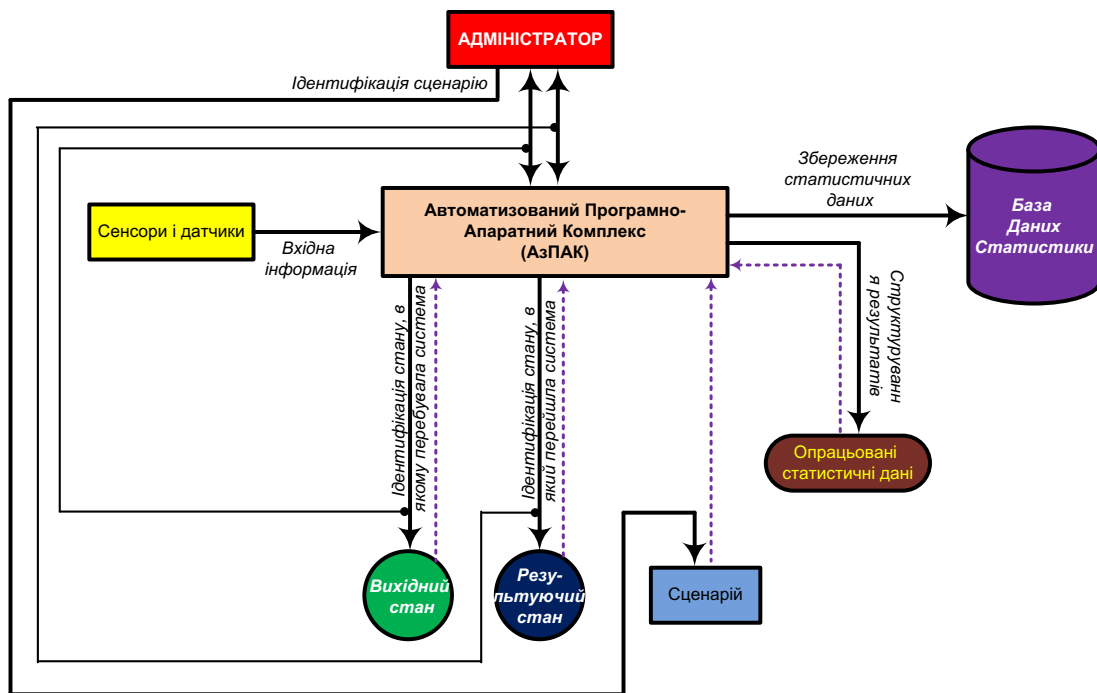


Рис. 3.5. Структурна схема автоматизованого режиму збору та опрацювання вхідної інформації

Переваги автоматизованого режиму збору та опрацювання вхідної статистичної інформації полягають в наступному:

- а) висока продуктивність, швидкодія та можливість опрацювання вхідної інформації в режимі реального часу;
- б) можливість паралельно збирати та опрацьовувати значні об'єми вхідної інформації;
- в) більша економічна вигода при довготерміновому зборі та опрацюванні вхідної статистичної інформації, у порівнянні із ручним режимом;
- г) висока точність збору та опрацювання вхідних даних завдяки мінімізації людського впливу;
- г) адміністратор має можливість ручного керування процесом функціонування АзПАК, та налаштування основних його параметрів у випадку необхідності;
- д) стабільна обробка та збереження великого обсягу структурованої статистики;

е) відсутність відеоспостереження знижує психологічний тиск на користувача;

є) зміна сценаріїв розвитку подій в системі РБ практично не впливає на швидкодію та вимоги щодо технічних засобів опрацювання отриманої вхідної статистичної інформації, оскільки функція ідентифікації сценаріїв покладена виключно на людину-адміністратора АзПАК.

Серед недоліків автоматизованого режиму збору та опрацювання вхідної статистичної інформації можна відзначити наступне:

а) довгий період підготовки, встановлення, налаштування та корекції АзПАК;

б) довгий період модифікації АзПАК при необхідності розширити функціональні можливості комплексу або будь-яким іншим чином модифікувати його;

в) значні фінансові затрати на підготовку, впровадження, налаштування та корекцію АзПАК;

г) присутній незначний вплив людського фактору за рахунок наявності адміністратора АзПАК, який напряду може впливати на роботу комплексу.

Автоматизований режим має низку ключових переваг над ручним, зокрема: вища швидкодія, масштабованість, мінімізація людського фактору та відсутність втручання в приватність. Він особливо ефективний при довготривалому зборі даних, оскільки потребує істотних витрат лише на етапі впровадження. У подальшому система працює стабільно з мінімальним втручанням, що робить її економічно доцільною альтернативою ручному підходу.

В *автоматичному режимі* (див. рис. 3.6), як і у автоматизованому, основними засобами для збору вхідної інформації є сенсори і давачі, інтегровані в систему РБ. Основним засобом опрацювання вхідної інформації з подальшим структуруванням та збереженням результуючої статистичної інформації є автоматичний програмно-апаратний комплекс (АтПАК). Головна

відмінність АтПАК від АзПАК полягає у відсутності необхідності втручання адміністратора комплексу, оскільки комплекс функціонує повністю автономно. Відповідно, в автоматичному режимі з'являється ще одна додаткова перевага – відсутність впливу людського фактору, що призводить до покращення точності, стабільності та швидкодії обробки даних.

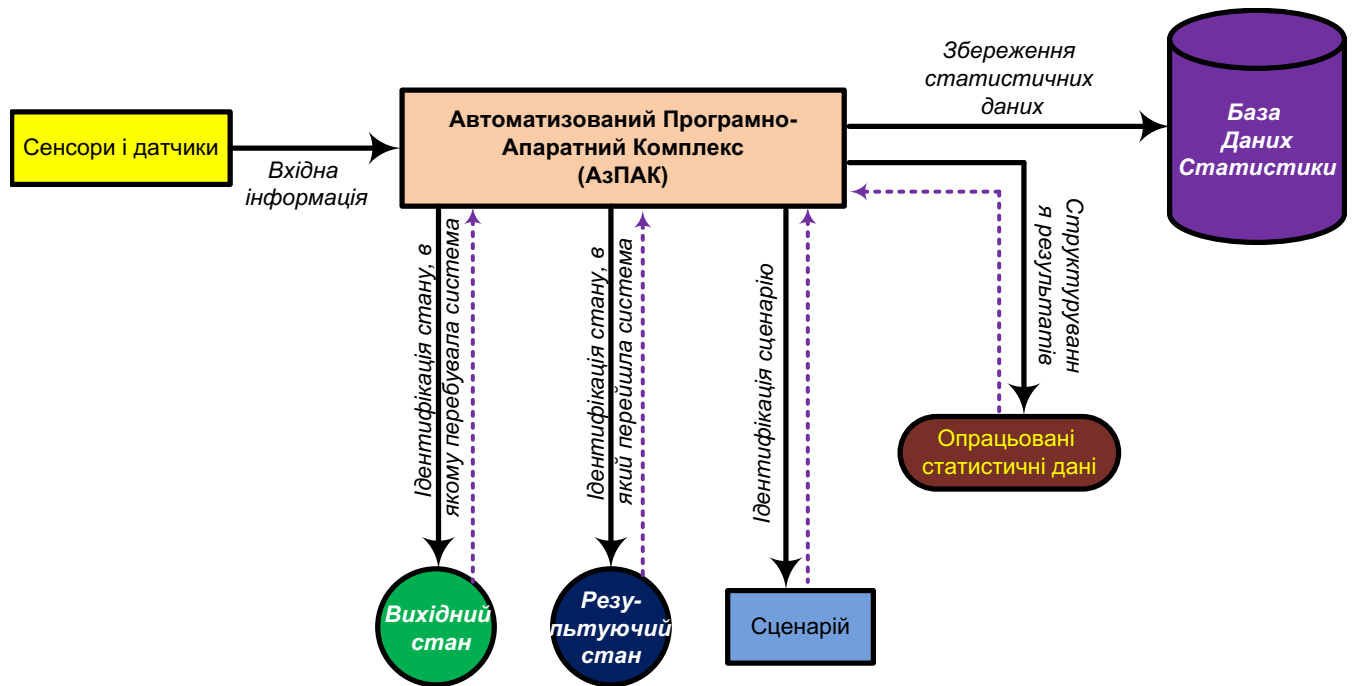


Рис. 3.6 - Структурна схема автоматичного режиму збору та опрацювання вхідної інформації

Та попри отримані переваги, у нас з'являються також і деякі недоліки:

- значний період підготовки, встановлення, налаштування та корекції АтПАК;
- значний період модифікації АтПАК при необхідності розширити функціональні можливості комплексу або будь-яким іншим чином модифікувати його;
- значні фінансові витрати на підготовку, встановлення, налаштування та корекцію АтПАК;
- будь-які зміни в сценаріях розвитку подій в системі РБ призводять до необхідності модифікації АтПАК, оскільки саме він відповідає за їх розпізнавання і опрацювання даних

Основна складність автоматичного режиму збору та опрацювання вхідної статистичної інформації полягає саме в реалізації механізмів ідентифікації сценаріїв розвитку подій системи РБ без залучення людини. Це потребує значних ресурсів на розроблення, налаштування та корекцію АтПАК. Зокрема, йдеться про побудову самостійно функціонуючих моделей, здатних визначати початок, хід і завершення сценаріїв лише на основі сигналів сенсорів, без участі оператора чи адміністратора.

Проте, всі перелічені вище недоліки та особливості даного режиму є незначними. У більшості випадків автоматичний режим є не набагато дорожчим чи складнішим з точки зору технічної реалізації, ніж автоматизований, особливо коли відбувається плавний перехід між автоматизованим режимом в автоматичний. Як альтернативний варіант, спочатку розробляється деякий АтПАК з можливістю ручного адміністрування, а вже після отримання достатньої кількості необхідної інформації, що стосується ідентифікації сценаріїв, всі функції адміністратора покладаються на сам комплекс.

У контексті навчання моделей систем РБ, сформована База Даних Статистики (БДС) (рис. 3.4–3.6) є ключовим фактором. На практиці успішність навчання залежить від повноти, точності та достовірності зібраної статистичної інформації. Саме на основі цієї інформації проводиться заміна початкових задекларованих усталених значень ймовірностей у матриці переходів моделі на реальні статистично обґрунтовані коефіцієнти [9].

Таким чином, якісний збір даних — незалежно від обраного режиму (ручного, автоматизованого чи автоматичного) — є основою для формування надійної, адаптивної моделі системи РБ, побудованої на основі МПМ та доповненої функціональними компонентами.

3.3. Розроблення алгоритму реалізації методу навчання моделей систем РБ, створених на основі МПМ та доповнених функціональними компонентами

На рис. 3.7 представлено блок-схему базового алгоритму навчання моделей систем РБ, створених на основі МПМ та доповнених функціональними компонентами.

Робота алгоритму розпочинається з підключення до файлу, в якому збережена конфігурація моделі: наявні рівні декомпозиції, стани моделі, доступні сценарії, взаємозв'язки між станами, а також відношення типу «батько-син» переходів між декомпозиційними рівнями моделі.

На наступному етапі виконується ініціалізація моделі шляхом обнулення лічильників спрацювання всіх можливих переходів для кожного сценарію. Після цього відбувається підключення до бази даних статистики (БДС) з подальшим покроковим аналізом кожного запису. В ході даного аналізу відбувається збільшення лічильників спрацювання переходів в межах відповідних сценаріїв [9].

Після завершення опрацювання всіх наявних в БДС статистичних даних, отримуємо в параметрі SUM для кожного переходу кожного задекларованого сценарію моделі фактичну кількість спрацювань даного переходу в межах свого сценарію.

Наступним етапом є підрахунок для всіх кожного стану моделі загальної кількості спрацювань усіх вихідних переходів (*GeneralSum*). Після цього розраховуються коефіцієнти ймовірності кожного вихідного переходу з поточного стану за формулою (3.2).

У результаті виконання алгоритму модель системи РБ адаптується до реальних умов функціонування, оскільки переходи між станами моделі вже не є умовними, а засновані на реальній поведінці користувача, зафіксованій в процесі дослідної експлуатації [9].

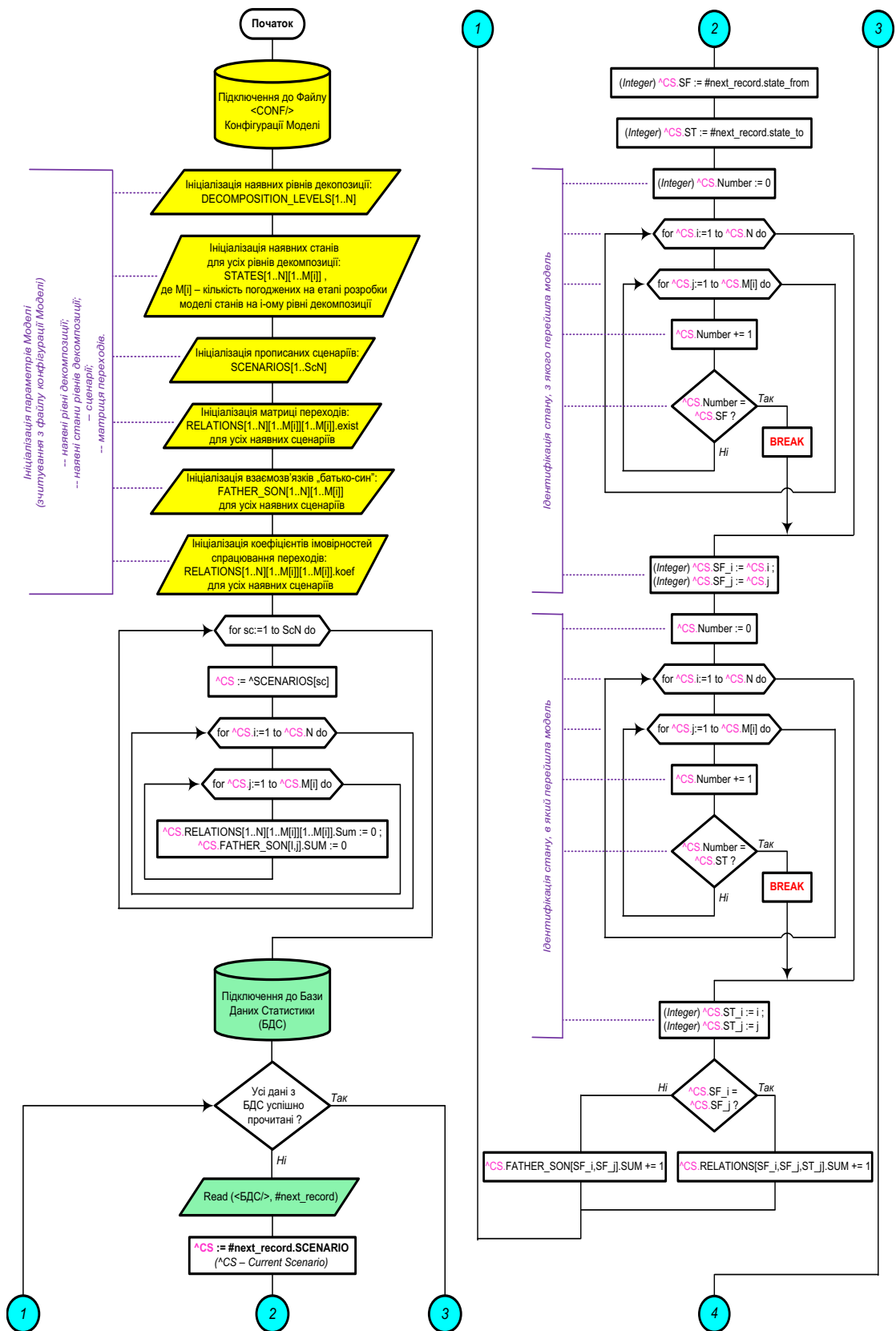
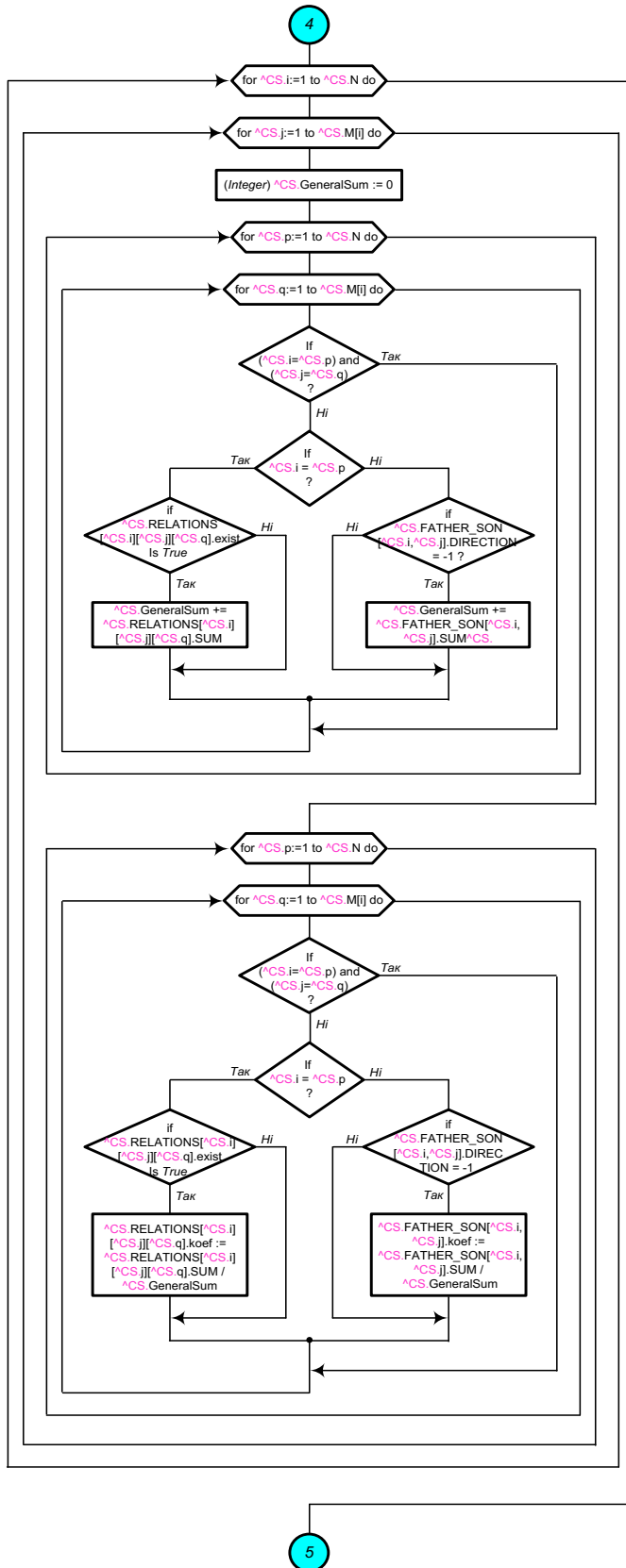


Рис. 3.7 - Блок-схема базового алгоритму навчання моделей систем РБ, створених на основі МПМ та доповнених функціональними компонентами



Продовження рисунку 3.7

3.4. Оптимізація алгоритму навчання моделей систем РБ на основі МПМ та доповнених функціональними компонентами

Основною перевагою розробленого базового алгоритму навчання моделей систем РБ, створених на основі МПМ та доповнених функціональними компонентами є представлення структури даних моделі та логіки процесів в максимально простій та зрозумілій для звичайного користувача формі. Зокрема, структура моделі передбачає чітке розмежування рівнів декомпозиції, визначені переходи між цими рівнями, а сам процес функціонування реалізовано покроково (step-by-step), що полегшує візуальне сприйняття та логічне розуміння.

Проте, для розробника така структура представлення даних моделі є надмірно деталізованою. Вона вимагає використання тривимірних і навіть чотиривимірних масивів, що призводить до підвищених обчислювальних витрат та ускладнення реалізації. У зв'язку з цим виникає необхідність в оптимізації структури подання даних моделі з метою підвищення її компактності та ефективності.

Для досягнення цієї мети розроблено спеціальний алгоритм конвертації структури представлення станів моделі з двовимірної (де стани розділено відповідно до рівнів декомпозиції) в одновимірну лінійну форму, що передбачає послідовне представлення усіх станів у вигляді одномірного масиву (рис. 3.9) [1]. Крім того, реалізовано зворотний алгоритм, який дає змогу виконати реконструкцію лінійного представлення у початкову двовимірну структуру відповідно до ієрархії рівнів декомпозиції моделі (рис. 3.10).

Штрихпунктирною лінією на рис. 3.9 позначено фрагмент блок-схеми, що відповідає етапу ініціалізації двовимірної матриці реляцій (взаємозв'язків) між усіма станами моделі. Цей етап є характерним лише для алгоритму конвертації моделі з двовимірної форми в лінійну (одновимірну), і, відповідно, відсутній у блок-схемі зворотної конвертації, поданої на рис. 3.10.

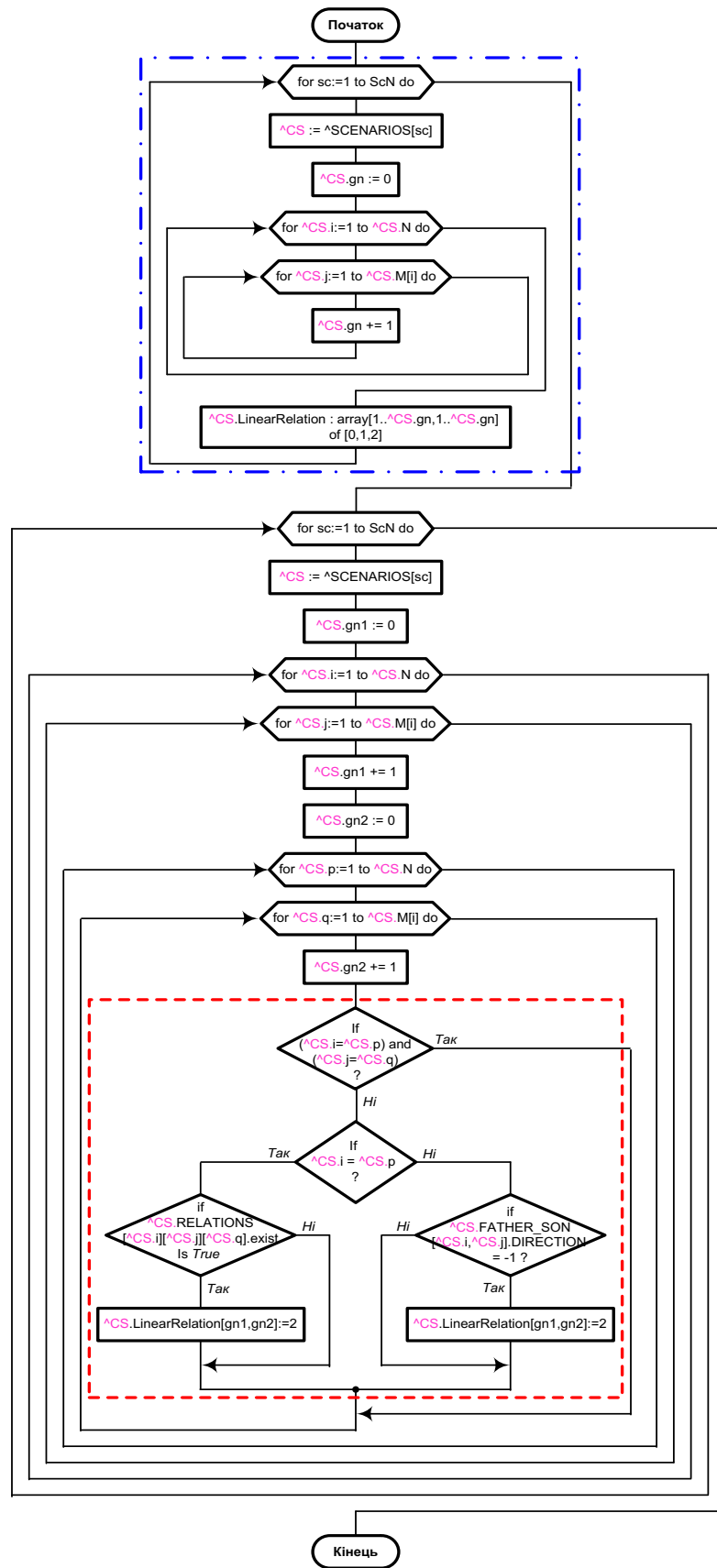


Рис. 3.9 - Блок-схема конвертації форми представлення моделі з двовимірної в лінійну

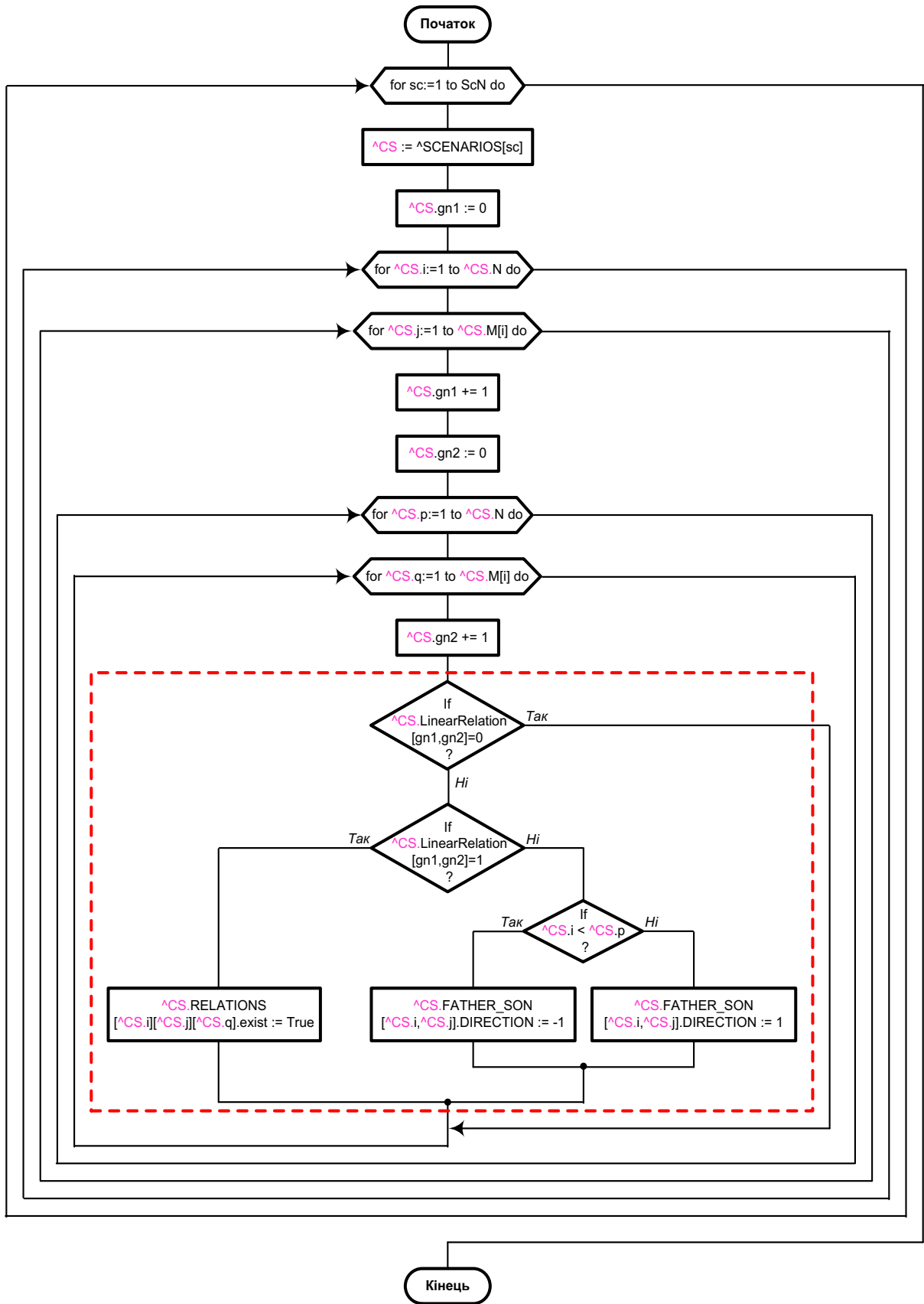


Рис. 3.10 - Фрагмент блок-схеми для конвертації форми представлення моделі з лінійної (одновимірної) в двовимірну

Пунктирною лінією на рис. 3.9 і на рис. 3.10 позначений фрагмент блок-схеми, в якому полягає основна відмінність між алгоритмом конвертації форми представлення моделі з двовимірної в лінійну (рис. 3.9) та алгоритмом зворотної конвертації форми представлення моделі – з лінійної (одновимірної) в двовимірну (рис.3.10).

Таким чином, формалізована математична модель конвертації форми представлення моделі з двовимірної в лінійну (одновимірну), а також навпаки – з лінійної (одновимірної) в двовимірну, матиме наступний вигляд:

Частина 1: пряма конвертація (з двовимірної в лінійну):

$$\begin{bmatrix} d_1^1 & d_1^2 & \dots & d_1^n \\ d_2^1 & d_2^2 & \dots & d_2^n \\ \dots & \dots & \dots & \dots \\ d_m^1 & d_m^2 & \dots & d_m^n \end{bmatrix} \xrightarrow[\substack{\text{for } i:=1 \text{ to } n \text{ do} \\ \text{for } j:=1 \text{ to } m \text{ do} \\ o_{[(i-1)*n+j]}:=d_j^i}]{\text{}} [o_1 \quad o_2 \quad \dots \quad o_{n*m}] \quad (3.16)$$

Частина 2: зворотна конвертація (з лінійної в двовимірну):

$$[o_1 \quad o_2 \quad \dots \quad o_{n*m}] \xrightarrow[\substack{\text{for } i:=1 \text{ to } n*m \text{ do} \\ \text{trunc}\left(\frac{i}{n}\right) \\ d_{i-\text{trunc}\left(\frac{i}{n}\right)*n}:=o_i}]{\text{}} \begin{bmatrix} d_1^1 & d_1^2 & \dots & d_1^n \\ d_2^1 & d_2^2 & \dots & d_2^n \\ \dots & \dots & \dots & \dots \\ d_m^1 & d_m^2 & \dots & d_m^n \end{bmatrix}$$

де: d_j^i - значення елемента у j-му стовпці та i-му рядку двовимірної моделі

m — кількість рядків (наприклад, рівні декомпозиції);

n — кількість стовпців (наприклад, кількість станів на рівні);

o_k — елемент у лінійному одновимірному масиві.

Отже, в результаті здійсненої оптимізації (переходу від двовимірної форми представлення моделі до одновимірної (лінійної) в алгоритмі навчання

моделей систем РБ створених на основі МПМ та доповнених функціональними компонентами, вдалося досягти низки суттєвих переваг:

- спрощення структури даних. Якщо раніше множина станів моделі була представлена двовимірним масивом, множина реляцій (взаємозв'язків) між станами різних рівнів декомпозиції (взаємозв'язки типу «батько-син») були представлені тривимірною матрицею, а множина реляцій (взаємозв'язків) станів одного рівня декомпозиції була представлена чотиривимірною матрицею, то після оптимізації множина всіх станів моделі представлена одновимірним масивом, множина реляцій всіх станів моделі (незалежно від того чи це стани різних рівнів декомпозиції чи одного) представлена двовимірним масивом. Це дає позитивний ефект як в плані раціоналізації об'ємів використання пам'яті, так і в плані збільшення швидкодії;

- уніфікація нумерації станів. Початково кожен рівень декомпозиції мав власну локальну нумерацію. У результаті оптимізації реалізовано наскрізну нумерацію всіх станів моделі незалежно від рівня, але з урахуванням ієрархії. Це відповідає загальним принципам побудови моделей Петрі та моделей Петрі-Маркова. Окрім цього, загальна нумерація всіх станів моделі спрощує роботу оператора (адміністратора) системи в процесі збору та опрацювання вхідних статистичних даних, а також при записі та зчитуванні результируючих статистичних даних із бази даних статистики (БДС) на етапі навчання. Це полегшує процес ідентифікації стану, з якого перейшла система та стану, в який перейшла система в оптимізованому алгоритмі навчання моделей систем РБ (рис. 3.11);

- об'єднання структури реляцій. До оптимізації всі реляції між станами були поділені на реляції між станами різних рівнів декомпозиції та реляції між станами одного рівня декомпозиції та зберігались в двох різних репозиторіях (матрицях). Це ускладнювало подальшу обробку. Оптимізована модель містить усі реляції в єдиному масиві, що спростило логіку навчання та реалізацію алгоритмів.

Хоча оптимізована (лінійна) структур стала, можливо, менш інформативною для кінцевих користувачів, проте набагато зручнішою для інженерів, адміністраторів та модераторів на стадії впровадження, навчання та експлуатації системи. Натомість на ранніх етапах (наприклад, узгодження з замовником) доцільно використовувати вихідне, двовимірне представлення, оскільки воно є більш наочним і гнучким.

Таким чином, реалізувавши конвертацію форми представлення моделі з двовимірної в лінійну (одновимірну), ми можемо використати отримані результати та переваги лінійної форми представлення моделі для розробки оптимізованого алгоритму навчання моделей систем РБ, створених на основі МПМ та доповнених функціональними компонентами. Блок-схема оптимізованого алгоритму представлена нижче, на рис. 3.11. Саме даний (оптимізований) алгоритм закладений в основу розроблюваного методу навчання моделей систем РБ, створених на основі МПМ та доповнених функціональними компонентами.

Робота оптимізованого алгоритму розпочинається з підключення до файлу конфігурації моделі та зчитування всіх даних із цього файлу, після чого відбувається конвертація форми представлення моделі з двовимірної в лінійну.

Наступним етапом роботи розробленого оптимізованого алгоритму є обнулення лічильників спрацювань всіх переходів для всіх сценаріїв моделі, після чого відбувається підключення до файлу бази даних статистики з подальшим заповненням інформації по кількості спрацювань переходів моделі на основі даних з БДС. Після цього здійснюється перебір всіх станів моделі та підрахунок: а) кількості спрацювань кожного переходу, вихідним станом якого є поточний стан моделі; б) загальної кількості спрацювань усіх переходів, вихідним станом яких є поточний стан моделі. І на основі отриманих даних по кількості спрацювань кожного окремого переходу та загальної кількості спрацювань усіх переходів (для поточного стану моделі) проводиться розрахунок ймовірності спрацювання кожного із переходів.

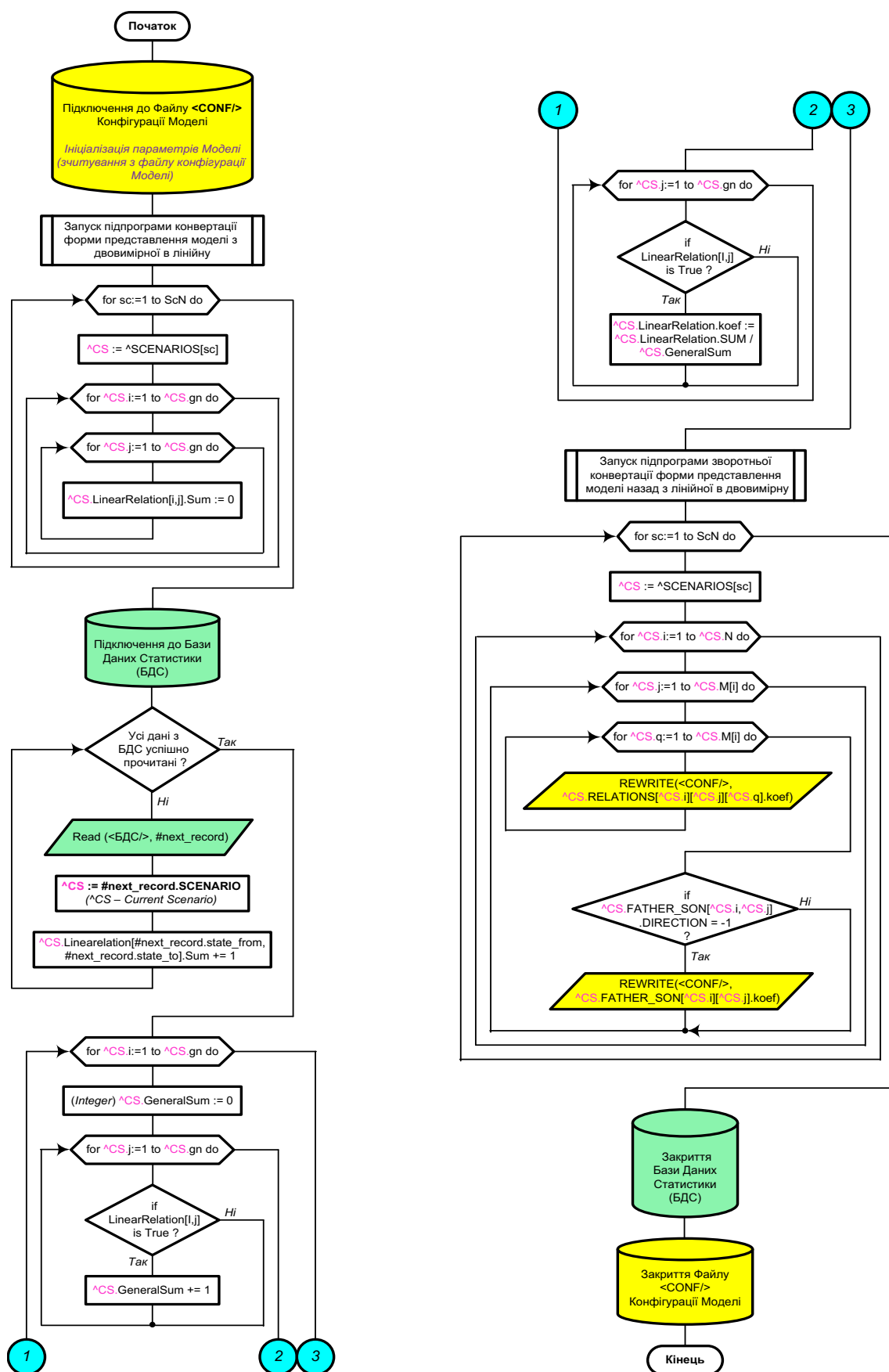


Рис. 3.11 - Блок-схема оптимізованого алгоритму навчання моделей систем РБ

Далі, згідно розробленого алгоритму, відбувається зворотня конвертація форми представлення моделі з лінійної в двовимірну, з подальшим перезаписом отриманих в результаті проведених розрахунків даних по коефіцієнтах ймовірностей спрацювання усіх переходів моделі у файлі конфігурації моделі, закриття БДС, файлу конфігурації моделі та завершення роботи алгоритму.

Висновки до розділу 3

Розроблено метод синтезу моделей систем «Розумний будинок» на основі мереж Петрі-Маркова (МПМ), доповнених функціональними компонентами, який забезпечує багаторівневу декомпозицію структури системи від узагальненого до рівня сенсорів та актуаторів. Це дає змогу описувати широкий спектр сценаріїв функціонування, а також враховувати випадкові процеси у взаємодії користувача з системою, при цьому забезпечуючи можливість верифікації ступеня детермінованості початкових та кінцевих станів кожного сценарію.

Розроблено метод навчання моделей систем РБ, який охоплює два ключові етапи: збір та опрацювання статистичної інформації й подальшу адаптацію моделі. Представлено три режими збору даних – ручний, автоматизований та автоматичний – кожен з яких проаналізовано за перевагами та обмеженнями з урахуванням практичної доцільності.

Розроблено алгоритм реалізації навчання моделі управління адаптивною системою розумного будинку, що ґрунтується на мережах Петрі-Маркова і включає функціональні елементи. Він забезпечує заміну ймовірностей переходів між станами, заданих за замовчуванням, на статистично обґрунтовані значення на основі даних, отриманих під час функціонування системи. У структурі моделі враховано поведінку користувача, динаміку сценаріїв та випадковість подій.

Розроблено механізм побудови узагальненої моделі системи РБ з подальшою генерацією множини похідних моделей окремих сценаріїв.

Реалізовано алгоритм побудови моделі із врахуванням ієрархії декомпозиції, зв'язків між станами, початкових та кінцевих умов, що погоджуються з користувачем.

Оптимізовано структуру подання моделей, що дозволило перейти від громіздких тривимірних і чотирирівимірних масивів до більш компактного одновимірного подання з наскрізною нумерацією станів та єдиним репозиторієм реляцій.

РОЗДІЛ 4 РОЗРОБЛЕННЯ ЗАСОБІВ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УПРАВЛІННЯ АДАПТИВНОЮ СИСТЕМОЮ «РОЗУМНИЙ БУДИНОК» НА ОСНОВІ МОДЕЛЕЙ ПЕТРІ- МАРКОВА І ДОПОВНЕНИХ ФУНКЦІОНАЛЬНИМИ КОМПОНЕНТАМИ ТА РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Розроблення структури інформаційної технології управління адаптивною системою «Розумний будинок» на основі моделей Петрі-Маркова та доповнених функціональними компонентами

В роботі розроблено інформаційну технологію (ІТ) управління адаптивною системою РБ, яка базується на моделях Петрі-Маркова та доповнена функціональними компонентами. Структура цієї технології представлена на рис. 4.1.

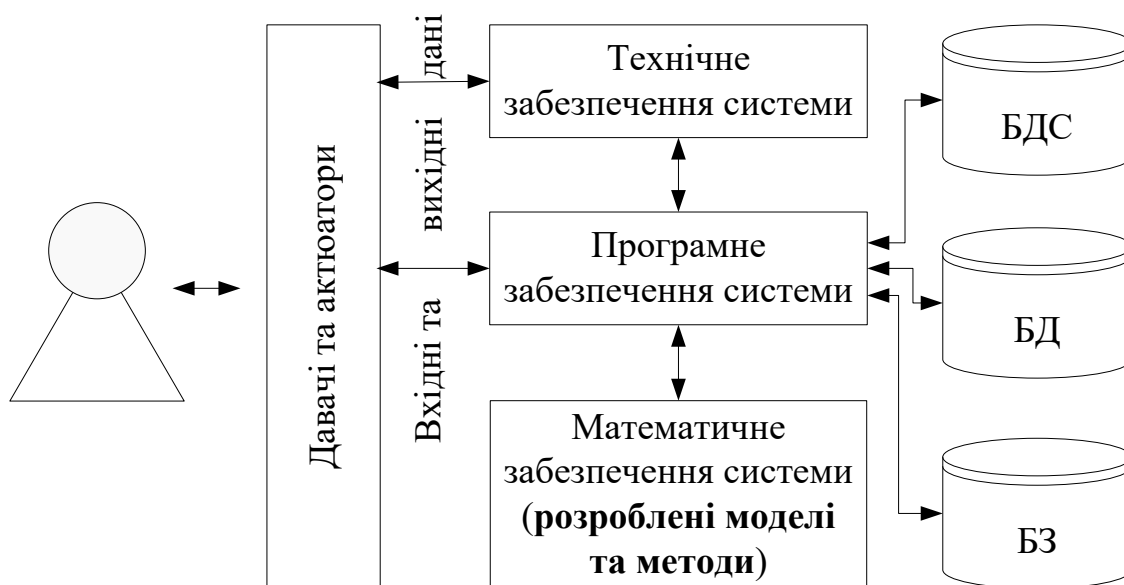


Рис. 4.1 - Структура ІТ управління адаптивною системою РБ на основі МПМ та доповнених функціональними компонентами.

Вхідні дані ІТ отримує від дачів. Ці дані вона опрацьовує з допомогою математичних методів і моделей розроблених у другому та третьому розділах даного дисертаційного дослідження з використанням програмних і технічних

засобів. Отримані вихідні дані передаються на актуатори, що забезпечують включення/виключення виконуючих пристроїв.

Математичне забезпечення системи управління РБ, що використовується для роботи ІТ, включає моделі на основі мереж Петрі-Маркова, метод синтезу моделей на основі мереж Петрі-Маркова та метод їх навчання [1].

4.2 Загальна структура автоматизованого програмно-апаратного комплексу

Загальна структура розроблюваного програмно-апаратного комплексу зображена на рис. 4.2.

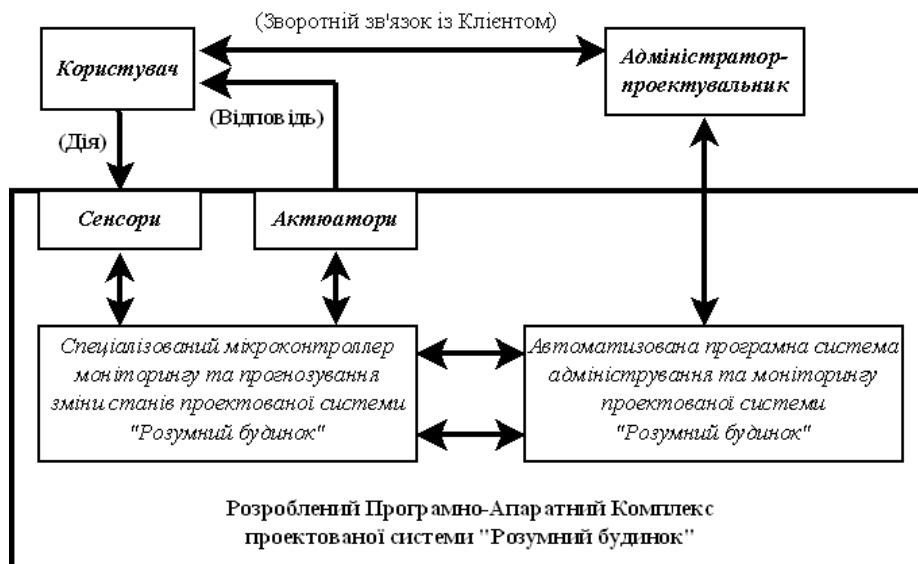


Рис. 4.2 - Загальна структура розроблюваного програмно-апаратного комплексу

Основним завданням розроблюваного спеціалізованого мікроконтролера є моніторинг та прогнозування зміни станів проектованої системи РБ [3]. Для цього мікроконтролер здійснює неперервне потактове опитування всіх доступних сенсорів, що оточують користувача в місці його поточної дислокації та контролює інші параметри роботи системи, такі як дату, час, попередні виконані дії [7].

Отримавши дані від сенсорів, мікроконтролер, на основі прописаних в моделі значень коефіцієнтів ймовірностей досягнення станів, проводить розрахунок ймовірностей досягнення наступних станів, що є суміжними з станом в якому система перебуває в даний час і який встановився завдяки нещодавно активованим актуаторам.

Таким чином, відбувається прогнозування зміни станів моделі. Якщо прогнозоване значення ймовірності досягнення стану моделі, що відповідає конкретному актуатору системи, перевищує порогове значення, відбувається прогнозований перехід моделі в наступні стани (згідно поточного сценарію) [7]. Цей перехід реалізується шляхом формування мікроконтролером сигналу керування, передавання його на вхід відповідного актуатора та його спрацювання.

Автоматизована програмна система адміністрування та моніторингу проєктованої системи РБ має прямий зв'язок зі спеціалізованим мікроконтролером. Вона може як приймати інформацію про поточний стан сенсорів і моделі від мікроконтролера, так і надсилати керуючі сигнали, які впливають на вибір подальших дій системи [3].

Наступним етапом розробки програмно-апаратного комплексу є створення UML-діаграм, які дають змогу наочно описати ключові сценарії та внутрішню логіку системи. Зокрема, розроблено такі UML-діаграми: діаграма прецедентів (рис. 4.3); діаграма станів автомата (рис. 4.4); діаграма послідовності (рис. 4.5) та діаграма діяльності (рис. 4.6).

В розробленій UML-діаграмі прецедентів (рис. 4.3) акторами виступають: «Користувач», «Сенсори», «Актуатори» і «Адміністратор». Прецедентами - «Дія користувача», «Спрацювання сенсора», «Запуск актуатора», «Збір статистики та навчання системи», «Побажання користувача», «Конфігурація системи».

В UML-діаграмі станів автомата (рис. 4.4) в формі теорії автоматів представлений процес функціонування розроблюваного автоматизованого програмно-апаратного комплексу [7].

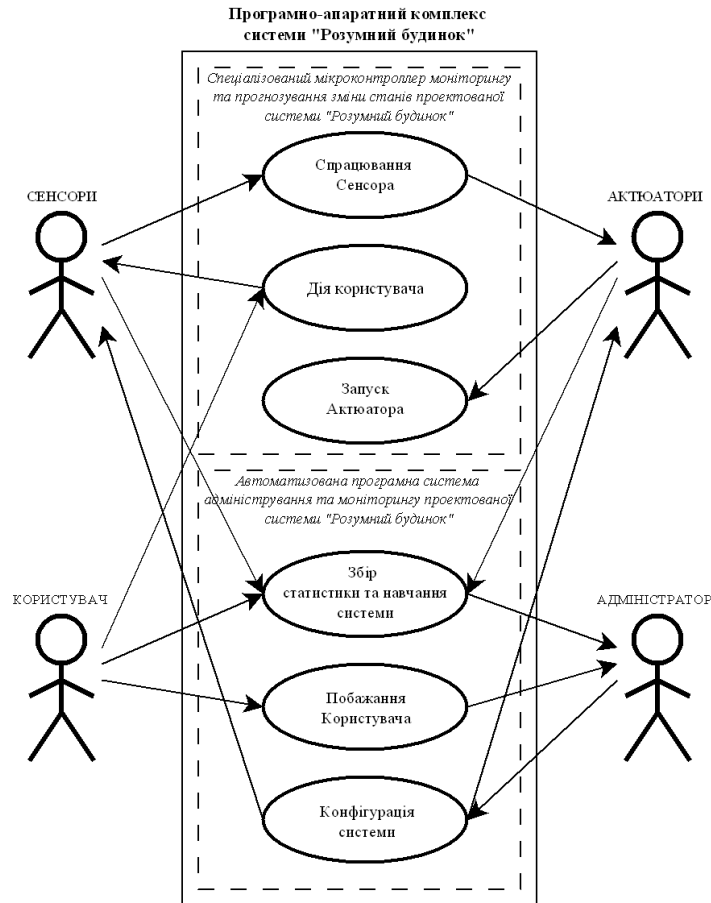


Рис. 4.3 - UML-діаграма прецедентів розроблюваного автоматизованого програмно-апаратного комплексу

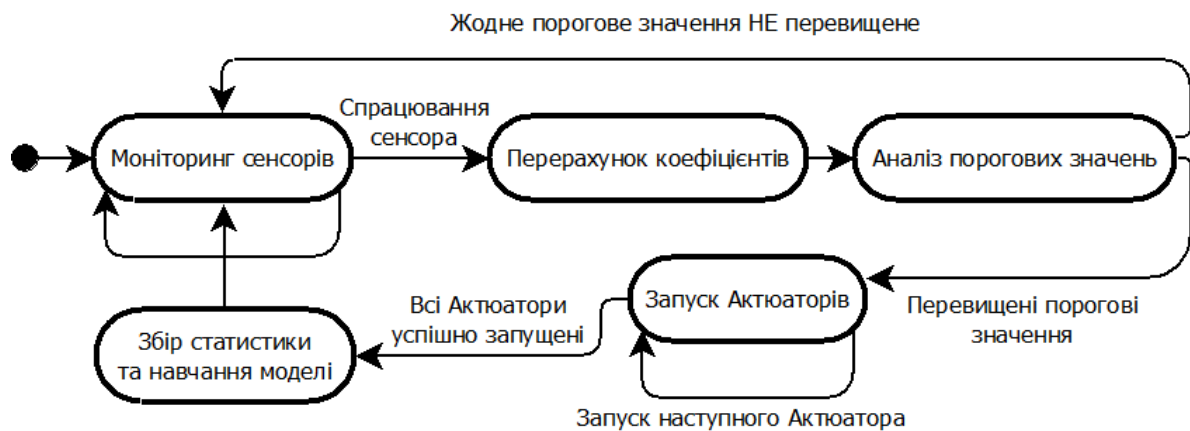


Рис. 4.4 - UML-діаграма станів автомата розроблюваного автоматизованого програмно-апаратного комплексу

Розроблена діаграма послідовності (рис. 4.5) представляє потактову інтерпретацію загального алгоритму функціонування розроблюваного автоматизованого програмно-апаратного комплексу.

UML-діаграма діяльності (рис. 4.6) відображає логічну послідовність виконання операцій під час керування РБ.

Отже, кожна із розроблених і представлених UML-діаграм описує функціонування розроблюваного автоматизованого програмно-апаратного комплексу з певного ракурсу.

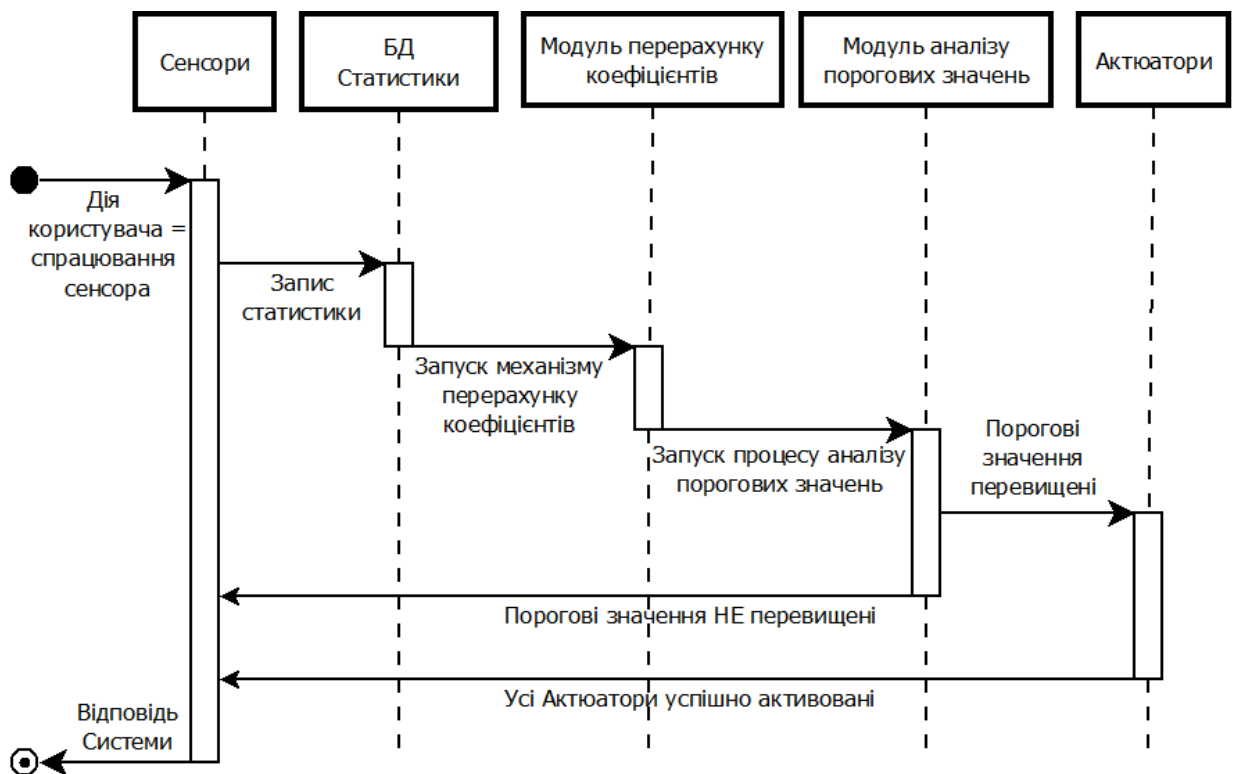


Рис. 4.5 - UML-діаграма послідовності розроблюваного автоматизованого програмно-апаратного комплексу

В сукупності всі розроблені та представлені UML-діаграми дають змогу здійснити комплексне представлення функціонування розроблюваного автоматизованого програмно-апаратного комплексу.

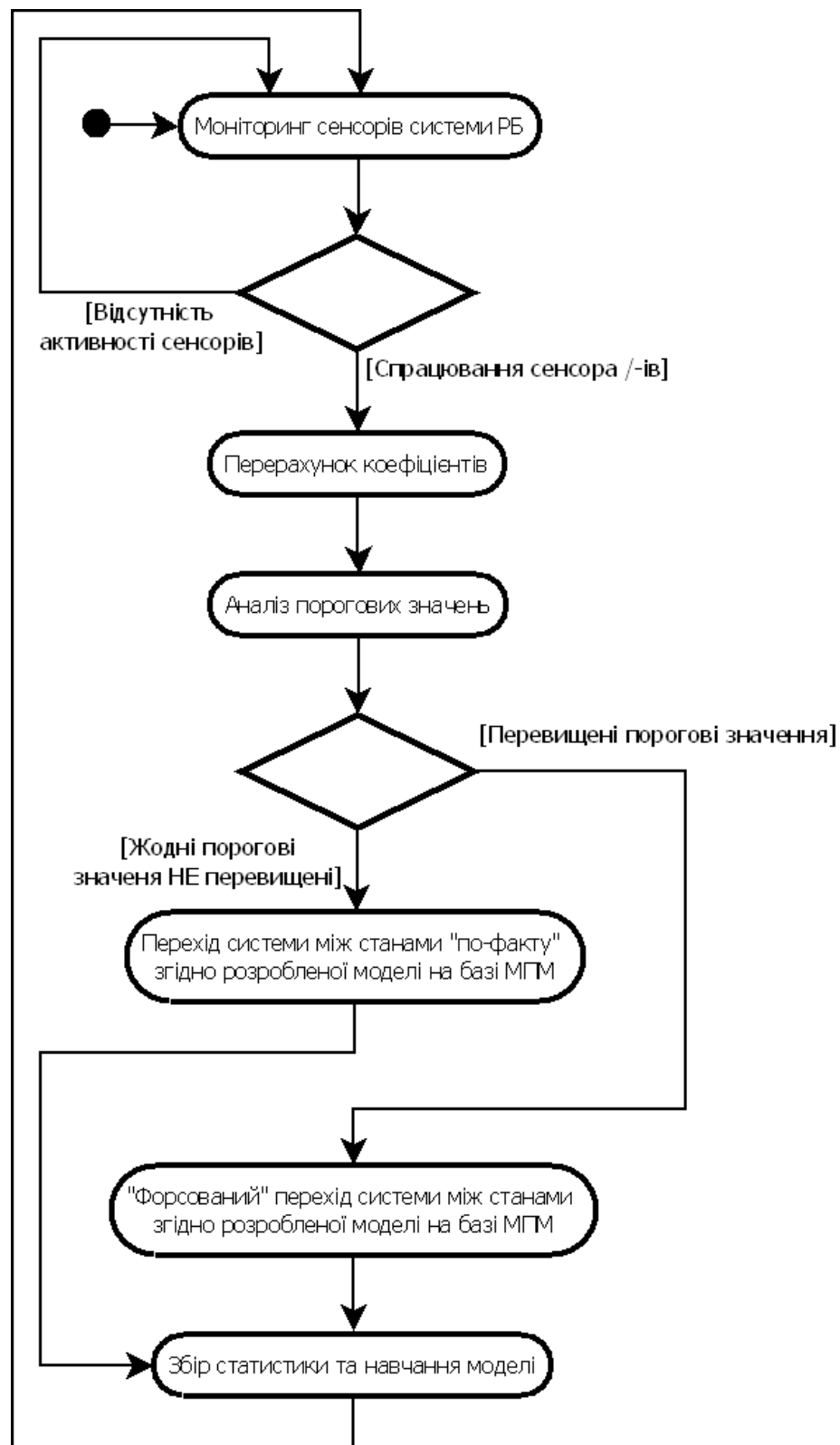


Рис. 4.6 - UML-діаграма діяльності розроблюваного автоматизованого програмно-апаратного комплексу

4.3 Алгоритм функціонування та структура спеціалізованого мікроконтролера

Блок-схема розробленого алгоритму моніторингу та імовірнісного прогнозування зміни станів систем РБ представлена на рис. 4.7.

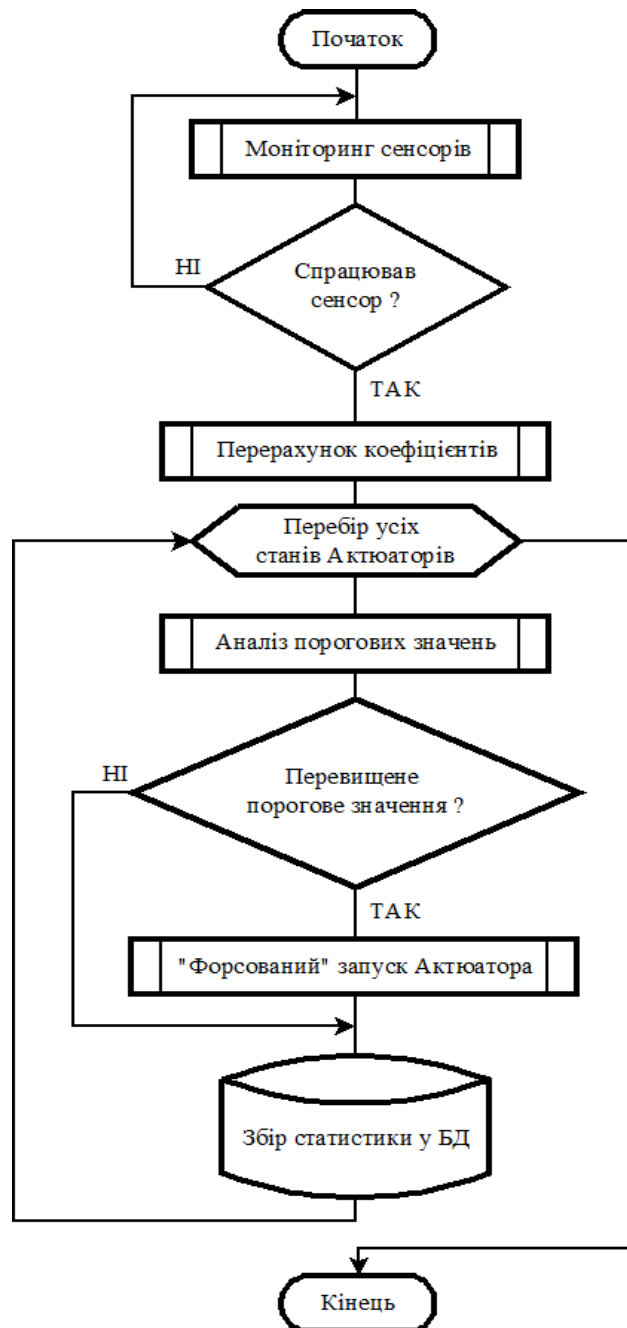


Рис. 4.7 - Алгоритм моніторингу та імовірнісного прогнозування зміни станів РБ

Робота алгоритму розпочинається з моніторингу активності сенсорів проєктованої системи РБ. У випадку спрацювання будь-якого сенсора виконується перерахунок коефіцієнтів переходів моделі, а також обхід дерева моделі зверху-вниз від стану, що відповідає сенсору, який спрацював, аж до кінцевих станів усіх вихідних віток дерева моделі. Під час цього здійснюється поетапний розрахунок сумарної ймовірності досягнення кінцевих станів, які представляють актуатори проєктованої системи РБ.

В результаті, для кожного кінцевого стану усіх вихідних віток дерева, що беруть свій початок від сенсора, який спрацював, ми отримаємо сумарну ймовірність досягнення цього стану, і у випадку, якщо це значення досягає порогового, відбувається форсований запуск відповідного актуатора та форсований перехід системи в наступні стани згідно розробленої моделі.

У випадку, якщо сумарна ймовірність не досягає порогового значення для цього стану, відбувається подальший аналіз для стану наступного актуатора. Це здійснюється поки не будуть розглянуті всі кінцеві стани усіх вихідних віток дерева, що беруть свій початок від сенсора, який спрацював.

Після перевірки всіх станів та відповідних порогових значень результати моніторингу та обчислень записуються до бази даних статистики. Така інформація використовується для подальшого навчання та вдосконалення моделі РБ.

Крім того, доцільно здійснювати не загальний моніторинг всієї системи, а попередньо поділити модель РБ на кілька структурно-функціональних підсистем. Це дає змогу виконувати паралельний моніторинг кожної підсистеми та оперативніше реагувати на зміни у відповідних частинах комплексу.

На рис. 4.8 зображена розроблена структура спеціалізованого мікроконтролера моніторингу та ймовірнісного прогнозування зміни станів проєктованої системи РБ [3].

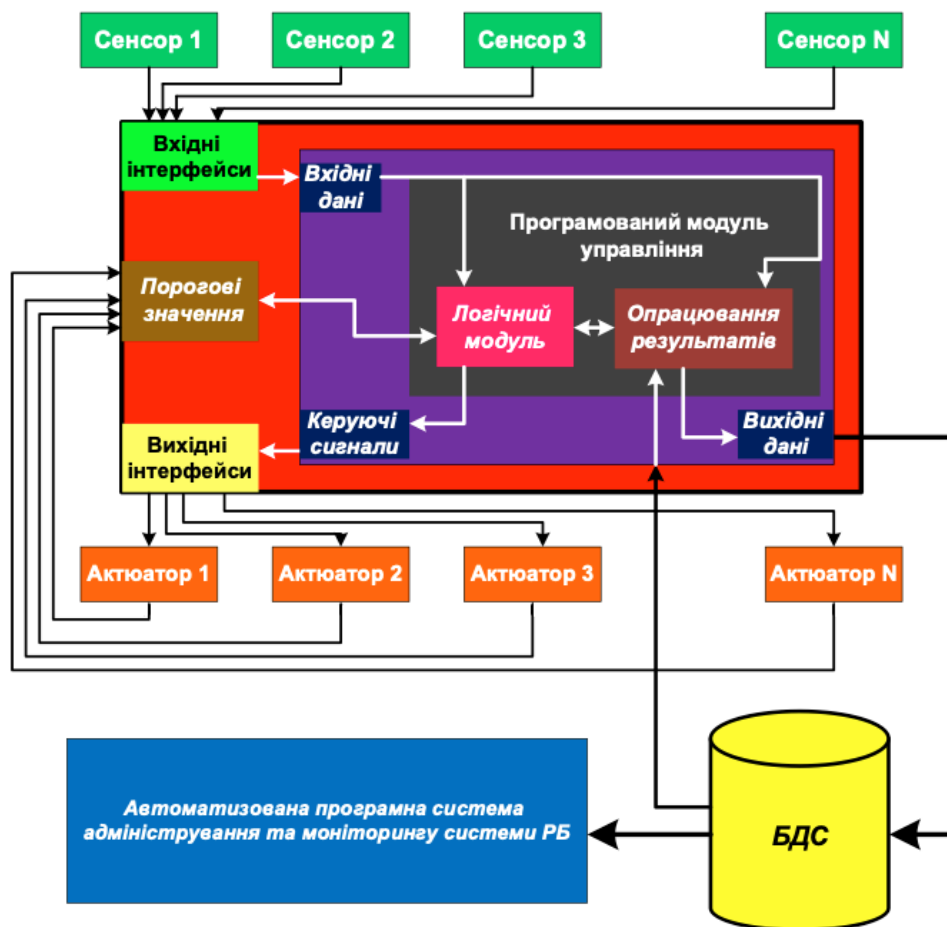


Рис. 4.8 - Структура спеціалізованого мікроконтролера моніторингу та імовірнісного прогнозування зміни станів проєктованої системи РБ

Усі зібрані та оброблені дані мікроконтролер зберігає у відповідній базі даних статистики (БДС) [3]. Оскільки переконфігурація мікроконтролера для різних сценаріїв і завдань дослідження систем РБ є досить складним, ресурсомістким та тривалим процесом, що потребує додаткових фінансових і часових затрат, було прийнято рішення розробити додаткове програмне забезпечення. Це програмне забезпечення моделює роботу мікроконтролера та наповнює БДС. Таким чином реалізовується оптимізація ресурсів для здійснення досліджень з моделювання різноманітних сценаріїв функціонування проєктованих систем.

4.4 Алгоритм імовірнісного прогнозування зміни станів для адаптивної системи РБ

На рис. 4.9 наведено блок-схему алгоритму імовірнісного прогнозування зміни станів для адаптивної системи РБ. Розглянемо основні кроки алгоритму на прикладі.

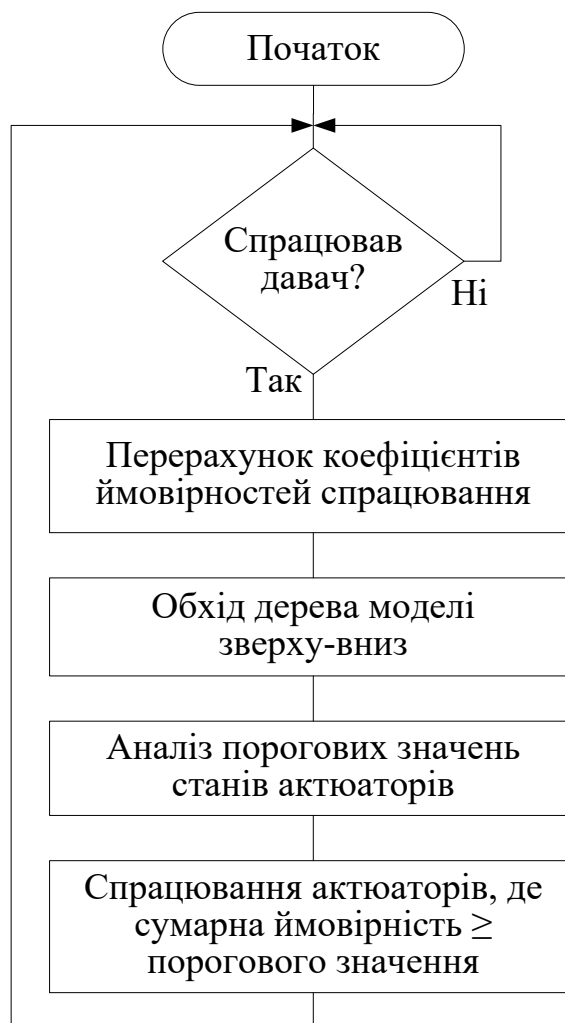


Рис. 4.9 - Блок-схема алгоритму імовірнісного прогнозування зміни станів для адаптивної системи РБ

На рис. 4.10 представлений фрагмент моделі, побудованої на основі МПМ та доповненої функціональними компонентами, для системи РБ. Припустімо, що в ході моніторингу моделі проєктованої системи РБ спрацював «Сенсор2», що знаходиться у цьому фрагменті моделі. Згідно розробленого алгоритму

відбувається перерахунок коефіцієнтів ймовірностей спрацювання переходів моделі. Отримуємо маркер в позиції «Р3» і, відповідно, в нас спрацює перехід «t2». Це автоматично збільшить ймовірність спрацювання саме цього переходу.

Наступним етапом роботи алгоритму є обхід дерева моделі зверху вниз від позиції сенсора, який спрацював, тобто, в розглядуваному нами прикладі – від позиції «Р3» вниз по дереву моделі аж до досягнення тупикових станів, або виходу за межі фрагменту, в рамках якого здійснюється моніторинг. Це зумовлено попереднім розбиттям моделі всієї проєктованої системи РБ на окремі структурно-функціональні частини, і реалізацією відповідного локального моніторингу.

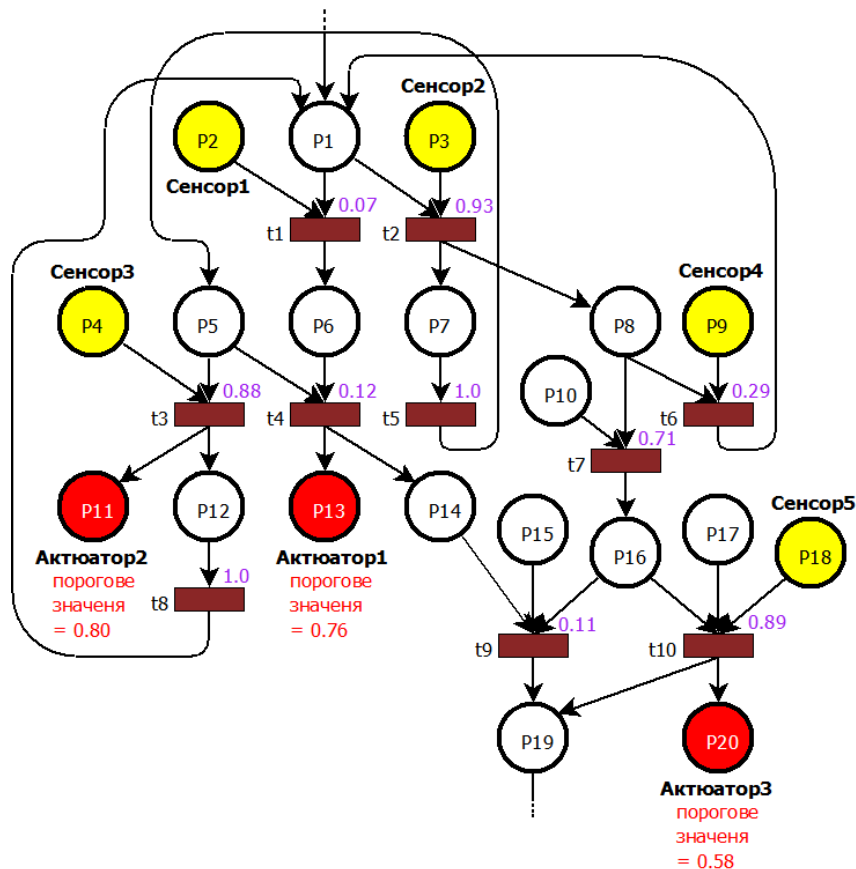


Рис. 4.10 - Фрагмент моделі проєктованої системи РБ, на основі МПМ та доповненої функціональними компонентами

На рис. 4.11 представлені можливі шляхи обходу дерева моделі зверху-вниз від стану «Р3» активованого сенсора «Сенсор2». Таким чином, кожна окрема вітка можливого варіанту обходу дерева моделі завершується або при

досягненні тупикового стану, або при виході за межі області локального моніторингу, або ж при поверненні у вихідний стан.

Наступним етапом роботи розробленого алгоритму моніторингу та імовірнісного прогнозування зміни станів РБ є аналіз порогових значень станів актуаторів, задіяних в ході обходу дерева моделі.

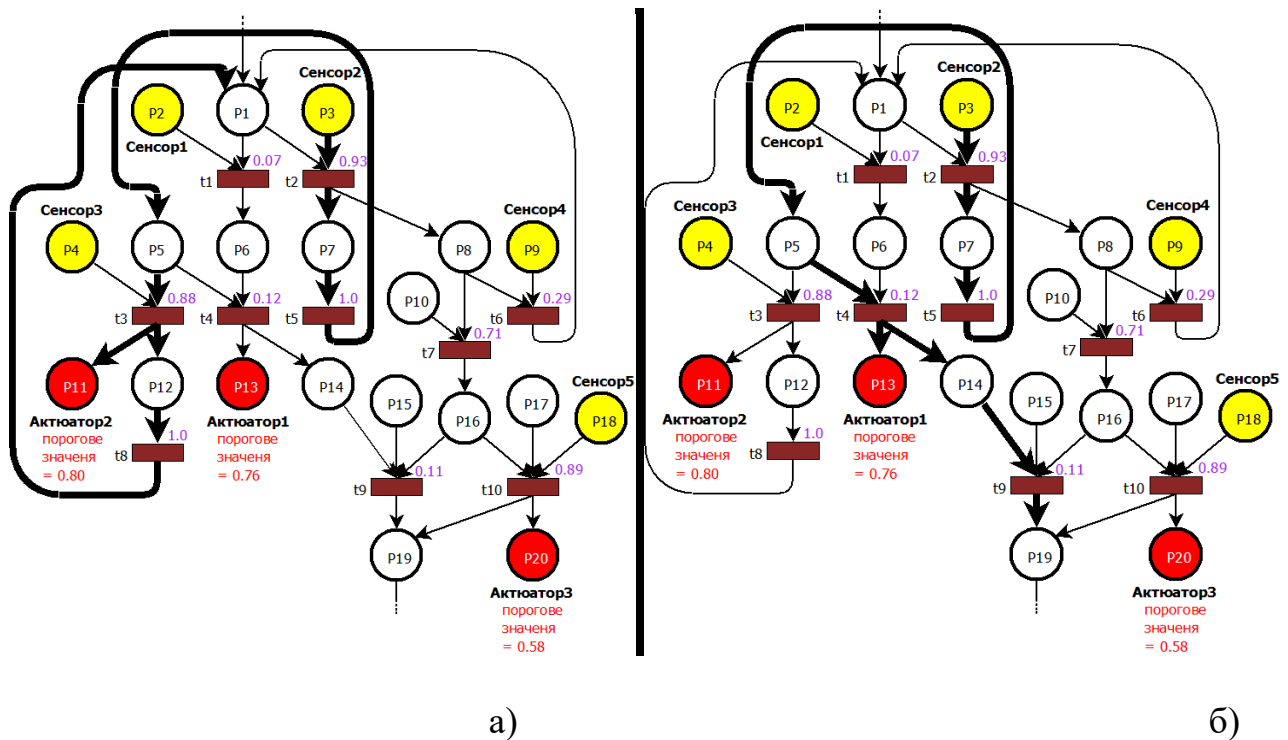
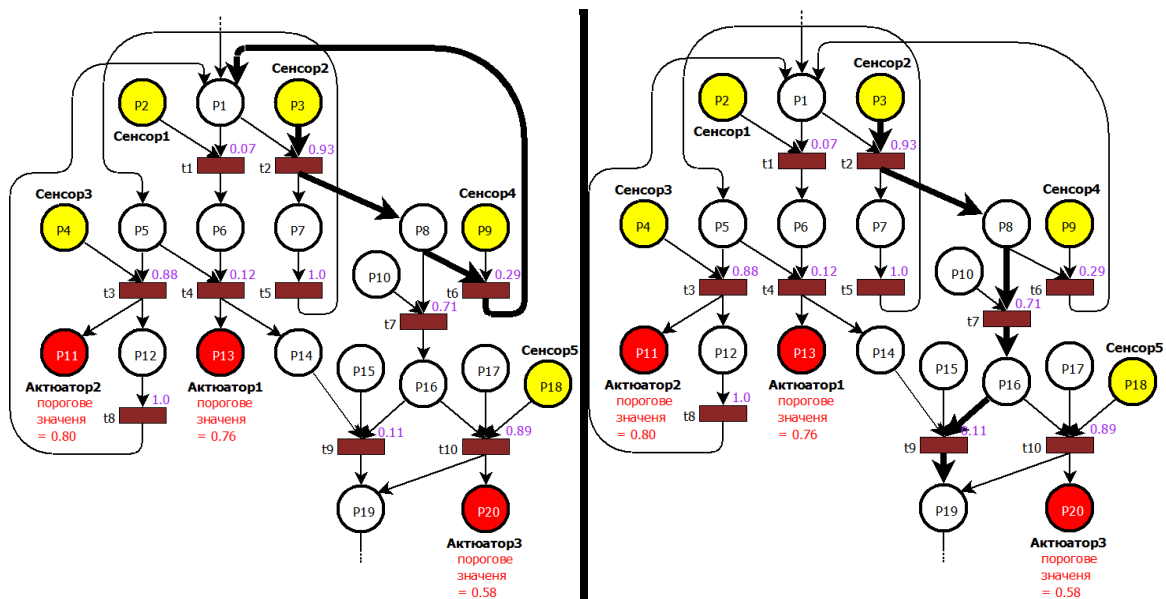


Рис. 4.11 - Можливі шляхи обходу дерева моделі зверху-вниз від стану «P3» активованого сенсора «Сенсор2»

Розглянемо детальніше для кожного із можливих шляхів обходу:

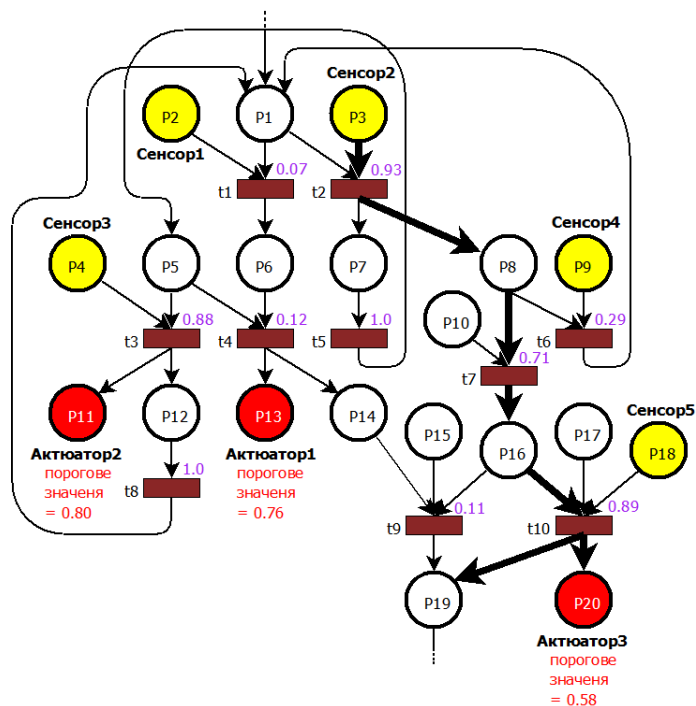
– варіант обходу (рис. 4.11, а) – задіяний «Актuator2» (стан «P11» моделі), порогове значення спрацювання якого = 0.80, в той час як сумарна імовірність досягнення стану «P11» актуатора «Актuator2» = $0.93 \cdot 1 \cdot 0.88 = 0.8184$, тобто, сумарна імовірність перевищує порогове значення, і система здійснює прискорений перехід до стану «P11» моделі, що призводить до автоматичного запуску актуатора «Актuator3», який при звичайних умовах повинен би був спрацювати лише при активації сенсора «Сенсор3», проте

завдяки розробленим принципам проєктована система РБ здійснює випереджаючу активацію;



В)

Г)



Д)

Рис. 4.11 - Возможные шляхи обходу дерева моделі зверху-вниз від стану «P3» активованого сенсора «Сенсор2» (Продовження).

– варіант обходу (рис. 4.11, б) – задіяний «Актuator1» (стан «P13» моделі), порогове значення спрацювання якого $= 0.76$, в той час як сумарна імовірність досягнення стану «P13» актуатора «Актuator1» $= 0.93 * 1 * 0.12 = 0.1116$, тобто, сумарна імовірність менша від порогового значення і жодного «форсованого» переходу до стану «P13» актуатора «Актuator1» згідно алгоритму реалізовано не буде. Модель продовжує функціонувати в звичайному режимі згідно наявних в позиціях маркерів;

– варіант обходу (рис. 4.11, в) – можливий лише при спрацюванні сенсора «Сенсор4». Жодного задіяного актуатора в даному варіант обходу немає. Модель продовжує функціонувати в звичайному режимі згідно наявних в позиціях маркерів;

– варіант обходу (рис. 4.11, г) – жодного задіяного актуатора в даному варіант обходу немає. Модель продовжує функціонувати в звичайному режимі згідно наявних в позиціях маркерів;

– варіант обходу (рис. 4.11, д) – задіяний «Актuator3» (стан «P20» моделі), порогове значення спрацювання якого $= 0.58$, в той час як сумарна імовірність досягнення стану «P20» актуатора «Актuator3» $= 0.93 * 0.71 * 0.89 = 0.587667$. Таким чином, сумарна імовірність перевищує порогове значення і система здійснює прискорений перехід до стану «P20» моделі. Це призводить до автоматичного запуску актуатора «Актuator3», який при звичайних умовах повинен би був спрацювати лише при активації сенсора «Сенсор5». В даному випадку також має місце випереджаючої активації.

4.5 Особливості розроблення технічного забезпечення адаптивної системи РБ

4.5.1 Загальна структура технічного забезпечення адаптивної системи РБ

Розроблена адаптивна структура системи РБ включає чотири основні складові (рис. 4.12) [3].

Першою складовою є контролер (універсальний контролер на основі STM8), який призначений для збору та попереднього опрацювання даних про власника приміщення чи групи мешканців. Сюди відноситься ідентифікація особи, місце її знаходження, час, ідентифікація дій власника тощо.

Другою складовою є база даних (БД) та знань (БЗ) БДС, які зберігають інформацію, отриману від контролера, а також дані, зібрані під час функціонування системи РБ. Варто зазначити, що знання формуються на основі аналізу та обробки даних контролера.

Третьою складовою є модуль опрацювання вхідних даних. Він обробляє інформацію, що надходить як із системи зберігання даних, так і з бази знань. Він використовує методи та моделі, засновані на теорії Петрі-Маркова [93], для формування сигналів управління, які передаються до підсистеми виконуючих пристроїв (актуаторів).

Четвертою складовою є безпосередньо функціональні пристрої (давачі та актуатори). Запропонована структура адаптивної системи РБ використовує клієнт-серверну організацію. Система містить у складі контролер адаптивної системи РБ (КАСРБ).

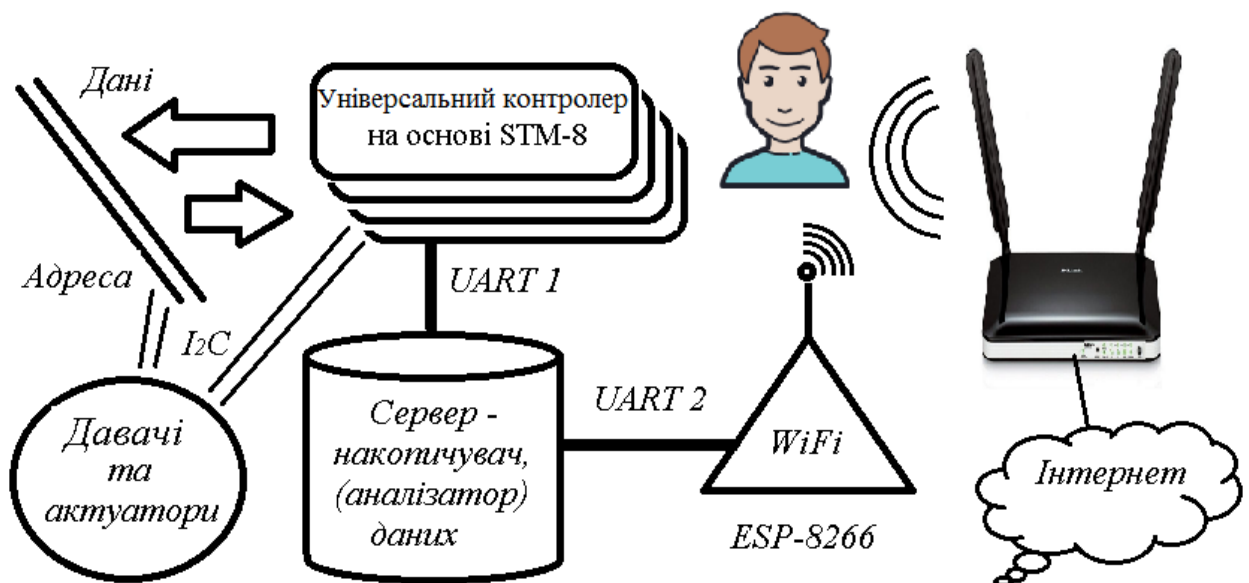


Рис. 4.12 - Загальна структура адаптивної системи РБ

4.5.2 Особливості реалізації технічного забезпечення контролера

Структурна схема контролера адаптивної системи РБ (КАСРБ) зображена на рис. 4.13. Збір інформації від датчиків та передача даних здійснюється на основі мікроконтролерів STM8 [3], [96], [97]. Комунікацію даних між контролерами реалізовано з використанням протоколу I²C [98], а зв'язок системи з мережею Internet здійснюється за допомогою SoC-мікроконтролера ESP8266 [99].

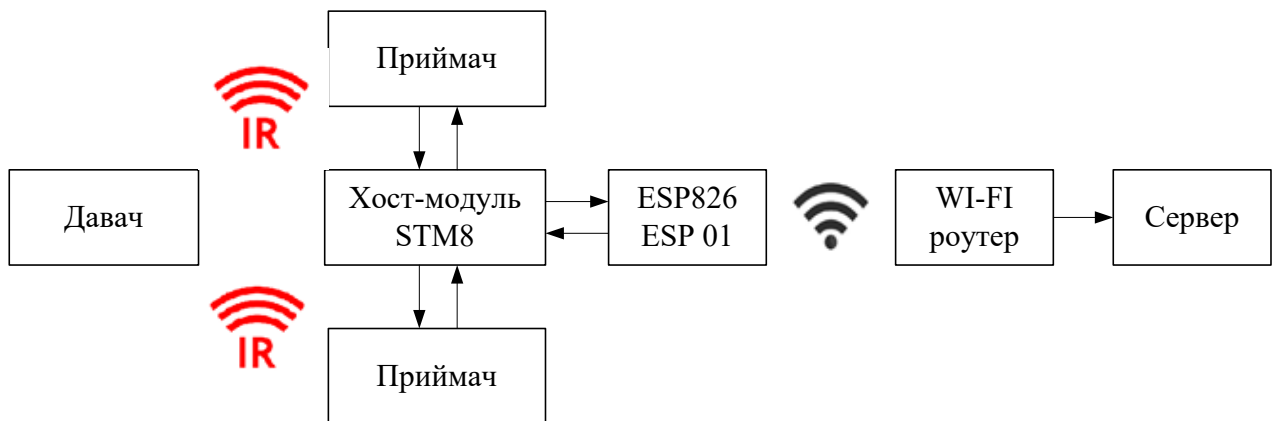


Рис.4.13 - Спрощена структурна схема КАСРБ

Схема підключення виконуючих пристроїв та датчиків КАСРБ зображена на рис. 4.14. [3]

Для прикладу розглянемо двокімнатне приміщення. В кожній з кімнат встановлено інфрачервоний (ІЧ) приймач і відповідний аналізатор на мікроконтролері STM8. Усі такі приймачі об'єднані спільною шиною та підключені до хоста (головного мікроконтролера), який із певною періодичністю опитує кожен приймач про наявність людини в зоні дії. Сам хост, реалізований також на мікроконтролері STM8, з'єднується з Wi-Fi модулем ESP8266, що передає зібрані дані на віддалений сервер для подальшої обробки.

Ідентифікації власника приміщення здійснюється за допомогою RFID-мітки, яка постійно повинна бути у користувача [3].

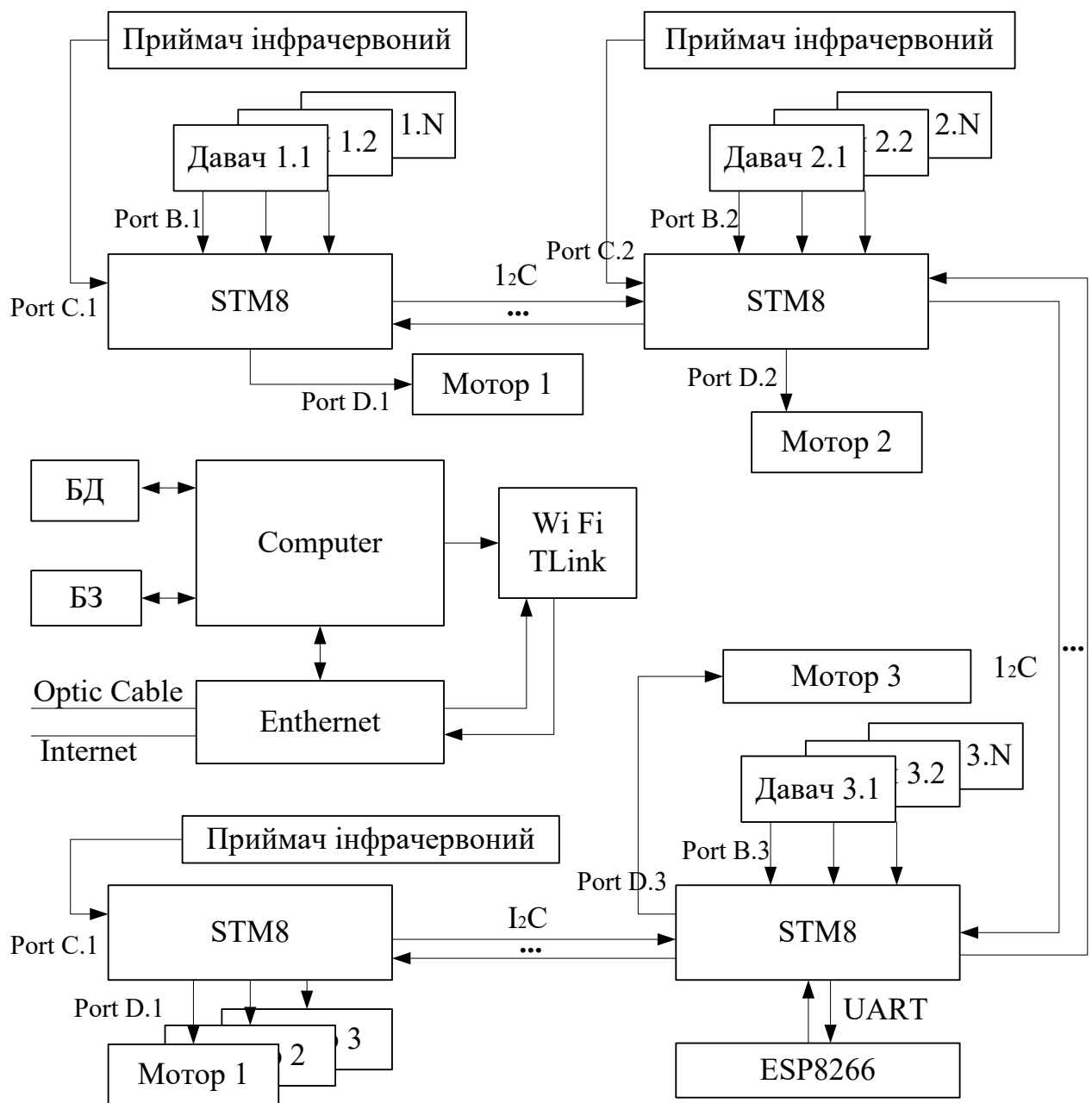


Рис. 4.14 - Схема підключення виконавчих пристроїв та давачів КАСРБ

Система контролю поведінки користувача реалізована з використанням інфрачервоних давачів, що забезпечує отримання інформації при відсутності освітлення. Завдяки великій кількості таких давачів, система точно визначає місце знаходження людини у кімнаті та положення її тіла, щоб мати змогу проаналізувати поведінку людини для прийняття правильного рішення.

Для організації спостереження використовувались модулі з вбудованим інфрачервоним приймальним модулем TSOP38238 [96] та інфрачервоним світлодіодом L-53F3C [99] в передавачі.

КАСРБ функціонує згідно з розробленим алгоритмом. В загальному випадку, алгоритм роботи контролера передбачає такі основні кроки (рис. 4.15) [3]:

Крок 1. Ініціалізація системи - запуск основних модулів контролера та налаштування початкових параметрів.

Крок 2. При виявленні помилки в роботі системи, формується повідомлення про помилку, після чого відбувається повернення на крок 1. Якщо все працює коректно, переходимо до кроку 3.

Крок 3. Почергове опитування усіх контролерів системи.

Крок 4. При відсутності інформація від контролерів системи (немає даних про зміни в системі), алгоритм повертається до кроку 3. В іншому випадку – крок 5.

Крок 5. Попереднє опрацювання отриманої інформації від контролерів - попередній аналіз та перерахунок відповідних коефіцієнтів.

Крок 6. Збереження корисної інформації у базі даних. Оброблені дані фіксуються в БД для подальшого аналізу, навчання чи протоколювання.

Крок 7. Подача послідовності керуючих сигналів на виконавчі пристрої. З урахуванням отриманих даних система формує команди для моторів, актуаторів, реле тощо, які реалізують відповідні дії в системі РБ.

Крок 8. При надходженні команди «Завершення роботи системи», переходимо до кроку 9. У протилежному разі — повертаємося до кроку 3 для продовження циклу опитування.

Крок 9. Підготовка та завершення роботи системи.

При реалізації засобів РБ використано принципи інтелектуальної адаптації системи під поведінку користувача. Розроблена структура адаптивної

системи та контролера РБ ґрунтуються на модульному принципі, що дає змогу швидко вдосконалювати їх у майбутньому.

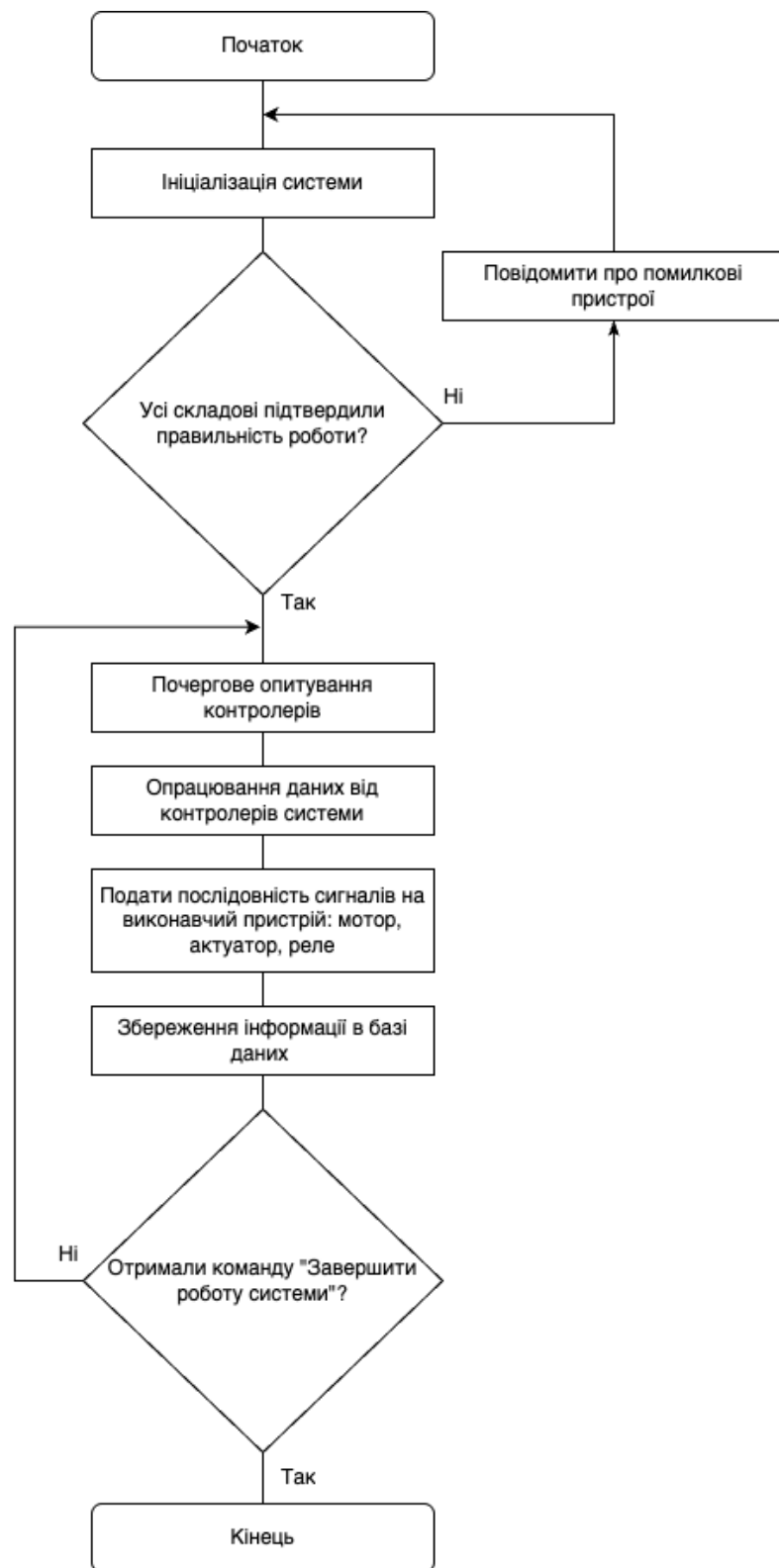


Рис. 4.15 - Блок-схема алгоритму роботи КАСРБ

В основі розробки апаратно-програмних систем лежить вибір засобів реакції системи на зовнішню подію для досягнення поставленої мети. Події бувають однозначні, коли з певної події слідує лише один логічний висновок про вибір реакції на неї, або багатозначні, коли після настання події неможливо зробити конкретний висновок, але можна відкинути деяку множину хибних висновків, якщо використовувати логіку функціонування системи.

Наведемо приклад. Припустимо, що змінився стан одного з входів мікроконтролера. Виявивши при скануванні входів цей сигнал, мікроконтролер переходить на виконання підпрограми обробки цієї інформації. Вибір підпрограми залежать від того який давач системи контролю підключений до цього входу мікроконтролера чи, можливо, цей вхід контролює надходження сигналу від приймача системи дистанційного керування і отримана інформація є командою, надісланою безпосередньо користувачем за допомогою пульта дистанційного керування.

Для того, щоб правильно прийняти інформаційний код який надійшов на вхід мікроконтролера необхідно виконати наступну послідовність дій [3]:

Крок 1. Перервати виконання поточного алгоритму і перейти на обробку переривання для даного входу мікроконтролера.

Крок 2. Якщо цей імпульс - єдиний, і далі інформація не надходить, то треба очистити масив даних для прийому коду, скинути значення робочого таймера в нуль та повернутися у перервану програму щоб продовжити її виконання.

Крок 3. Якщо кількість переданих імпульсів менше заданої за форматом команд керування, то записати в масив поточне значення робочого таймера, скинути цей таймер та повернутися у перервану програму.

Крок 4. Якщо кількість імпульсів більше заданої кількості, то заборонити переривання від даної ніжки, проаналізувати отриманий у масиві код, виконати відповідну йому дію, після чого знову дозволити переривання і повернутися до перерваної програми.

Крок 5. Якщо отриманий код не збігається з кодом жодної з дозволених команд, то надіслати на хост (концентратор даних) повідомлення про помилку.

Наведений приклад ілюструє базову логіку обробки подій із використанням переривань на апаратно-програмному рівні.

Розробка адаптивної системи, проведена в даній роботі, відрізняється від класичних апаратно-програмних рішень. Сам принцип адаптації обумовлює, що система змінюється в процесі функціонування для покращення своєї роботи. Розрізняють параметричну і структурну адаптації. Алгоритми параметричної адаптації знаходять оптимальні значення критичних параметрів, на основі яких система приймає рішення. Структурна адаптація починається із заміни окремих блоків системи на більш вигідні з точки зору загальної мети, і закінчується повним перебором усіх можливих структур системи для пошуку зв'язку структури з конкретизованою метою функціонування [3].

Перед початком розробки необхідно скласти таблицю всіх можливих станів у які може переходити система. Теоретично, ця таблиця буде достатньо великою і може з часом доповнюватись, особливо, якщо врахувати навмисну шкоду, фальсифікацію даних з метою обману, дитячі ігри, домашніх тварин, малоймовірні природні явища тощо. Один стан відрізняється від іншого множиною характерних ознак, які можна отримати, проаналізувавши показники датчиків. Реакція системи повинна бути не 100%, а виваженою, з огляду на те, що може існувати невідомий вплив, який вимагає більш ретельного вивчення. Тому в системі і передбачений концентратор та інтерпретатор даних, на основі яких буде відбуватися подальша адаптація системи.

Аналіз заснований на способі трактування отриманих даних від датчиків як настання одного з станів, що представлений в таблиці переліку можливих. З математичної точки зору – це задача кластеризації і розпізнавання. Пропонується наступна структура розробленого апаратно-програмного забезпечення (рис. 4.16) [3].

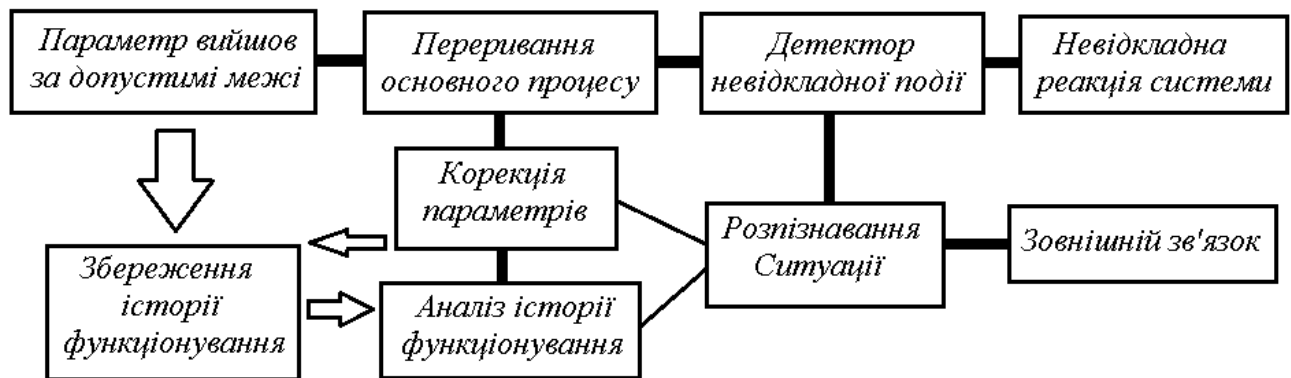


Рис. 4.16 - Структура апаратно-програмного забезпечення

Ядром апаратного забезпечення КАСРБ є мікроконтролер. З огляду на широкий вибір рішень на ринку було проаналізовано низку варіантів і, зрештою, обрано мікроконтролер із лінійки STM8 як бюджетний та доступний. Продуктивності й функціональності цього мікроконтролера достатньо для реалізації поставлених завдань. Зокрема, модель STM8S103F3 має вбудований набір периферійних пристроїв, 8 КБ флеш-пам'яті та 1 КБ оперативної пам'яті [3]. Завдяки невисокій вартості цей мікроконтролер може застосовуватися в багатьох системах.

Мікроконтролер ESP8266 має вбудований модуль Wi-Fi, який дає змогу іншим мікроконтролерам створеної системи отримувати доступ до мережі. Цей мікроконтролер є самодостатньою однокристальною системою (SoC - System On Chip) і забезпечує керування входами та виходами. Залежно від версії реалізації конкретного модуля на ESP8266, можливо мати до 9 інтерфейсів для зв'язку між компонентами комп'ютерної системи (GPIO). Це робить ESP8266 універсальним для застосування у проєктах розумного будинку [3], [100].

Для реалізації давача до плати на основі мікроконтролера STM8S103F3P6 під'єднано інфрачервоний світлодіод. Для забезпечення постійної безперебійної роботи застосовано літієву батарею. Для реалізації приймача до аналогічної плати приєднано приймач інфрачервоного випромінювання TSOP38238. Для побудови макету системи було реалізовано два таких приймачі[3].

Приклад макету давача адаптивної системи РБ наведено на рис. 4.16, а приклад реалізації приймача адаптивної системи РБ наведено на рис. 4.17.



Рис. 4.17 - Приклад макету реалізації давача адаптивної системи РБ

Отже, розроблене технічне забезпечення КАСРБ характеризується низькою ціною та широкими функціональними можливостями, що обумовлено використанням мікроконтролерів.

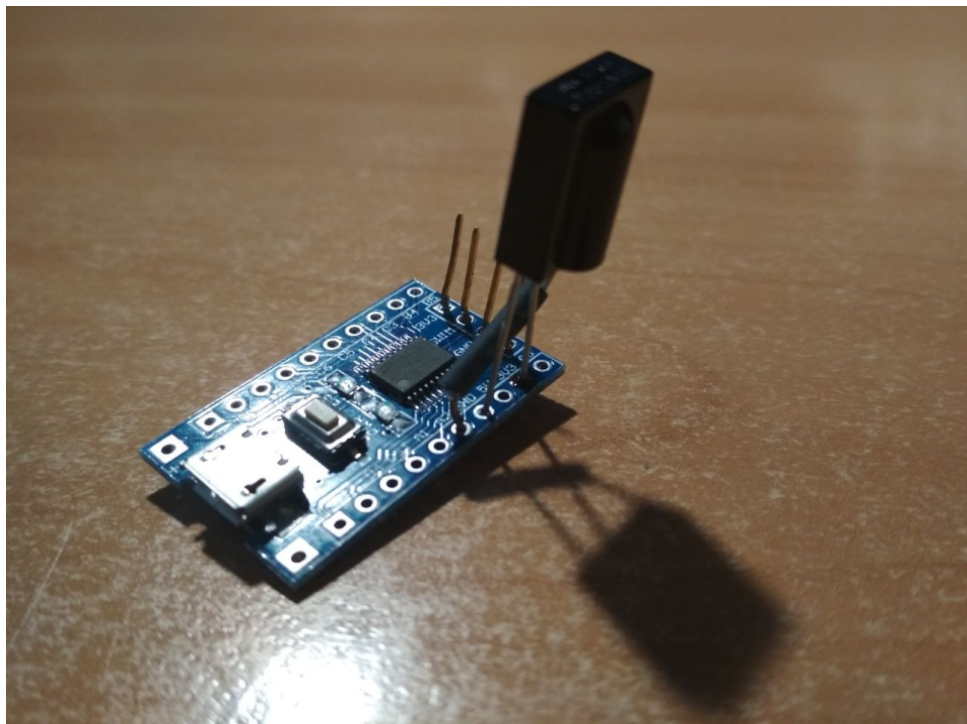


Рис. 4.18 - Приклад реалізація приймача адаптивної системи РБ

Для реалізації хоста до плати STM8S103F3P6 було приєднано Wi-Fi модуль ESP8266 ESP01.

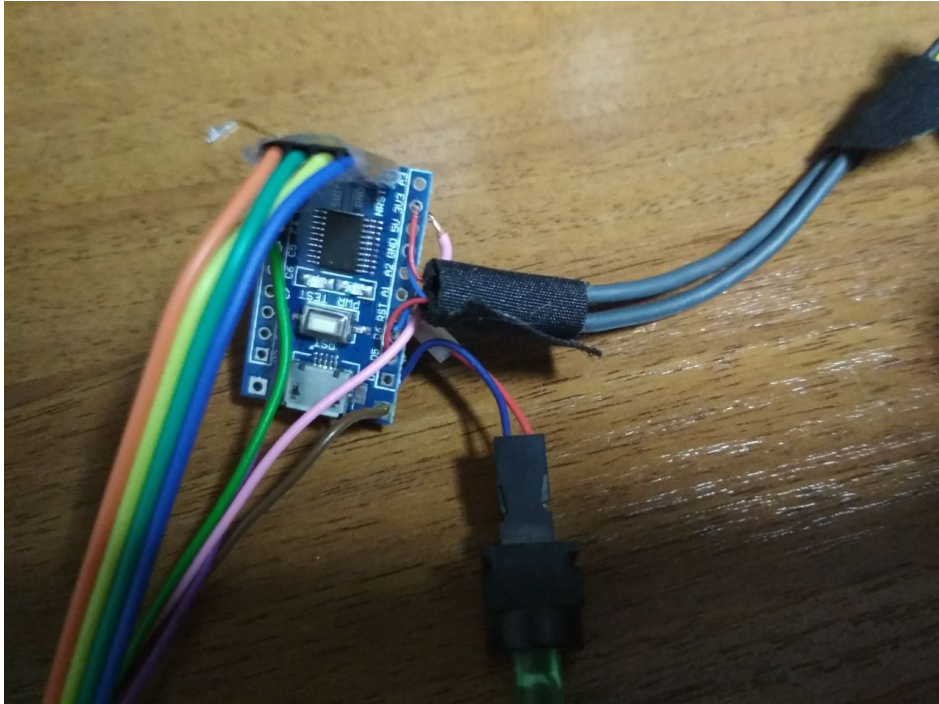


Рис. 4.19 - Реалізація хоста

Для реалізації цілісної системи всі пристрої з'єднані спільною шиною яка передає дані за протоколом I²C (рис. 4.20).

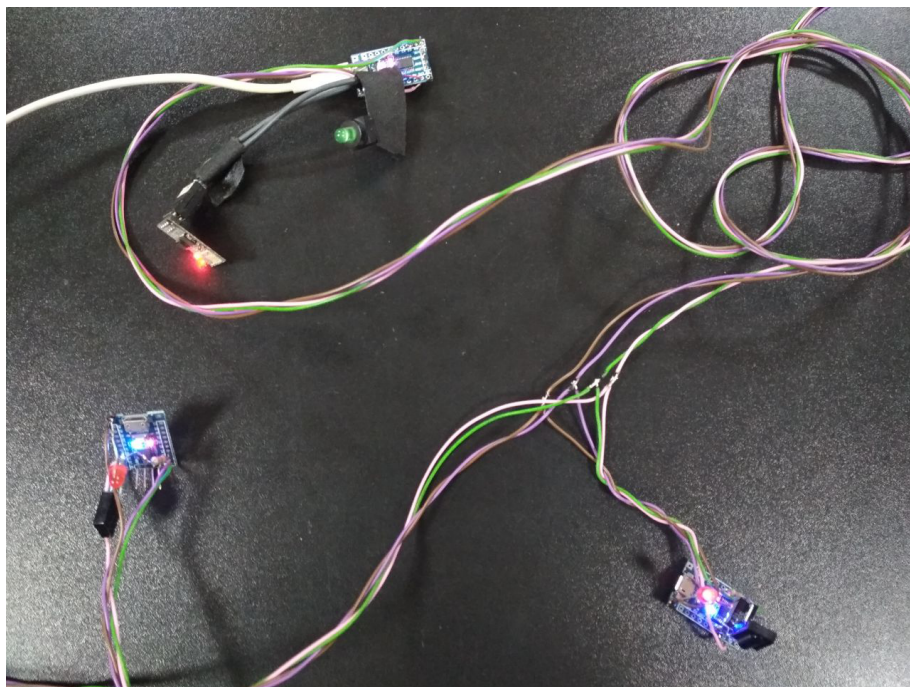


Рис. 4.20 - Реалізація системи

4.5.3 Особливості розроблення спеціалізованого програмного забезпечення контролера адаптивної системи РБ

Спеціалізоване програмне забезпечення для КАСРБ розроблялось в середовищі ST Visual Develop [3]. Для програмування мікроконтролерів STM8 використано спеціальний програматор ST-Link V2 mini. Програми реалізовані з використанням АТ-команд та мови програмування асемблер для мікроконтролерів STM8.

Для реалізації функціонування давача було проведено дослідження параметрів модуля. Визначено резонансну частоту передавача за допомогою алгоритму, який змінює тривалість імпульсів [3]. Після декількох експериментів, було визначено резонансну частоту при тривалості імпульсів 7, 14 та 15 одиниць. Фрагмент коду функціонування давача показано на рис. 4.21, а фрагмент коду функціонування приймача показано на рис. 4.22.

```
91 ;=====
92 PAUSE:
93     ldw X,#65000
94 L_PAUSE_X:
95     decw X
96     jrne L_PAUSE_X
97     ret
98
99 ;-----
100 SEND_1:
101     ld A,#7
102     ld PAUSE_C3,A ; початкове значення між імпульсами
103     ldw X,#400 ; кількість імпульсів
104 L3_C:
105     pushw X
106     bset PC_ODR,#3
107
108     ld A,PAUSE_C3
109 L1_C:
110     dec A
```

Рис. 4.21 - Фрагмент коду реалізації функціонування давача

```

45     ldw X,#stack_end
46     ldw SP,X
47
48     sim
49     call init_port      ;ініціалізація портів
50
51     ld A,#10    ; початковий тест світлодіода B5
52 MM1:
53     push A
54     bres PB_ODR,#5
55     call PAUSE
56     bset PB_ODR,#5
57     call PAUSE
58     pop A
59     dec A
60     jrne MM1
61
62     ; читаємо з порту A2 та передаємо на B5
63 L_TEST_A2:
64     btit PA IDR.#2.PA2 ON

```

Рис. 4.22 - Фрагмент коду реалізації функціонування приймача

Реалізація протоколу I²C для хоста показано на рис. 4.23.

```

112
113     ldw X,#10000    ; цикл очікування
114     ld A,#50    ; адреса для SLAVE or 1 для читання з SLAVE-пристрою
115     or A,#1
116     ld I2C_DR,A ; подаємо адресу на шину I2C
117 L_WAIT_ADDR_MATCHED1:
118     decw X
119     jreq SEND_AGAIN ; не дочекалися – починаємо спочатку
120     btjif I2C_SR1,#1,L_WAIT_ADDR_MATCHED1 ; чекаємо, доки SLAVE не підтвердить
        свою адресу
121     ld A,I2C_SR3    ; очищення біту ADDR
122
123
124     ld A,#111
125     ld I2C_DR,A ; подаємо ID команди на пристрій
126
127     ; останній байт не підтверджувати
128     bres I2C_CR2,#2
129
130     ldw X.#10000    ; цикл очікування

```

Рис. 4.23 - Фрагмент коду реалізації протоколу I²C для хоста

Реалізація протоколу I²C для приймача (рис. 4.24):

```

50
51     sim
52     call init_port      ;ініціалізація портів
53
54     ;-----
55     ; Загальна ініціалізація I2C
56     ; set I2C_FREQR=00000001, (1 Mhz) clock I2C_CCRL =60
57
58     ld A,#1
59     ld I2C_FREQR,A
60
61     ; відключаємо I2C
62     ld A,#0
63     ld I2C_CR1,A
64
65     ld A,#0
66     ld I2C_CCRH,A
67
68     ld A,#80 ; CCR = Fmaster/2*Fiic= 16MHz/2*100kHz
69     ld I2C_CCRL,A

```

Рис. 4.24 - Частина коду реалізації протоколу I²C для приймача

Розроблене ПЗ з використанням Асемблера STM8 забезпечує високу швидкодію контролера [3].

4.6 Розроблення автоматизованої програмної системи адміністрування та моніторингу адаптивної системи РБ

Основною задачею, яку необхідно реалізувати в даному підрозділі дисертаційної роботи, є розроблення універсальної автоматизованої програмної системи адміністрування та моніторингу (АПСАМ) адаптивної системи РБ [8], [17]. При цьому, універсальність полягає в можливості імпорту в розроблену програмну систему моделей різноманітних систем РБ, із забезпеченням повноцінного моніторингу проєктованої системи РБ та базових механізмів її адміністрування. Реалізувати таку універсальність розроблюваної програмної системи можна виключно шляхом максимально детермінованої класифікації абсолютно всіх складових компонентів проєктованої системи РБ. В свою чергу, моніторинг проєктованої системи РБ в розроблюваній програмній системі найбільш доцільно реалізувати шляхом графічної візуалізації проєктованої

системи РБ, її основних складових компонентів, а також індикатора найбільш імовірної розрахункової локації користувачів, нанесених на план будівлі (фізичної оболонки функціонування проєктованої системи РБ), і представити всю цю графічну інформацію в основному вікні розроблюваної програмної системи.

Враховуючи вищезазначене, конфігураційний файл кожної окремої моделі проєктованої системи РБ, що імпортується до розробленої АПСАМ, обов'язково має містити наступну інформацію про проєктовану систему РБ у чітко визначеному порядку [17]:

1. графічний план будівлі (з уніфікованим позначенням приміщень, стін, дверей, вікон, побутової техніки, і т.д.);

2. <сенсори>

<загальний порядковий номер функціонального компонента>

*ціле число

</загальний порядковий номер функціонального компонента>

<категорія>

*«сенсор_освітлення»/«сенсор_вологості»/«сенсор_руху»/і т.п.

</категорія>

<заводське маркування>

*текст: заводське маркування даного конкретного сенсора

</заводське маркування>

<координати розміщення сенсора в межах будівлі>

<координата X>

*ціле число

</координата X>

<координата Y>

*ціле число

</координата Y>

</сенсори>

3. <актуатори>

<загальний порядковий номер функціонального компонента>

*ціле число

</загальний порядковий номер функціонального компонента>

<категорія>

*«реле_ввімкнення_кавоварки»/«контролер_ТВ»/«реле_освітлення»/і т.п.

</категорія>

<заводське маркування>

*текст: заводське маркування даного конкретного актуатора

</заводське маркування>

<координати розміщення актуатора в межах будівлі>

<координата X>

*ціле число

</координата X>

<координата Y>

*ціле число

</координата Y>

<порогове значення ймовірності запуску актуатора>

*дійсне число

</актуатори>

4. <користувачі>

<загальний порядковий номер користувача>

*ціле число

</загальний порядковий номер користувача>

<пріоритет користувача>

*ціле число

</пріоритет користувача>

<стать користувача>

*Чол./Жін.

</стать користувача>

<ім'я користувача>

*тест – ім'я користувача

</ім'я користувача>

<короткий опис особистості користувача>

*тест – короткий опис основних особистісних характеристик користувача

</короткий опис особистості користувача>

</користувачі>

Для розробки автоматизованої програмної системи адміністрування та моніторингу систем РБ було використано середовище розробки Java IntelliJ IDEA та мову програмування Java [8]. Ключовими причинами вибору цього середовища стали наступні переваги Java:

- платформонезалежність: завдяки Java Virtual Machine (JVM) додатки, написані на Java, працюють на будь-якій операційній системі, що значно спрощує їх розгортання та експлуатацію;
- об'єктно-орієнтований підхід та модульність: Java забезпечує сувору типізацію та чітке розділення модулів, що сприяє створенню масштабованих і легких для підтримки програмних рішень;
- багата екосистема: широкий набір бібліотек і фреймворків (Spring, Hibernate, JavaFX тощо) дозволяє швидко реалізувати складний функціонал, інтегрувати зовнішні сервіси та ефективно вирішувати завдання бізнес-логіки;
- надійність і безпека: вбудовані засоби перевірки типів, обробки виключень, а також розвинені механізми безпеки забезпечують стабільну і захищену роботу системи;
- автоматичне управління пам'яттю: наявність збирача сміття (garbage collector) дозволяє звільнити розробника від ручного керування пам'яттю, що знижує ризик помилок, пов'язаних із витоками пам'яті;

– підтримка багатопоточності: мова Java забезпечує потужні механізми для реалізації паралельних обчислень, що є важливим для систем реального часу і високопродуктивних додатків.

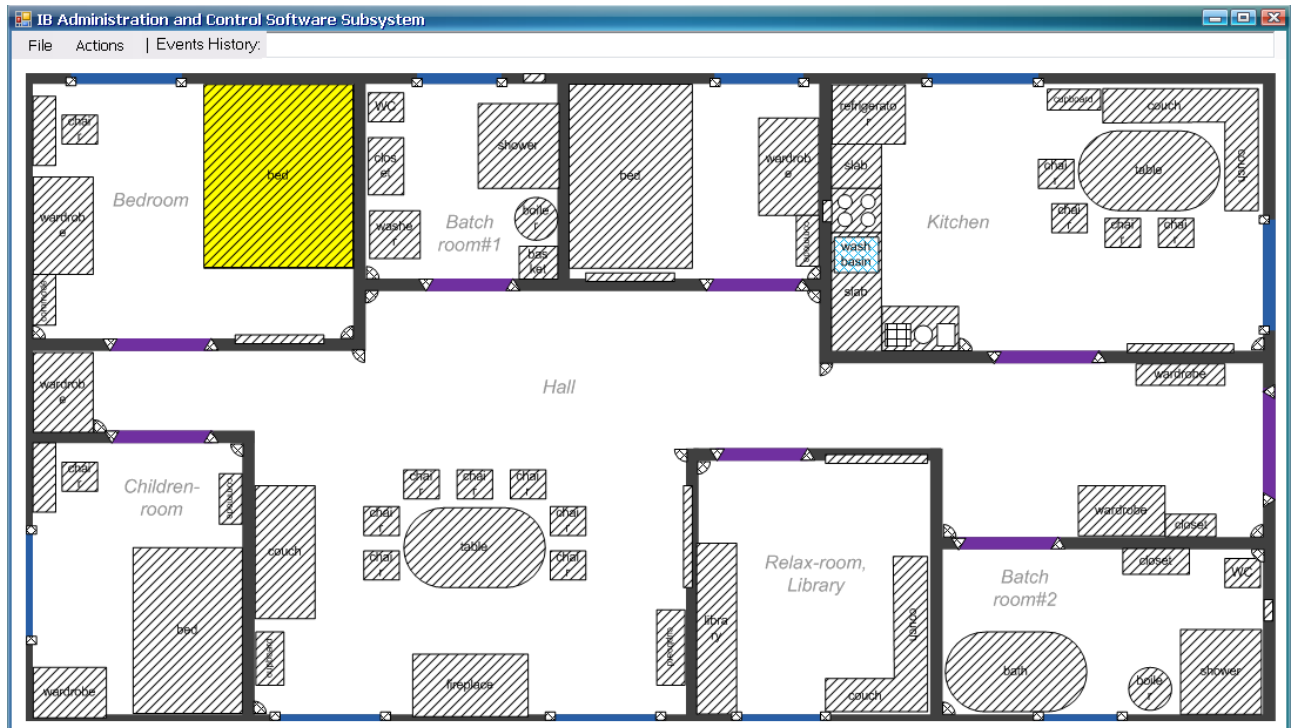


Рис. 4.25 - Головне вікно розробленої АПСАМ з інтерактивним зображенням - моделлю проєктованої системи РБ

Ці переваги роблять Java надзвичайно популярною платформою для розробки як серверних, так і десктопних програм, що особливо важливо при створенні комплексних систем з високим рівнем інтеграції.

Отже, інтерфейс розроблений АПСАМ надзвичайно лаконічний, і складається з основного вікна підсистеми, де представлено інтерактивну модель проєктованої системи РБ (рис. 4.25) [17].

Основне меню розробленої АПСАМ містить наступні елементи (рис. 4.26):

– “File” → “Load IB-scheme” – діалогове вікно відкриття/завантаження графічного файлу-схеми (плану) будівлі проєктованої системи РБ;

– “File” → “Load IB-parameters” – діалогове вікно відкриття/завантаження файлу параметрів проєктованої системи РБ;

- “File”→”Save IB-parameters” – діалогове вікно збереження та перезапису файлу параметрів проєктованої системи РБ після того, як параметри елементів інтерактивного зображення-моделі проєктованої системи РБ були змінені користувачем чи адміністратором АПСАМ;
- “File”→”Load IB DB” – діалогове вікно відкриття/завантаження файлу бази даних, отриманого в результаті функціонування апаратної частини – розробленого спеціалізованого мікроконтролера моніторингу та імовірнісного прогнозування зміни станів систем РБ;
- “Actions”→”Run loaded IB DB” (стає доступним тільки після ”→”Load IB DB”) – запускає зчитування даних із завантаженого файлу (дані, отримані від апаратної частини – спеціалізованого мікроконтролера моніторингу та імовірнісного прогнозування) для подальшого моделювання і відображення взаємодії користувача з системою в інтерактивній моделі. У режимі реального часу відкривається файл потоку даних від мікроконтролера, з якого і відбувається зчитування даних та відображення їх в розробленій АПСАМ в режимі реального часу.

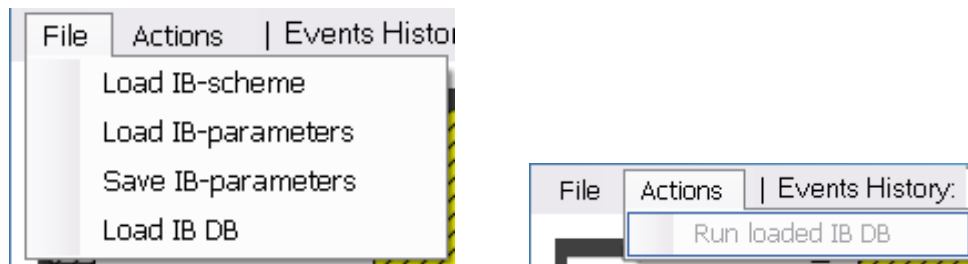


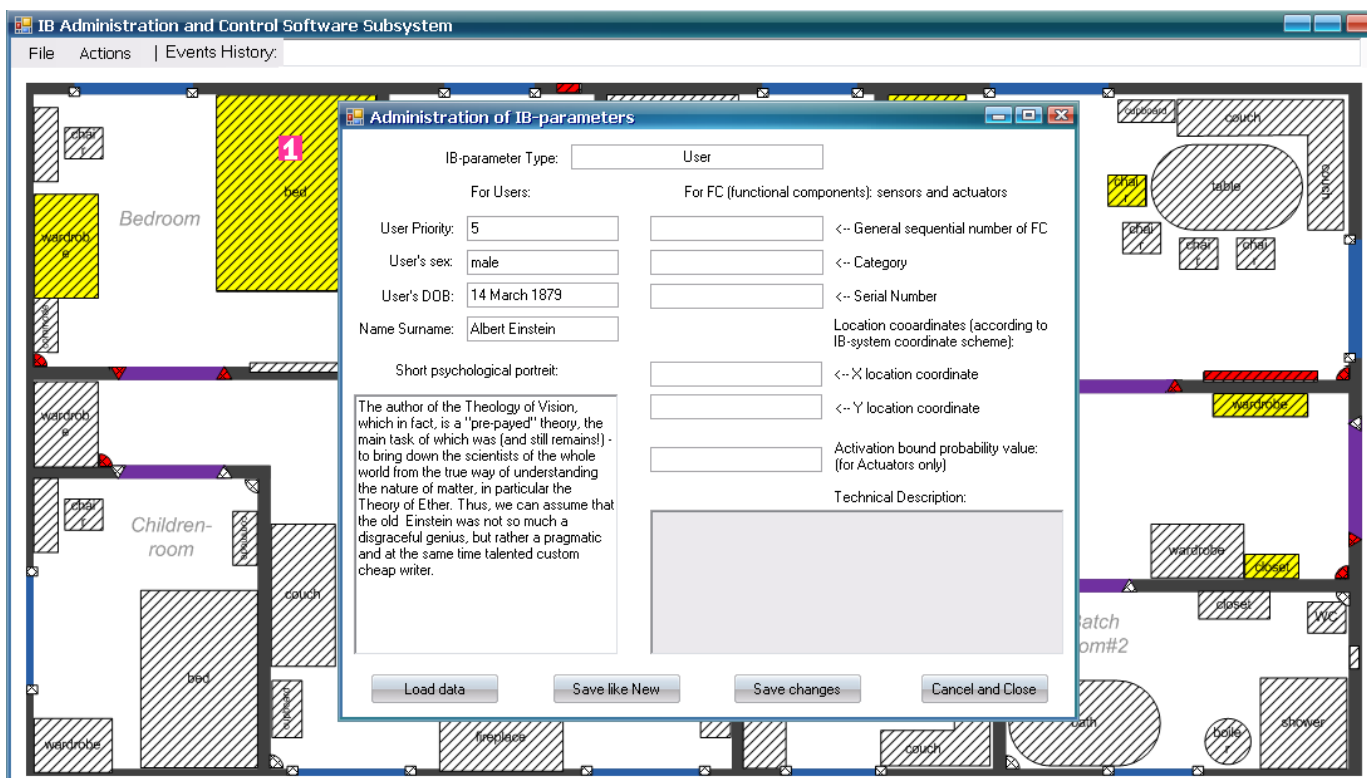
Рис. 4.26 - Меню розробленої АПСАМ

У полі “Events History” записується інформація про спрацювання сенсорів і актуаторів системи РБ із зазначенням дати та часу їх активації чи деактивації, а також впливаючою з цього інформацією про активність користувачів [17].

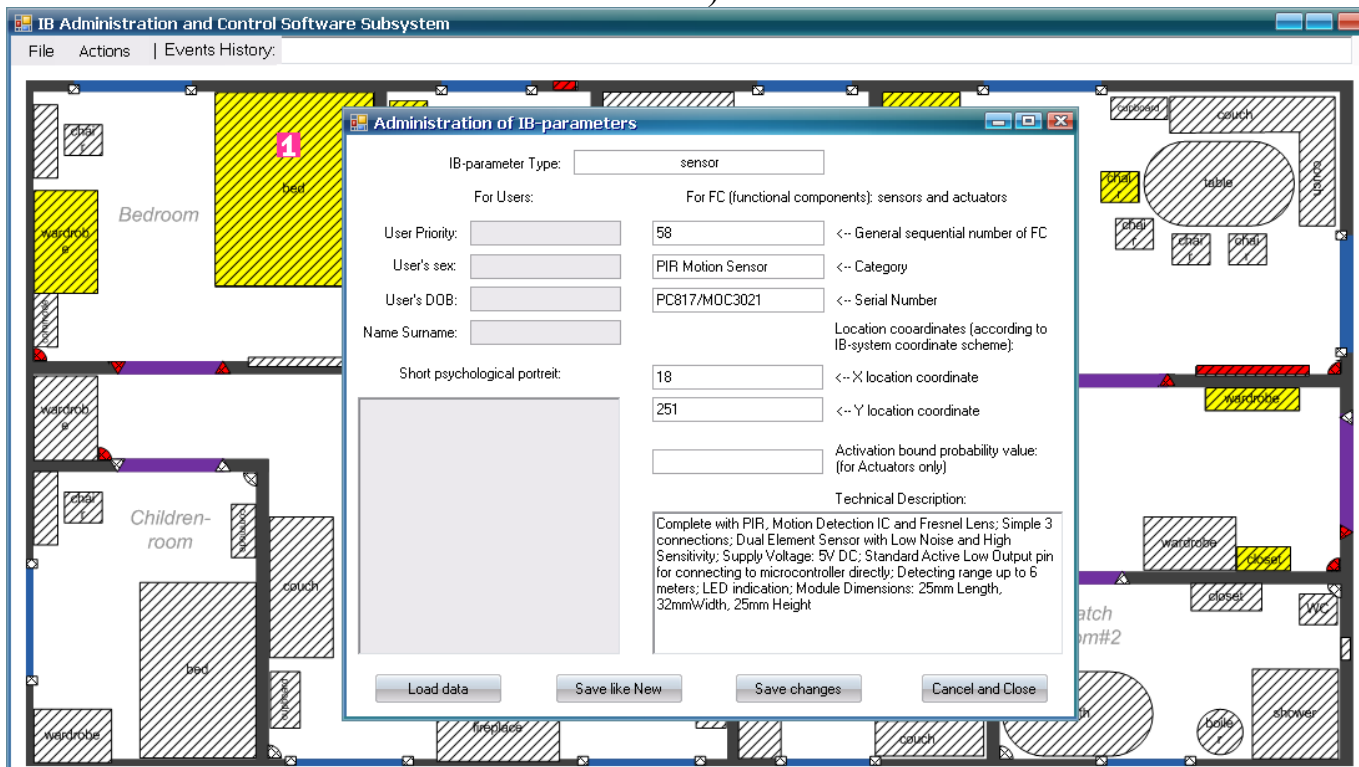
eventHistory_example.evh				
1	09.09.2024	09:42:11	Sensor #58	Activated. => User is moving in bedroom.
2	09.09.2024	09:42:11	Sensor #57	Activated. => User is moving in bedroom.
3	09.09.2024	09:42:13	Sensor #58	Deactivated. => User is Not moving in bedroom.
4	09.09.2024	09:42:14	Sensor #57	Deactivated. => User is Not moving in bedroom.
5	09.09.2024	09:42:16	Sensor #58	Activated. => User is moving in bedroom.
6	09.09.2024	09:42:16	Sensor #57	Activated. => User is moving in bedroom.
7	09.09.2024	09:42:16	Sensor #56	Deactivated. => User stands up from the bed.
8	09.09.2024	09:42:17	Actuator #65	Activated. => Boiler in bathroom1 is turned ON.
9	09.09.2024	09:42:18	Sensor #65	Activated. => User location changed: bedroom -> hall
10	09.09.2024	09:42:18	Sensor #62	Activated. => User is moving in hall near bedroom children-room.
11	09.09.2024	09:42:18	Sensor #63	Activated. => User is moving in hall near coach.
12	09.09.2024	09:42:19	Sensor #58	Deactivated. => User is Not moving in bedroom.
13	09.09.2024	09:42:19	Sensor #57	Deactivated. => User is Not moving in bedroom.
14	09.09.2024	09:42:21	Sensor #58	Activated. => User is moving in bedroom.
15	09.09.2024	09:42:21	Sensor #57	Activated. => User is moving in bedroom.
16	09.09.2024	09:42:23	Sensor #58	Deactivated. => User is Not moving in bedroom.
17	09.09.2024	09:42:23	Sensor #57	Deactivated. => User is Not moving in bedroom.
18	09.09.2024	09:42:23	Sensor #62	Deactivated. => User is Not moving in hall near bedroom children-room.
19	09.09.2024	09:42:23	Sensor #63	Deactivated. => User is Not moving in hall near coach.
20	09.09.2024	09:42:24	Sensor #61	Activated. => User is moving in hall near bathroom1.
21	09.09.2024	09:42:25	Sensor #61	Deactivated. => User is Not moving in hall near bathroom1.
22	09.09.2024	09:42:25	Sensor #63	Activated. => User is moving in hall near coach.
23	09.09.2024	09:42:26	Sensor #62	Activated. => User is moving in hall near bedroom children-room.
24	09.09.2024	09:42:26	Sensor #63	Deactivated. => User is Not moving in hall near coach.
25	09.09.2024	09:42:27	Sensor #64	Activated. => User location changed: hall -> bathroom1.
26	09.09.2024	09:42:27	Sensor #60	Activated. => User is moving in bathroom1.
27	09.09.2024	09:42:28	Sensor #61	Activated. => User is moving in hall near bathroom1.
28	09.09.2024	09:42:29	Sensor #60	Deactivated. => User is Not moving in bathroom1.
29	09.09.2024	09:42:29	Sensor #61	Deactivated. => User is Not moving in hall near bathroom1.
30	09.09.2024	09:42:30	Sensor #60	Activated. => User is moving in bathroom1.
31	09.09.2024	09:42:31	Sensor #60	Deactivated. => User is Not moving in bathroom1.
32	09.09.2024	09:42:32	Sensor #66	Activated. => User is on bowl in bathroom1.
33	09.09.2024	09:42:33	Sensor #66	Deactivated. => User stands up from toilet in bathroom1.
34	09.09.2024	09:42:34	Sensor #66	Activated. => User is on bowl in bathroom1.
35	09.09.2024	09:42:35	Sensor #66	Deactivated. => User stands up from toilet in bathroom1.
36	09.09.2024	09:42:36	Sensor #66	Activated. => User is on bowl in bathroom1.
37	09.09.2024	09:42:37	Sensor #66	Deactivated. => User stands up from toilet in bathroom1.
38	09.09.2024	09:42:38	Sensor #67	Activated. => User came into showerbox in bathroom1.
39	09.09.2024	09:42:39	Sensor #67	Deactivated. => User left showerbox in bathroom1.
40	09.09.2024	09:42:39	Sensor #60	Activated. => User is moving in bathroom1.
41	09.09.2024	09:42:40	Sensor #60	Deactivated. => User is Not moving in bathroom1.
42	09.09.2024	09:42:41	Sensor #67	Activated. => User came into showerbox in bathroom1.
43	09.09.2024	09:42:42	Sensor #67	Deactivated. => User left showerbox in bathroom1.
44	09.09.2024	09:42:42	Sensor #60	Activated. => User is moving in bathroom1.
45	09.09.2024	09:42:43	Sensor #60	Deactivated. => User is Not moving in bathroom1.
46	09.09.2024	09:42:44	Sensor #64	Activated. => User location changed: bathroom1 -> hall.
47	09.09.2024	09:42:44	Sensor #61	Activated. => User is moving in hall near bathroom1.
48	09.09.2024	09:42:45	Sensor #60	Activated. => User is moving in bathroom1.
49	09.09.2024	09:42:46	Sensor #60	Deactivated. => User is Not moving in bathroom1.
50	09.09.2024	09:42:46	Sensor #61	Deactivated. => User is Not moving in hall near bathroom1.
51	09.09.2024	09:42:47	Sensor #61	Activated. => User is moving in hall near bathroom1.
52	09.09.2024	09:42:48	Sensor #61	Deactivated. => User is Not moving in hall near bathroom1.
53	09.09.2024	09:42:49	Sensor #64	Activated. => User location changed: hall -> bathroom1.
54	09.09.2024	09:42:49	Sensor #60	Activated. => User is moving in bathroom1.
55	09.09.2024	09:42:50	Sensor #60	Deactivated. => User is Not moving in bathroom1.
56	09.09.2024	09:42:51	Sensor #60	Activated. => User is moving in bathroom1.
57	09.09.2024	09:42:52	Sensor #60	Deactivated. => User is Not moving in bathroom1.
58	09.09.2024	09:42:53	Sensor #64	Activated. => User location changed: bathroom1 -> hall.
59	09.09.2024	09:42:53	Sensor #61	Activated. => User is moving in hall near bathroom1.
60	09.09.2024	09:42:54	Sensor #61	Deactivated. => User is Not moving in hall near bathroom1.
61	09.09.2024	09:42:55	Sensor #64	Activated. => User location changed: hall -> bathroom1.

Рис. 4.27 - Фрагмент Events History для одного із модельованих сценаріїв

На рис. 4.27 наведений приклад Events History для одного із модельованих сценаріїв.



a)



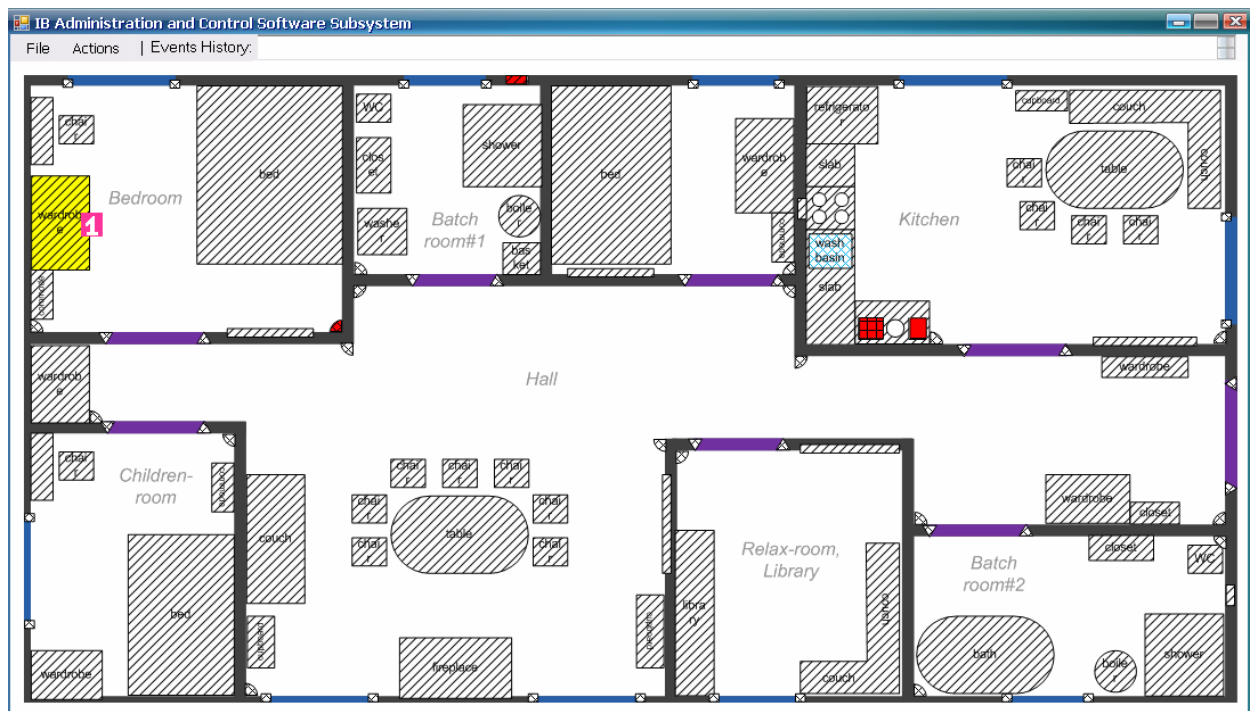
b)

Рис. 4.28 - Вікно з інформацією про характеристики елементів інтерактивного зображення-моделі проєктованої системи РБ

Також, якщо натиснути клавішею миші на будь-який елемент інтерактивного зображення-моделі основного вікна розробленої АПСАМ, відкриється вікно з інформацією про характеристики даного елемента (рис. 4.28), з можливістю редагування, або ж створення нового елемента.

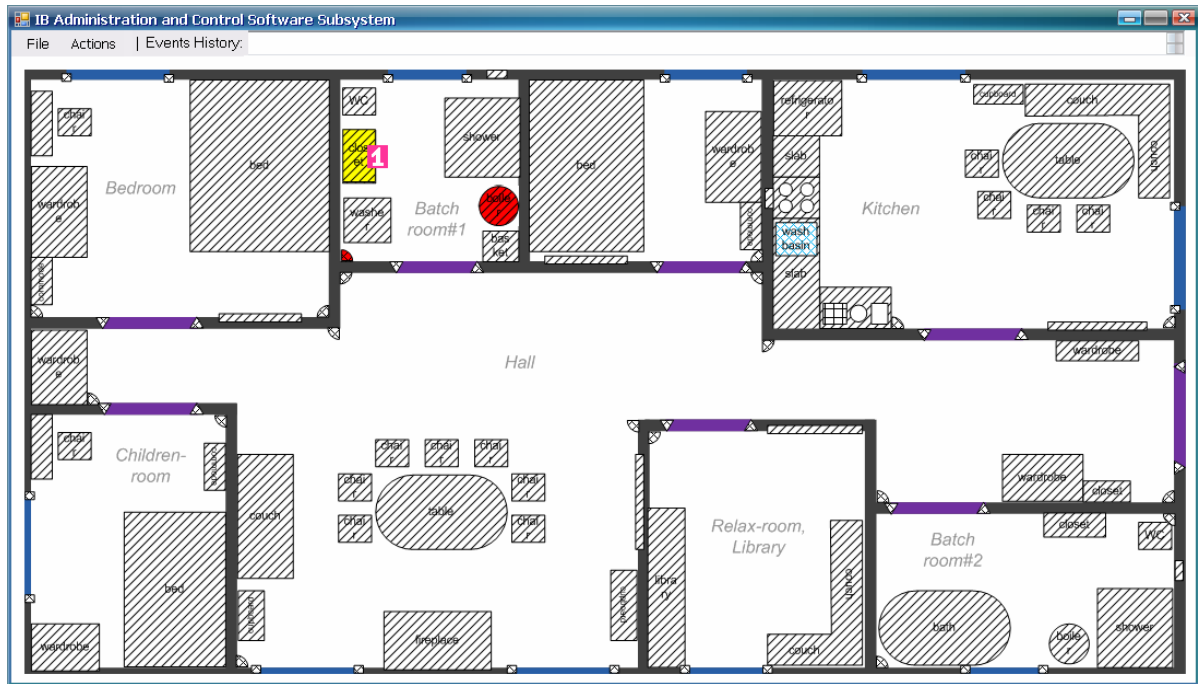
4.7 Результати моделювання

На рис. 4.29, наведені деякі кроки моделювання в ході функціонування розробленої АПСАМ на прикладі одного із можливих сценаріїв функціонування проєктованої системи РБ та її взаємодії із користувачем [17].

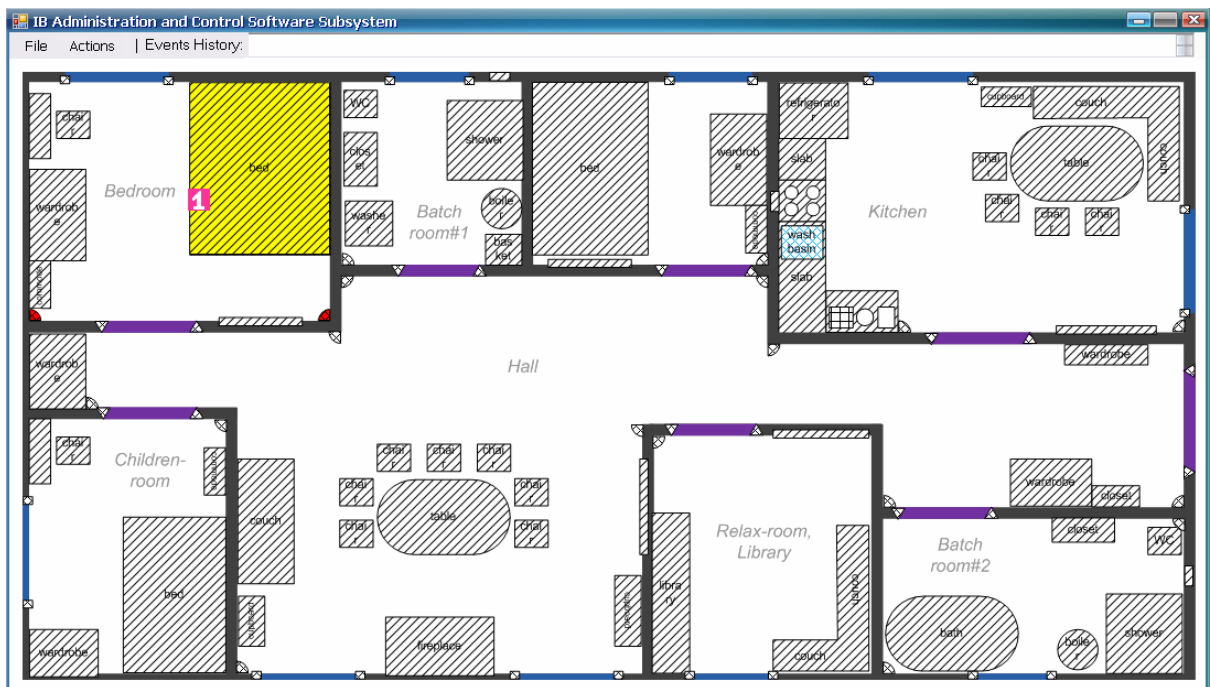


а)

Рис. 4.29 - Кроки функціонування АПСАМ на прикладі сценарію функціонування проєктованої системи РБ та її взаємодії із користувачем

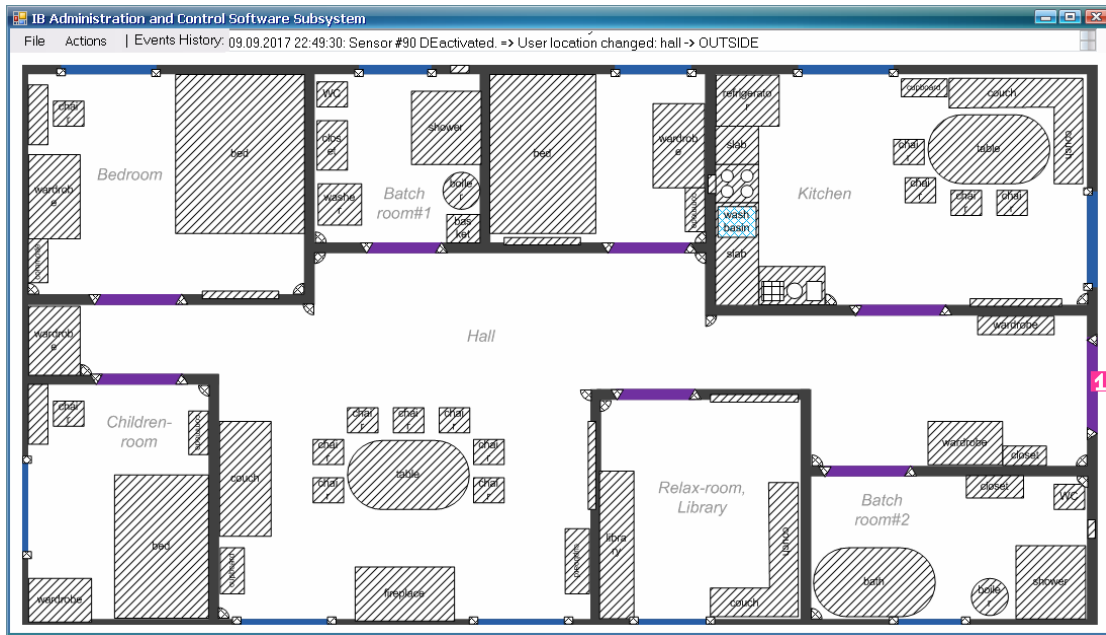


б)

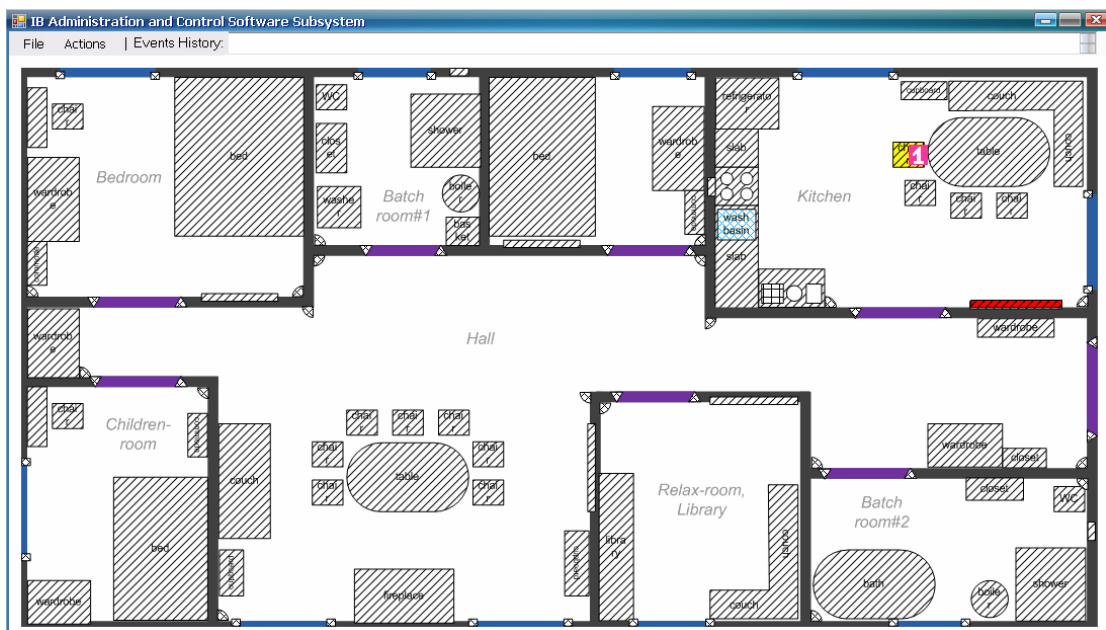


с)

Рис. 4.29 - Кроки функціонування АПСАМ на прикладі сценарію функціонування проєктованої системи РБ та її взаємодії із користувачем (Продовження)

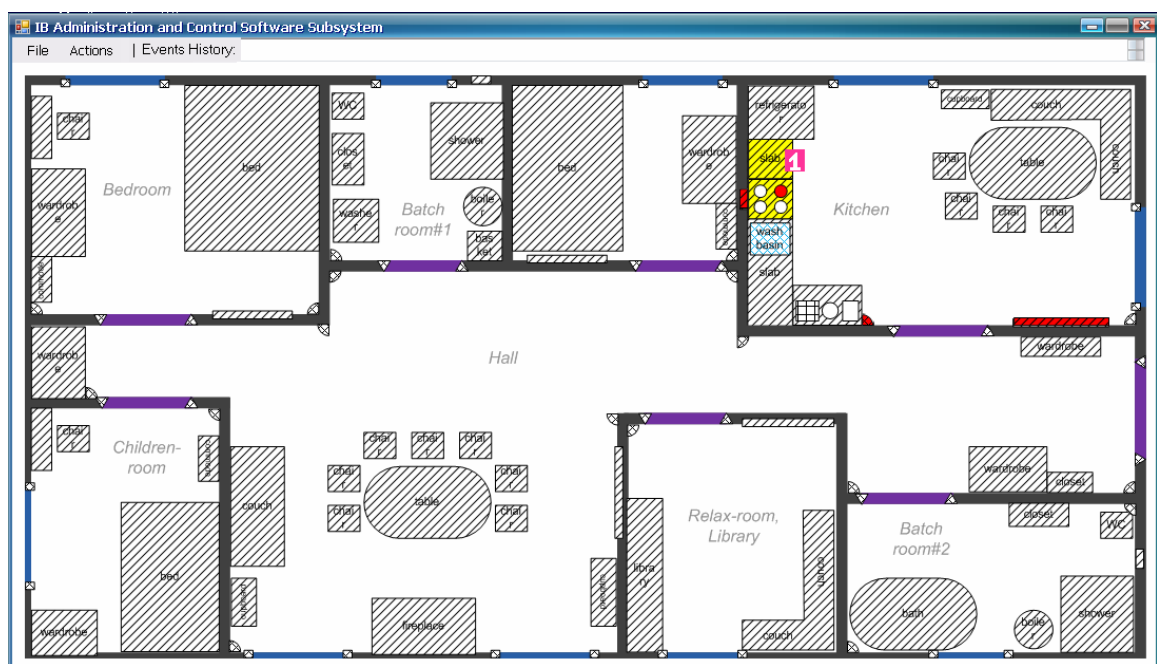


д)



е)

Рис. 4.29 - Кроки функціонування АПСАМ на прикладі сценарію функціонування проєктованої системи РБ та її взаємодії із користувачем (Продовження)



g)

Рис. 4.29 - Кроки функціонування АПСАМ на прикладі сценарію функціонування проєктованої системи РБ та її взаємодії із користувачем (Закінчення)

В таблиці 4.1 наведений повний перелік подій з EventHistory-модуля розробленої АПСАМ в ході її функціонування [17]:

Таблиця 4.1

Перелік подій з EventHistory-модуля розробленої АПСАМ

Час	Подія
17.11.2024 07:57:46:	Sensor #58 Activated. => User is moving in bedroom.
17.11.2024 07:57:47:	Sensor #57 Activated. => User is moving in bedroom.
17.11.2024 07:57:53:	Sensor #56 DEactivated. => User stands up from the bed.
17.11.2024 07:57:53:	Actuator #65 Activated. => Boiler in bathroom#1 is turned ON.
17.11.2024 07:57:56:	Sensor #59 Activated. => User location change: bedroom -> hall
17.11.2024 07:57:56:	Sensor #60 Activated. => User is moving in hall near bedroom children-room.
17.11.2024 07:57:56:	Sensor #61 Activated. => User is moving in hall near coach.
17.11.2024 07:57:56:	Sensor #57 DEactivated. => User is Not moving in bedroom.
17.11.2024 07:57:56:	Sensor #58 DEactivated. => User is Not moving in bedroom.
17.11.2024 07:57:56:	Sensor #59 DEactivated. => User location changed: bedroom -> hall
17.11.2024 07:58:00:	Sensor #60 DEactivated. => User is Not moving in hall near bedroom children-room.

Таблица 4.1 (Продовження)

17.11.2024 07:58:00:	Sensor #61 DEactivated. => User is Not moving in hall near coach.
17.11.2024 07:58:00:	Sensor #62 Activated. => User is moving in hall near bathroom#1.
17.11.2024 07:58:03:	Sensor #63 Activated. => User location change: hall -> bathroom#1
17.11.2024 07:58:04:	Sensor #62 DEactivated. => User is Not moving in hall near bathroom#1.
17.11.2024 07:58:04:	Sensor #63 DEactivated. => User location changed: hall -> bathroom#1
17.11.2024 07:58:04:	Sensor #64 Activated. => User is moving in bathroom#1.
17.11.2024 07:58:06:	Sensor #68 Activated. => User is near closet in bathroom#1.
17.11.2024 07:58:52:	Sensor #68 DEactivated. => User is away from closet in bathroom#1.
17.11.2024 07:58:53:	Sensor #66 Activated. => User set on bowl in bathroom#1.
17.11.2024 07:59:34:	Sensor #66 DEactivated. => User stands up from bowl in bathroom#1.
17.11.2024 07:59:34:	Actuator #71 Activated. => Fen in bathroom#1 is turned ON.
17.11.2024 07:59:36:	Sensor #67 Activated. => User came into showerbox in bathroom#1.
17.11.2024 07:59:36:	Sensor #64 DEactivated. => User is Not moving in bathroom#1.
17.11.2024 08:02:04:	Sensor #64 Activated. => User is moving in bathroom#1.
17.11.2024 08:02:04:	Sensor #67 DEactivated. => User came out from the showerbox in bathroom#1.
17.11.2024 08:02:06:	Sensor #69 Activated. => User location change: bathroom#1 -> hall
17.11.2024 08:02:07:	Sensor #62 Activated. => User is moving in hall near bathroom#1.
17.11.2024 08:02:07:	Sensor #64 DEactivated. => User is Not moving in bathroom#1.
17.11.2024 08:02:07:	Actuator #65 DEactivated. => Boiler in bathroom#1 is turned OFF.
17.11.2024 08:02:07:	Sensor #69 DEactivated. => User location changed: bathroom#1 -> hall
17.11.2024 08:02:10:	Sensor #60 Activated. => User is moving in hall near bedroom children-room.
17.11.2024 08:02:10:	Sensor #61 Activated. => User is moving in hall near coach.
17.11.2024 08:02:10:	Sensor #62 DEactivated. => User is Not moving in hall near bathroom#1.
17.11.2024 08:02:14:	Sensor #70 Activated. => User location change: hall -> bedroom
17.11.2024 08:02:14:	Actuator #82 Activated. => Cofee-machine is turned ON.
17.11.2024 08:02:14:	Actuator #83 Activated. => Toster-machine is turned ON.
17.11.2024 08:02:14:	Sensor #57 Activated. => User is moving in bedroom.
17.11.2024 08:02:14:	Sensor #58 Activated. => User is moving in bedroom.
17.11.2024 08:02:14:	Sensor #60 DEactivated. => User is Not moving in hall near bedroom children-room.
17.11.2024 08:02:14:	Sensor #61 DEactivated. => User is Not moving in hall near coach.

Таблиця 4.1 (Продовження)

17.11.2024 08:02:14:	Sensor #70 DEactivated. => User location changed: hall -> bedroom
17.11.2024 08:02:17:	Sensor #57 DEactivated. => User is Not moving in bedroom.
17.11.2024 08:02:18:	Sensor #72 Activated. => User is near wardrobe in bedroom.
17.11.2024 08:03:09:	Sensor #72 DEactivated. => User is away from wardrobe in bedroom.
17.11.2024 08:03:09:	Sensor #57 Activated. => User is moving in bedroom.
17.11.2024 08:03:11:	Sensor #59 Activated. => User location change: bedroom -> hall
17.11.2024 08:03:12:	Sensor #60 Activated. => User is moving in hall near bedroom children-room.
17.11.2024 08:03:12:	Sensor #61 Activated. => User is moving in hall near coach.
17.11.2024 08:03:12:	Sensor #57 DEactivated. => User is Not moving in bedroom.
17.11.2024 08:03:12:	Sensor #58 DEactivated. => User is Not moving in bedroom.
17.11.2024 08:03:12:	Sensor #59 DEactivated. => User location changed: bedroom -> hall
17.11.2024 08:03:14:	Actuator #82 DEactivated. => Cofee-machine is turned OFF.
17.11.2024 08:03:16:	Sensor #60 DEactivated. => User is Not moving in hall near bedroom children-room.
17.11.2024 08:03:16:	Sensor #61 DEactivated. => User is Not moving in hall near coach.
17.11.2024 08:03:16:	Sensor #62 Activated. => User is moving in hall near bathroom#1.
17.11.2024 08:03:19:	Sensor #73 Activated. => User in moving in hall near guest-room relax-room.
17.11.2024 08:03:20:	Actuator #71 DEactivated. => Fen in bathroom#1 is turned OFF.
17.11.2024 08:03:08:	Sensor #62 DEactivated. => User is Not moving in hall near bathroom#1.
17.11.2024 08:03:25:	Sensor #73 DEactivated. => User in Not moving in hall near guest-room relax-room.
17.11.2024 08:03:25:	Sensor #74 Activated. => User is moving in hall near kitchen relax-roomю
17.11.2024 08:03:25:	Actuator #83 DEactivated. => Toster-machine is turned OFF.
17.11.2024 08:03:27:	Sensor #75 Activated. => User is moving in hall near bathroom#2 kitchen.
17.11.2024 08:03:29:	Sensor #77 Activated. => User location change: hall -> kitchen
17.11.2024 08:03:29:	Sensor #75 DEactivated. => User is Not moving in hall near bathroom#2 kitchen.
17.11.2024 08:03:29:	Sensor #74 DEactivated. => User is Not moving in hall near kitchen relax-roomю
17.11.2024 08:03:29:	Actuator #79 Activated. => TV on kitchen is turned ON.
17.11.2024 08:03:29:	Sensor #80 Activated. => User is moving in kitchen.
17.11.2024 08:03:29:	Sensor #77 DEactivated. => User location changed: hall -> kitchen
17.11.2024 08:03:57:	Sensor #84 Activated. => User is near refrigerator in kitchen.

Таблиця 4.1 (Закінчення)

17.11.2024 08:04:13:	Sensor #84 DEactivated. => User is away from refrigerator in kitchen.
17.11.2024 08:04:14:	Sensor #85 Activated. => User is near slab refrigerator in kitchen.
17.11.2024 08:04:14:	Actuator #86 Activated. => Oven in kitchen is turned ON.
17.11.2024 08:04:14:	Actuator #87 Activated. => Fen in kitchen is turned ON.
17.11.2024 08:04:24:	Sensor #85 DEactivated. => User is away from slab refrigerator in kitchen.
17.11.2024 08:05:05:	Actuator #86 DEactivated. => Oven in kitchen is turned OFF.
17.11.2024 08:05:07:	Sensor #81 Activated. => User is moving in kitchen.
17.11.2024 08:05:09:	Actuator #87 DEactivated. => Fen in kitchen is turned OFF.
17.11.2024 08:05:09:	Sensor #91 Activated. => User set on chair in kitchen.
17.11.2024 08:07:49:	Sensor #80 Activated. => User is moving in kitchen.
17.11.2024 08:07:49:	Sensor #81 Activated. => User is moving in kitchen.
17.11.2024 08:07:50:	Sensor #91 DEactivated. => User stands up from the chair in kitchen.
17.11.2024 08:07:52:	Sensor #81 DEactivated. => User is Not moving in kitchen.
17.11.2024 08:07:53:	Sensor #78 Activated. => User location change: kitchen -> hall
17.11.2024 08:07:54:	Sensor #75 Activated. => User is moving in hall near bathroom#2 kitchen.
17.11.2024 08:07:54:	Sensor #74 Activated. => User is moving in hall near kitchen relax-roomю
17.11.2024 08:07:54:	Sensor #78 DEactivated. => User location changed: kitchen -> hall
17.11.2024 08:07:54:	Sensor #80 DEactivated. => User is Not moving in kitchen.
17.11.2024 08:07:54:	Actuator #79 DEactivated. => TV on kitchen is turned OFF.
17.11.2024 08:07:56:	Sensor #76 Activated. => User is near main-entrance-door.
17.11.2024 08:07:56:	Sensor #74 DEactivated. => User is Not moving in hall near kitchen relax-roomю
17.11.2024 08:07:57:	Sensor #88 Activated. => User is near small-wardrobe in hall near main-entrance-door.
17.11.2024 08:08:03:	Sensor #88 DEactivated. => User is away from small-wardrobe in hall near main-entrance-door.
17.11.2024 08:08:03:	Sensor #75 DEactivated. => User is Not moving in hall near bathroom#2 kitchen.
17.11.2024 08:08:05:	Sensor #89 Activated. => User is near closet in hall near main-entrance-door.
17.11.2024 08:08:10:	Sensor #89 DEactivated. => User is away from closet in hall near main-entrance-door.
17.11.2024 08:08:12:	Sensor #90 Activated. => User location change: hall -> OUTSIDE
17.11.2024 08:08:13:	Sensor #76 DEactivated. => User is away from main-entrance-door.
17.11.2024 08:08:13:	Sensor #90 DEactivated. => User location changed: hall -> OUTSIDE

4.8. Тестування системи

Для оцінювання якості навчання та функціонування системи «Розумний будинок» використано ключові метрики, рекомендовані стандартом, описаним у ISTQB Certified Tester – AI Testing (CT-AI) [101]. Метрики обчислюються на основі матриці плутанини (confusion matrix), сформованої за результатами моделювання сценаріїв за 14 днів роботи системи з частотою 100 фіксацій на годину (усього 33 600 спостережень).

Основні метрики якості класифікації:

1. Precision (Прецизійність, точність позитивних передбачень):

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4.1)$$

де:

TP — кількість істинно позитивних передбачень (система правильно передбачила дію користувача);

FP — кількість хибно позитивних передбачень (система передбачила дію, яка не відбулася).

2. Recall (Повнота, чутливість):

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4.2)$$

де:

FN — кількість хибно негативних випадків (система не передбачила подію, яка мала місце).

3. F1-score (Гармонічне середнє Precision і Recall) – зведена метрика, що балансує Precision та Recall:

$$F1_{\text{-score}} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.3)$$

4. Accuracy (Загальна точність роботи системи):

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.4)$$

де:

TN — кількість істинно негативних результатів (не передбачено і не відбулось).

Під час проведення дослідження, система передбачила 3 528 подій як позитивні, з яких 2 964 були правильно передбачені, що в контексті даних метрик трактується як істинно позитивні події (TP), а 564 — хибно позитивні передбачення (тобто система передбачила подію, але її не було) (FP):

$$\text{Precision} = \frac{2964}{2964 + 564} = \frac{2964}{3528} \approx 0.84$$

Серед 3 330 реальних позитивних подій (тобто очікуваних активацій пристроїв), система правильно передбачила 2 964 (TP), а 366 пропустила (FN):

$$\text{Recall} = \frac{2964}{2964 + 366} = \frac{2964}{3330} \approx 0.89$$

$$F1_{\text{-score}} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.5)$$

$$F1_{\text{-score}} = 2 \cdot \frac{0.84 \cdot 0.89}{0.84 + 0.89} = 2 \cdot \frac{0.7476}{1.73} \approx 0.94$$

$TP = 2964$

$TN = 29706$ – «негативні події» - ситуації, коли не повинно було бути дії і її не передбачалось

$$\text{Accuracy} = \frac{2964 + 29706}{2964 + 29706 + 564 + 366} = \frac{32670}{33600} \approx 0.97$$

На рис.4.30 представлено гістограму ефективності моделей з варіативними метриками.

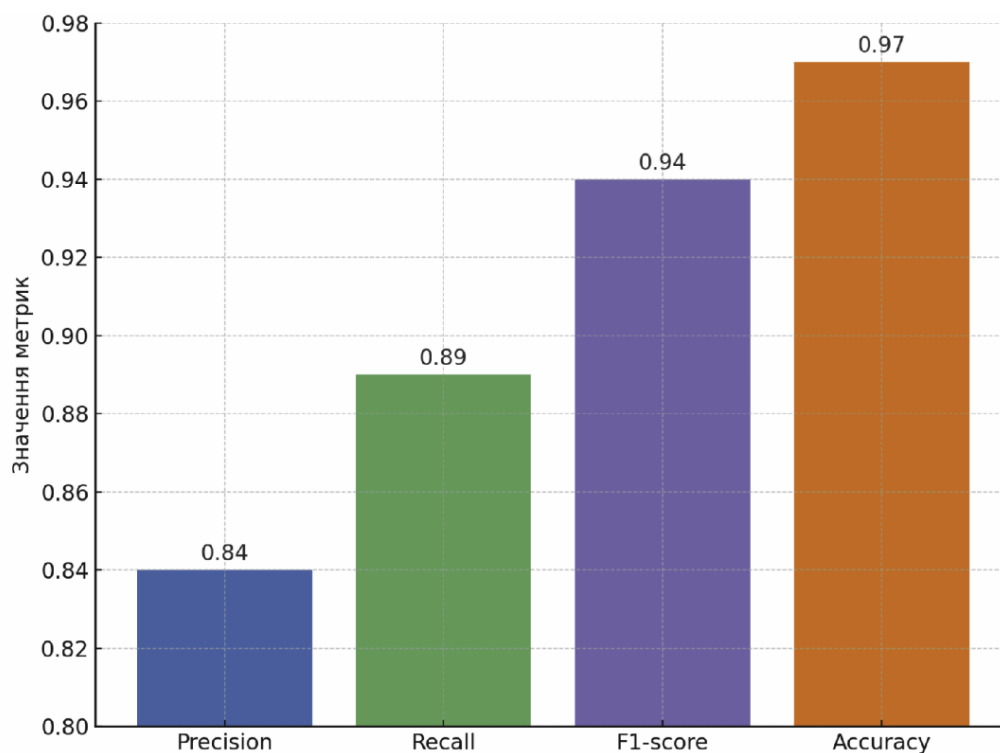


Рис. 4.30 – Ефективність моделі з варіативними метриками

Система демонструє високу якість передбачення з певним балансом між точністю і чутливістю, що дозволяє ефективно керувати пристроями на основі передбачених дій користувача.

Для визначення ефективності системи, проводилось також А/В тестування, що спрямоване на порівняння роботи до та після навчання [101].

На рисунку подано результати порівняльного аналізу ефективності класифікаційної моделі системи «Розумний будинок» до та після етапу навчання. Візуалізовано чотири ключові метрики якості моделі, рекомендовані стандартом ISTQB CT-AI: Precision, Recall, F1-score та Accuracy.

- Precision (точність позитивних передбачень) зросла з 0.72 до 0.84, що свідчить про зменшення кількості хибно позитивних спрацювань (необґрунтованих активацій пристроїв).
- Recall (повнота) підвищилась з 0.78 до 0.89 — система навчилась краще виявляти дії користувача, які реально відбуваються.
- F1-score, як узагальнена метрика якості, зросла з 0.74 до 0.94, демонструючи значне покращення балансу між precision і recall.

- Accuracy (загальна точність класифікації) зросла з 0.76 до 0.97, підтверджуючи загальне зростання продуктивності моделі.

На рис.4.31 представлено гістограму порівняння ефективності моделі, базуючись на результатах її А/В тестування.

Зазначені результати підтверджують, що застосований метод навчання на основі зібраної статистики та моделі Петрі-Маркова з функціональними компонентами є ефективним для адаптації системи РБ до реальної поведінки користувача.

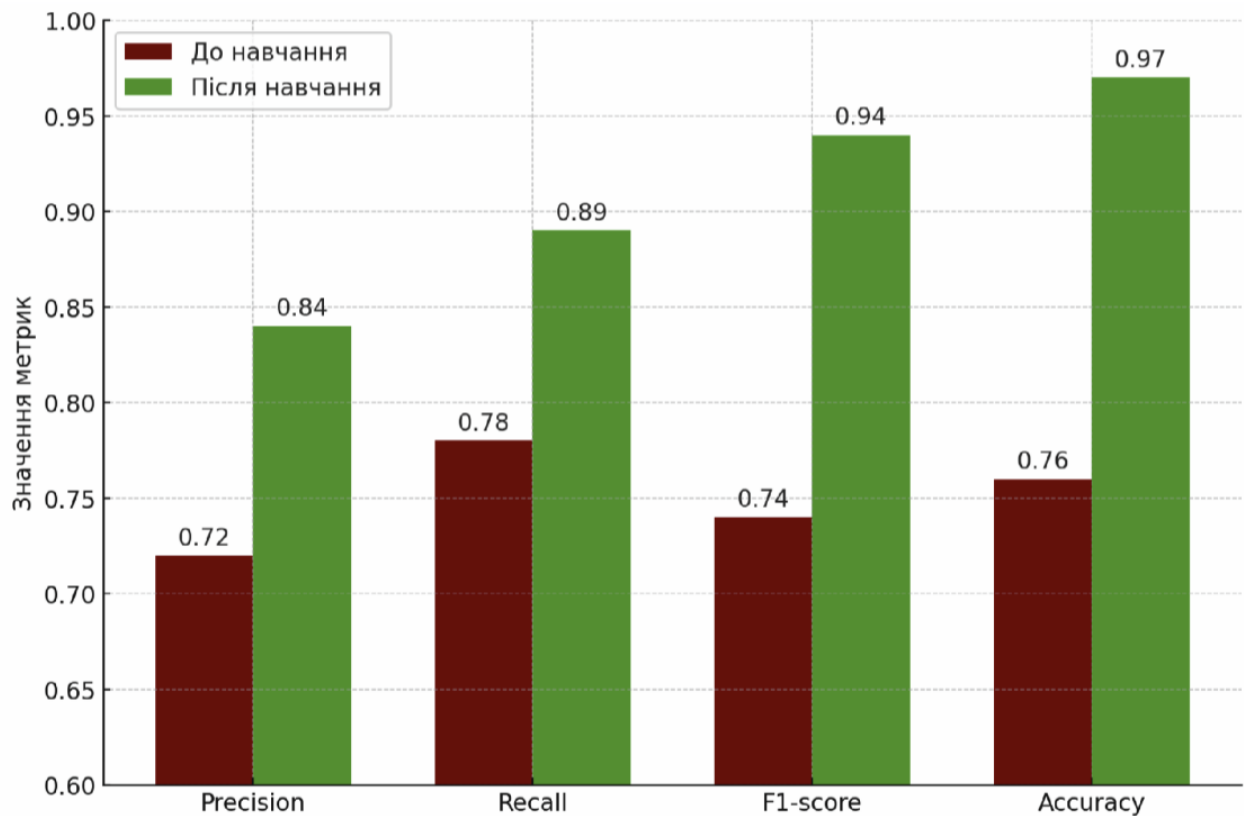


Рис. 4.31 – Порівняння ефективності моделі до та після навчання (А/В тестування)

Висновки до розділу 4

Розроблено інформаційну технологію управління адаптивною системою РБ, побудовану на основі моделей Петрі-Маркова, доповнених функціональними компонентами. Запропонована структура забезпечує опрацювання вхідних даних від сенсорів, оцінювання ймовірностей переходів між станами та ініціацію керуючих дій, базуючись на адаптивному сценарному підході.

Розроблено алгоритм функціонування та структура спеціалізованого мікроконтролера моніторингу та імовірнісного прогнозування зміни станів проєктованої систем РБ, який відповідає за збір інформації від сенсорів і актуаторів в процесі функціонування системи та її взаємодії з користувачем, обробку та збереження результуючої інформації в базу даних [3]. Дані з неї в подальшому зчитуються розробленою універсальною автоматизованою програмною системою адміністрування та моніторингу проєктованої системи РБ. Основне її призначення це - адміністрування та моніторинг процесу функціонування елементів проєктованою системи РБ, а також відображення (моніторинг) функціонування системи на основі інформації з бази даних [8].

Розроблено автоматизовану програмну систему адміністрування та моніторингу (АПСАМ) [8], яка дозволяє графічно візуалізувати розміщення пристроїв у фізичному просторі, контролювати поточний стан системи та здійснювати адміністрування параметрів. АПСАМ підтримує роботу з великою кількістю сценаріїв та користувачів, реалізує імпорт моделей у форматі конфігураційних файлів.

Забезпечено повний цикл адаптації системи до поведінки користувача шляхом реалізації алгоритмів збору статистичної інформації, зберігання її в базі даних та подальшого використання для навчання моделі. Навчання системи здійснюється шляхом заміни апріорних коефіцієнтів ймовірностей переходів на значення, отримані з аналізу реальних сценаріїв взаємодії користувача із системою.

Побудовано структуру автоматизованого програмно-апаратного комплексу та ряд UML-діаграм (прецедентів, станів автомата, послідовності і діяльності), які дозволяють комплексно описати його функціонування.

Побудовано алгоритм функціонування та технічне забезпечення для адаптивної системи РБ.

Розроблено технічне забезпечення контролера адаптивної системи РБ, яке базується на використанні мікроконтролера STM8, що забезпечує низьку ціну пристрою та можливість зміни функціональності мікроконтролера [3].

Розроблено програмне забезпечення адаптивної системи РБ, яке реалізовано на мові Асемблер мікроконтролера STM8, що забезпечує високу швидкодію функціонування пристрою.

Результати моделювання продемонстрували здатність системи до випереджального реагування. Завдяки використанню механізмів обрахунку ймовірностей досягнення фінальних станів, система здатна ініціювати дії (наприклад, вмикання побутових приладів) у відповідь на типовий сценарій користувача ще до його повної реалізації, що підтверджує адаптивність і інтелектуальність функціонування системи.

Якість навчання адаптивної моделі системи було перевірено відповідно до стандарту ISTQB СТ-AI. За результатами тестування протягом 14 днів (33600 спостережень) модель досягла високих значень метрик: Precision – 0.84, Recall – 0.89, F1-score – 0.94, Accuracy – 0.97. Це свідчить про ефективність застосованого підходу та обґрунтовує можливість подальшого впровадження системи в реальні умови.

ЗАГАЛЬНІ ВИСНОВКИ ДО РОБОТИ

В дисертаційній роботі розв'язано актуальну наукову задачу з розроблення методів та засобів управління адаптивною системою розумного будинку, що дає змогу реалізувати механізм адаптації системи до особливостей поведінки користувачів, забезпечивши тим самим високий рівень комфорту.

У першому розділі роботи здійснено комплексний аналіз моделей, методів і засобів інформаційних технологій управління адаптивними системами розумного будинку. На основі проведеного аналізу визначено їх недоліки та переваги і сформульовано мету, об'єкт, предмет та задачі дисертаційного дослідження.

Розроблено моделі для системи управління розумним будинком, які використовують теорію мереж Петрі-Маркова та доповнених функціональними компонентами, що дає змогу реалізувати широкий спектр сценаріїв функціонування та функцію адаптації системи управління розумним будинком до поведінки та вимог його власника.

Розроблено метод навчання моделей для системи управління розумним будинком, який використовує імплентовані режими збору даних (ручний, автоматизований, автоматичний) та етапи опрацювання статистичної інформації та подальше налаштування моделей, що дає змогу здійснити адаптацію розроблених моделей до поведінки користувача. Якість навчання адаптивних моделей системи була протестована відповідно до стандартів, описаних у ISTQB CT-AI, а саме, було визначено, що точність позитивних передбачень складає 0.84%, повнота 0.89%, а загальна точність системи 0.97%, що є досить високим показником і підтверджує ефективність імплементованого підходу.

Розроблено метод синтезу моделей для системи управління розумним будинком, який базується на теорії мереж Петрі-Маркова та розроблених

алгоритмах, що дає змогу забезпечити високий рівень автоматизації процесу побудови адаптивних моделей.

Побудовано модель для системи управління, яка, на відміну від існуючих, використовує теорію стохастичних мереж Петрі та дає змогу врахувати випадкові процеси та стани складових розумного будинку.

Розроблено архітектуру та алгоритми функціонування інформаційної технології управління адаптивною системою розумного будинку. Результати моделювання і тестування довели ефективність розробленої інформаційної технології. Зокрема, система демонструє здатність до випереджувального реагування завдяки механізмам імовірнісного прогнозування. Спроектовано та реалізовано автоматизований програмно-апаратний комплекс, включно зі спеціалізованим мікроконтролером моніторингу та імовірнісного прогнозування, реалізовано автоматизовану програмну систему адміністрування та моніторингу (АПСАМ). Забезпечено повний цикл адаптації системи до поведінки користувача шляхом реалізації механізмів збору та навчання моделей на основі реальних даних.

Розроблено технічне забезпечення контролера адаптивної системи розумного будинку, яке базується на використанні мікроконтролера STM8, що забезпечує низьку ціну пристрою та можливість зміни функціональності мікроконтролера. Окрім того, розроблено програмне забезпечення адаптивної системи розумного будинку, яке реалізовано на мові Асемблер мікроконтролера STM8, що забезпечує високу швидкодію функціонування пристрою.

Результати дисертаційного дослідження впроваджені при виконанні науково-дослідної роботи кафедри автоматизованих систем управління Національного університету «Львівська політехніка» за темами: «Інтелектуальні інформаційні технології багаторівневого управління енергоефективністю регіону» (номер державної реєстрації 0117U004450, роки виконання: 2017-2018); «Методи та засоби нейронечіткого управління групою

мобільних робототехнічних платформ» (номер державної реєстрації: 0123U101688, роки виконання: 2023-2024); «Методи та засоби інтелектуального вимірювання параметрів руху та визначення просторової орієнтації наземних мобільних робототехнічних платформ» (номер державної реєстрації: 0124U000822, роки виконання: 2024-2026). Окрім того, результати дисертаційного дослідження впроваджені в навчальний процес кафедри автоматизованих систем управління Національного університету «Львівська політехніка» для викладання наступних дисциплін: «Основи смарт-технологій і систем» та «Моделювання процесів і смарт-систем». Документи, що підтверджують впровадження результатів дисертаційного дослідження надано у додатку 3.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Береговська Х., Теслюк В. Метод синтезу моделей для адаптивних систем розумного будинку з використанням мереж Петрі-Маркова. *Комп'ютерні технології друкарства*. 2024. Т.51, №1. С. 61–77. doi:[10.32403/2411-9210-2024-1-51-61-77](https://doi.org/10.32403/2411-9210-2024-1-51-61-77)
2. Береговська Х., Теслюк В. Моделі адаптивної системи розумного будинку на базі мереж Петрі та моделей Маркова. *Науковий вісник НЛТУ України*. 2024. Т. 34, № 6. С. 115–124. doi: [10.36930/40340616](https://doi.org/10.36930/40340616)
3. Teslyuk V., Beregovska K., Zerbino D., Teslyuk T., Seneta M. Universal controller for the distributed management in the adaptive smart home systems. *Український журнал інформаційних технологій*. 2024. Вип. 6, т. 2. С. 64–73. doi: [10.23939/ujit2024.02.064](https://doi.org/10.23939/ujit2024.02.064)
4. Береговська Х. В., Машевська М. В., Теслюк В. М. Розроблення моделей систем "Інтелектуальний будинок", побудованих на базі моделей Петрі-Маркова та доповнених функціональними компонентами. *Моделювання та інформаційні технології*. 2017. Вип. 78. С. 175–185. http://nbuv.gov.ua/UJRN/Mtit_2017_78_27
5. Береговська Х. В., Машевська М. В., Зелінський А. Я., Теслюк В. М. Розроблення методу побудови моделей систем "Інтелектуальний будинок", створених на базі моделей Петрі-Маркова та доповнених функціональними компонентами. *Моделювання та інформаційні технології*. 2017. Вип.79. С. 176–182. http://nbuv.gov.ua/UJRN/Mtit_2017_79_26
6. Теслюк В. М., Борецько О. Ю., Сидор А. Р., Береговська Х. В. Модель телекомунікаційної моделі мережі «інтелектуального будинку». *Науковий вісник НЛТУ України*. 2016. Т. 26, № 1. С. 351–357. doi: [10.15421/40260154](https://doi.org/10.15421/40260154)
7. Береговська Х. В., Коваль В. Я., Теслюк В. М. Розроблення спеціалізованого мікроконтролера моніторингу та імовірнісного

- прогнозування зміни станів систем «Розумний будинок» // Моделювання та інформаційні технології : збірник наукових праць. – Вип. 80. – Київ : Інститут моделювання в енергетиці ім. Г. Є. Пухова НАН України, 2017. – С. 121–129. <http://jnas.nbu.gov.ua/article/UJRN-0000855902>
8. Береговська Х. В., Теслюк В. М. Баран М. М. Розроблення автоматизованої програмної системи адміністрування та моніторингу систем "Розумний будинок". *Моделювання та інформаційні технології*. 2017. Вип. 81. С. 114–122. http://nbuv.gov.ua/UJRN/Mtit_2017_81_18
 9. Теслюк В. М., Береговська Х. В. Угрин Л. Є. Розроблення методу навчання моделей систем "розумного будинку" на базі моделей Петрі-Маркова та доповнених функціональними компонентами. *Науковий вісник НЛТУ України*. 2017. Т. 27, № 6. С. 139–144. [doi: 10.15421/40270628](https://doi.org/10.15421/40270628)
 10. Teslyuk V., Beregovska K., Holovatyy A. Analysis of the current state of progress and development of "smart" house systems. *Innovation Technologies in Economy and Society : series of monographs*, monogr. 20. Katowice : Wydawn. Wyższej Szkoły Technicznej w Katowicach, 2018. P. 88–105. <http://www.wydawnictwo.wst.pl/uploads/files/10ce27dda4eab6e1ad0c1e55ca02e9cd.pdf>
 11. Kazarian A., Beregovska K., Teslyuk V. Data analysis model and forms of cloud analytical functions for “smart” house systems. *Information and innovation technologies in economics and administration : monograph* / ed. by O. Chukurna, M. Gawron-Łapuszek. Katowice: Wydawnictwo Wyższej Szkoły Technicznej w Katowicach, 2019. P. 17–27. <http://www.wydawnictwo.wst.pl/uploads/files/2979ee6e305fe7fcd478539e48bb1fcd.pdf>
 12. Beregovska K., Teslyuk V., Beregovskyi V., Kazymyra I., Fabri L. Model and tools of adaptive control of a smart home system. *CEUR Workshop Proceedings*. 2021. Vol. 2917 : Modern machine learning technologies and

- data science workshop: proc. of the 3rd Intern. workshop MoML&T&DS 2021, Lviv, Shatsk, Ukraine, June 5-6, 2021. Vol. I: main conf. P. 263–272 (Scopus) <https://ceur-ws.org/Vol-2917/paper23.pdf>
13. Teslyuk V., Denysyuk P., Kryvinska N., Beregovska K., Teslyuk T. Neural controller for smart house security subsystem. *Procedia Computer Science*. 2019. Vol. 160 : 10th Intern. conf. on emerging ubiquitous systems and pervasive networks EUSPN-2019, 9th Intern. conf. on current and future trends of information and communication technologies in healthcare ICTH-2019)/ Affiliated workshops. P. 394–401 (Scopus, Web of Science) doi: 10.1016/j.procs.2019.11.075
 14. Teslyuk V., Beregovska K., Denysyuk P. Decomposition of models of smart-home systems. *Perspective technologies and methods in MEMS design (MEMSTECH)* : proc. of XIIIth Intern. conf., Polyana, Ukraine, Apr. 20–23, 2017. 2017. P. 22–24. (Scopus, Web of Science) [doi: 10.1109/MEMSTECH.2017.7937524](https://doi.org/10.1109/MEMSTECH.2017.7937524)
 15. Teslyuk V., Denysyuk P., Beregovska K., Mashevskaya M. Method of development of smart-house-systems models based on Petri-Markov nets and extended by functional components. *Computer Science and Information Technologies* : proc. of the XII Intern. sci. and techn. conf. CSIT'2017, Lviv, Ukraine, Sept. 5–8, 2017. 2017. P. 352–355. (Scopus, Web of Science) [doi: 10.1109/STC-CSIT.2017.8098803](https://doi.org/10.1109/STC-CSIT.2017.8098803)
 16. Boreiko O., Teslyuk V., Beregovska K., Mykhailiuk A. Model of telecommunication networks for intelligent building. *The Experience of Designing and Application of CAD Systems in Microelectronics* : proc. of 13th Intern. conf, CADSM 2015, Lviv-Polyana, Ukraine, 24-27 Febr. 2015. P. 349–351. (Scopus, Web of Science) [doi: 10.1109/CADSM.2015.7230873](https://doi.org/10.1109/CADSM.2015.7230873)
 17. Береговська Х. В. Автоматизований програмно-апаратний комплекс для систем «розумний будинок» на основі моделей Петрі-Маркова, доповнених функціональними компонентами // Інтелектуальні

- комп'ютерні системи та мережі : збірник праць IX Науково-практичної конференції молодих вчених і студентів (Тернопіль, 21 травня 2024 р.). – 2024. – С. 51
18. Береговська Х. В., Рутецький Ю. О. Моделі та засоби адаптивної системи розумного будинку // Інтелектуальні комп'ютерні системи та мережі : збірник праць VIII Науково-практичної конференції молодих вчених і студентів, Тернопіль, 5 грудня, 2023. – С. 52–53.
https://ki.wunu.edu.ua/conference/archive/2023_2.pdf
 19. Х. Береговська, В. Теслюк Інтеграція сенсорів і актюаторів в моделі систем «інтелектуальний будинок» // Проблеми та перспективи розвитку економіки і підприємництва та комп'ютерних технологій в Україні : XIII наук.-практ. конф., Львів, 27–31 березня 2017 р. – С. 19–21.
 20. Береговська Х., Теслюк В. Розроблення моделей систем «Інтелектуальний будинок» на базі моделей Маркова // Інформаційне суспільство: тенденції регіонального розвитку : міжнар. наук.-практ. конф. ISRDT-2016, Львів, 20–21 вересня 2016. – С. 8–9.
 21. Ю. Вербіцький, Х. Береговська Система безпеки інтелектуального будинку // V Всеукр. школа-семінар «Сучасні комп'ютерні інформаційні технології», АСІТ'2015, Тернопіль, 22–23 травня 2015 р.
 22. Boreiko O. Video Monitoring System's Module for Smart Home / Boreiko Oleg, Teslyuk Vasyl, Berezsky Oleg, Beregovska Cristina // Proceeding of the 10th International Conference "Perspective Technologies and Methods in MEMS Design" (MEMSTECH'2014), June 22-24, 2014. – Lviv : Lviv Polytechnic Publishing House, 2014. – P. 126-128. <https://orcid.org/0000-0002-1556-8753>
 23. Боре́йко, О. Ю. Модель комп'ютерної мережі інтелектуального будинку з використанням одноплатних комп'ютерів Raspberry Pi / О. Ю. Боре́йко, Х. В. Береговська, В. М. Теслюк // Сучасні комп'ютерні інформаційні технології : матеріали IV Всеукр. шк.-семінару молодих вчен. і студ.

АСІТ'2014 - Тернопіль, 16-17 трав. 2014 р. / відп. за вип. М. П. Дивак. - Тернопіль : ТНЕУ, 2014. - С. 48-51.

<http://dspace.tneu.edu.ua/handle/316497/6719>

24. В. Теслюк, П. Денисюк, Х. Береговська Використання технологій розумного будинку для поліпшення енергетичної ситуації в Україні // XIII міжнар. наук. семінар «Сучасні проблеми інформатики в управлінні, економіці, освіті». – Київ, 2014. – С. 215–219.
<https://nam.kyiv.ua/files/publications/978-966-8406-88-1-konf-compressed.pdf>
25. В. Теслюк, Н. Дудкіна, Х. Береговська Розроблення структури, алгоритму та моделі роботи підсистеми моніторингу інтелектуального будинку // IV Всеукр. школа-семінар АСІТ'2014. – Тернопіль : ТНЕУ
<http://dspace.tneu.edu.ua/handle/316497/6697>
26. W. C. Mann, The State of the Science, in Smart Technology for Aging, Disability and Independence, John Wiley and Sons, Jul. 7, 2005. ISBN: 0-471-69694-3.
27. B. Han, Y. Zahraoui, M. Mubin, S. Mekhilef, M. Seyedmahmoudian, A. Stojcevski, “Home Energy Management Systems: A Review of the Concept, Architecture, and Scheduling Strategies” *IEEE Trans. Software Eng.*, vol.11, IEEE Access, 2023, doi: 10.1109/ACCESS.2023.3248502.
28. Дж. А. Станкович, “Напрямки досліджень Інтернету речей”, *IEEE Internet Things J.*, vol. 1, № 1, с. 3–9, лютий 2014 р.
29. Дж. Губбі, Р. Буйя, С. Марусіч та М. Паланісвами, “Інтернет речей (IoT): бачення, архітектурні елементи та майбутні напрямки”, *Комп’ютерні системи майбутнього покоління*, вип. 29, випуск 7, 2013, с. 1645–1660.
<https://doi.org/10.1016/j.future.2013.01.010>
30. Marikyan Davit, Savvas Papagiannidis, and Eleftherios Alamanos. “A systematic review of the smart home literature: A user perspective.” *Technological Forecasting and Social Change* (2019)
31. І. О. Дужак “Розумний будинок” Автоматизація технологічних і бізнес-

процесів № 13,14/2013, doi:10.15673/2312-3125.13-14/2010.32920

32. Asadullah, Muhammad, and Ahsan Raza. "An overview of home automation systems." 2nd International Conference on Robotics and Artificial Intelligence . IEEE (2016)
33. С. В. Мінухін, О. М. Беседовський, С. В. Знахур, *Методи і моделі проєктування на основі сучасних CASE-засобів : навч. посіб. – Харків : Вид-во ХНЕУ, 2008.*
34. В. І. Слюсар, "Методологія ідентифікації критичних вимог до ОВТ (PDF)," Збірник матеріалів VI міжнародної науково-практичної конференції "Проблеми координації воєнно-технічної та оборонно-промислової політики в Україні. Перспективи розвитку озброєння та військової техніки", Київ, 2018, с. 53–56. doi: 10.13140/RG.2.2.36335.69281.
35. *Model Systems and SSADM*. Model Systems Ltd., 2002. Archived from the original on Apr. 2, 2009. Accessed: Apr. 2, 2009.
36. R. Rock-Evans, *Data Modelling and Process Modelling Using the Most Popular Methods: Covering SSADM, Yourdon, Inforem, Bachman, Information Engineering and 'Activity/Object' (Computer Weekly Professional Series)*, Butterworth-Heinemann, May 12, 2014. ISBN: 978-1483183800.
37. *Зам'ятіна, О. М. Моделювання мереж: навч, посібник / О. М. Зам'ятіна: Томський політехнічний університет. - Томськ: Изд-во Томського політехнічного університету, 2011. - 168 с.*
38. Рад, Б. Я. *Моделювання систем. Практикум: навч, посібник для бакалаврів / Б. Я. Рад, С. А. Яковлєв - 4-е вид. -, 2017. - 295 с.*
39. Рад, Б. Я. *Моделювання систем: підручник для академічного бакалаврату / Б. Я. Рад, С. А. Яковлєв - 7-е вид. -, 2017. -344 с.*
40. І. О. Казак, *Сучасні методи проєктування : конспект лекцій для студентів денної форми навчання освітньо-кваліфікаційного рівня спеціаліст, магістр спеціальності 131 «Прикладна механіка», спеціалізації «Машини і*

- технології пакування» / І. О. Казак ; наук. ред. Д. Е. Сідоров ; рец. А. Р. Степанюк. – Київ : НТУУ «КПІ ім. Ігоря Сікорського», 2017. – 65 с.
41. Акопов. А. С. Імітаційне моделювання: підручник і практикум для академічного бакалаврату / А. С. Акопов. -, 2017. - 389 с.
 42. Боев, В. Д. Імітаційне моделювання систем: навч. посібник для прикладного бакалаврату / В. Д. Боев. -, 2017. - 253 с.
 43. В. Кочкодан, М. Петрина, І. Станьковська, “Застосування машинного навчання та штучного інтелекту в розробці нафтових родовищ,” Scientific Bulletin of Ivano-Frankivsk National Technical University of Oil and Gas. Series: Economics and Management in the Oil and Gas Industry, vol. 1(27), pp. 16–26, Aug. 2023, doi: 10.31471/2409-0948-2023-1(27)-16-26.
 44. А. Мальцев, “Аналіз сучасних досягнень у галузі штучних нейронних мереж, машинного навчання та обчислювального інтелекту,” Information Technology and Society, no. 2, pp. 9–16, Aug. 2022, doi: 10.32689/maup.it.2022.2.9.
 45. C. C. Aggarwal, Neural Networks and Deep Learning: A Textbook, 2nd ed., Springer, Jun. 30, 2023. ISBN: 978-3031296413.
 46. J. A. Lat, Machine Learning Engineering on AWS: Operationalize and Optimize Generative AI Systems and LLMOps Pipelines in Production, Packt Publishing, Sep. 9, 2025. ISBN: 978-1835881095
 47. О. С. Коваленко, Л. М. Добровська, Проектування інформаційних систем: загальні питання теорії проектування ІС : конспект лекцій [Електронний ресурс] : навч. посіб. для студ. спеціальності 122 «Комп’ютерні науки» / НТУУ «КПІ ім. Ігоря Сікорського». – Київ : КПІ ім. Ігоря Сікорського, 2020. – 192 с.
 48. B. R. Fischer, Model-Based Enterprise, 1st ed., CRC Press, Dec. 17, 2024. ISBN: 978-1032067728.
 49. Електроніка і мікропроцесорна техніка / Сенько В.І., Лисенко В.П., Юрченко О.М., Лукін В.Є., Руденський А.А. — К. : «Агроосвіта», 2015.

— 676с.

50. С. Я. Крепич, І. Я. Співак, Якість програмного забезпечення та тестування: базовий курс : навч. посіб. для бакалаврів галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення» / за ред. С. Я. Крепича, І. Я. Співака. – Тернопіль : ФОП Паляниця В. А., 2020. – 478 с.
51. U. Vishniakou, C. Yu, “Simulation of IoT smart home network with decision making based on MajorDoMo platform,” Digital Transformation, vol. 29, no. 1, pp. 64–71, Mar. 2023, doi: 10.35596/1729-7648-2023-29-1-64-71.
52. [Електронний ресурс] : – Режим доступу: <https://home-sapiens.com/>
53. [Електронний ресурс] : – Режим доступу: <https://www.fibaro.com/en/compatibility/>
54. Dong X., Jin S., Lu X. Comparison of Xiaomi Smart Home and Huawei HarmonyOS Business Ecosystem // In: Li X., Yuan C., Ganchev I. (eds) Proceedings of the 2022 International Conference on Business and Policy Studies. CONF-BPS 2022. Applied Economics and Policy Studies. – Singapore: Springer, 2022. – P. 640–648. – doi: 10.1007/978-981-19-5727-7_66.
55. [Електронний ресурс] : – Режим доступу: <https://r2-control-for-crestron.soft112.com/>
56. [Електронний ресурс] : – Режим доступу: <https://easyhome-remote-service.software.informer.com/>
57. Asadi F. Essentials of Arduino™ Boards Programming: Step-by-Step Guide to Master Arduino Boards Hardware and Software. — New York: Apress, 2023. — 332 p. — ISBN 978-1-4842-9599-1. — doi: 10.1007/978-1-4842-9600-4.
58. Halfacree G. The Official Raspberry Pi Beginner's Guide: How to Use Your New Computer. — 5th ed. — Cambridge: Raspberry Pi Press, 2023. — 288 p. — ISBN 978-1-912047-26-0
59. Hayet Lamine and Hafedh Abid, ”Remote control of a domestic equipment

- from an Android application based on Raspberry pi card”, IEEE transaction 15th international conference on Sciences and Techniques of Automatic control & computer engineering - STA'2014, Hammamet, Tunisia, December 21-23, 2014.
60. Siemens AG. LOGO! Manual: Programming and Operating Manual. — Edition 06/2003. — Order No. 6ED1050-1AA00-0BE5. — Document ID: A5E00228550-01. — [Электронный ресурс]. — Режим доступа: https://cache.industry.siemens.com/dl/files/461/16527461/att_82564/v1/Logo_e.pdf.
 61. [Электронный ресурс] : — Режим доступа: <https://infosys.beckhoff.com/english.php?content=../content/1033/tcbacnet/12748426507.html&id=8540850313840101512>
 62. [Электронный ресурс] : — Режим доступа: <https://www.wago.com/global/automation-technology>
 63. [Электронный ресурс] : – Режим доступа: <https://velocio.net/>
 64. [Электронный ресурс] : — Режим доступа: https://ashersecurity.com.ua/kontroller-z-wave-vera-secure-mcvevera_secure
 65. Ralf Joost and Ralf Salomon. “Advantages of fpga-based multiprocessor systems in industrial applications”. In 31st Annual Conference of the IEEE Industrial Electronics Society (IECON 2005). IEEE-I ECON, November 2015.
 66. [Электронный ресурс] : – Режим доступа: <https://aqteck.ua/ua/arhiv-produkciji/novyj-opc-server-owen>
 67. Vimal Nakrani, Mayank Panchal, Dipesh Thakkar, Sanket Pednekar, Prof. Yogita Mane.” A Review: IoT Based Smart Home Automation”. International Journal of Recent Trends in Engineering & Research, March (2017)
 68. Doskaliuk B. The development of geriatric and anti-aging scientific school in Ukraine. Anti-Aging East Eur 2024;3(3):119-123.
 69. Chandramouli, N. A., Natarajan, S., Alharbi, A. H., Kannan, S., Khafaga, D. S., Raju, S. K., Eid, M. M., & El-kenawy, E.-S. M. (2024). Enhanced human

- activity recognition in medical emergencies using a hybrid deep CNN and bi-directional LSTM model with wearable sensors. *Scientific Reports*, 14, Article 30979. doi: 10.1038/s41598-024-82045-y
70. Ullah, R., Asghar, I., Akbar, S., Evans, G., Vermaak, J., Alblwi, A., & Bamaqa, A. (2025). Vision-Based Activity Recognition for Unobtrusive Monitoring of the Elderly in Care Settings. *Technologies*, 13(5), 184. doi: 10.3390/technologies13050184
 71. Huipeng Zhao. "A parallel CNN architecture for sport activity recognition based on minimal movement data." *Scientific Reports*, vol. 14, Article number: 31697 (2024). doi: 10.1038/s41598-024-81733-z.Nature
 72. Md Zia Uddin, Ahmet Soylu. "Human activity recognition using wearable sensors, discriminant analysis, and long short-term memory-based neural structured learning." *Scientific Reports*, vol. 11, Article number: 16455 (2021). doi: 10.1038/s41598-021-95947-y.Nature
 73. Chen, H., Gouin-Vallerand, C., Bouchard, K., Gaboury, S., Couture, M., Bier, N., & Giroux, S. (2024). Enhancing Human Activity Recognition in Smart Homes with Self-Supervised Learning and Self-Attention. *Sensors*, 24(3), 884. doi: 10.3390/s24030884
 74. Diraco, G., Rescio, G., Siciliano, P., & Leone, A. (2023). Review on Human Action Recognition in Smart Living: Sensing Technology, Multimodality, Real-Time Processing, Interoperability, and Resource-Constrained Processing. *Sensors*, 23(11), 5281. doi: 10.3390/s23115281
 75. Bouchabou, D., Nguyen, S. M., Lohr, C., Leduc, B., & Kanellos, I. (2021). A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning. *Sensors*, 21(18), 6037. doi: 10.3390/s21186037
 76. Teslyuk V., Tsmots I., Kryvinska N., Teslyuk T., Opotyak Y., Seneta M., Sydorenko R. Neuro-controller implementation for the embedded control

- system for mini-greenhouse // PeerJ Computer Science. — 2023. — Vol. 9. — Article e1680. — doi: 10.7717/peerj-cs.1680.
77. Liashenko O., Barkovska O., Al-Atroshi C., Datsok O., Liashenko S. Model of the Work of the Neurocontroller to Control Fuzzy Data from the Sensors of the Climate Control Subsystem "Smart House" // International Journal of Advanced Trends in Computer Science and Engineering. — 2019. — Vol. 8, No. 1.2. — P. 70–74. — doi: 10.30534/ijatcse/2019/1281.22019.
 78. Shymkovych V., Telenyk S., Kravets P. Hardware implementation of radial-basis neural networks with Gaussian activation functions on FPGA // Neural Computing and Applications. — 2021. — Vol. 33. — P. 9467–9479. — doi: 10.1007/s00521-021-05706-3
 79. Parisi G.I., Kemker R., Part J.L., Kanan C., Wermter S. Continual lifelong learning with neural networks: A review // Neural Networks. — 2019. — Vol. 113. — P. 54–71. — doi: 10.1016/j.neunet.2019.01.012.
 80. Klabjan D., Zhu X. Neural network retraining for model serving // arXiv preprint arXiv:2004.14203. — 2020. — doi: 10.48550/arXiv.2004.14203.
 81. Lobo J.L., Del Ser J., Bifet A., Kasabov N. Spiking Neural Networks and Online Learning: An Overview and Perspectives // arXiv preprint arXiv:1908.08019. — 2019. — doi: 10.48550/arXiv.1908.08019
 82. Ye, X., Sakurai, K., Nair, N.-K. C., & Wang, K. I.-K. (2024). Machine Learning Techniques for Sensor-Based Human Activity Recognition with Data Heterogeneity—A Review. *Sensors*, 24(24), 7975. doi: 10.3390/s24247975
 83. Siirtola, P., & Röning, J. (2021). Context-aware incremental learning-based method for personalized human activity recognition. *Journal of Ambient Intelligence and Humanized Computing*, 12, 10499–10513. doi: 10.1007/s12652-020-02808-z
 84. Civitarese, G., Presotto, R., & Bettini, C. (2019). Context-driven Active and Incremental Activity Recognition. doi: 10.48550/arXiv.1906.03033

85. Chen, L., Zhang, Y., Miao, S., Zhu, S., Hu, R., Peng, L., & Lv, M. (2021). SALIENCE: An Unsupervised User Adaptation Model for Multiple Wearable Sensors Based Human Activity Recognition. doi:10.48550/arXiv.2108.10213
86. Tao, W., Chen, H., Moniruzzaman, M., Leu, M. C., Yi, Z., & Qin, R. (2021). Attention-Based Sensor Fusion for Human Activity Recognition Using IMU Signals. doi:10.48550/arXiv.2112.11224
87. Liu N.K., Dillon J. Detection of consistency and completeness in expert systems using numerical Petri nets // Joint Artif. Intell. Conf., Sydney, November 2-4, 1987, pp. 119-134.
88. M. Ajmone Marsan, Stochastic Petri Nets: An Elementary Introduction, in Advances in Petri Nets 1989, G. Rozenberg, ed., vol. 424 of Lecture Notes in Computer Science, Springer-Verlag, New York, 1990, pp. 129.
89. R. S. Sreenivas and B. H. Krogh Petri Net Based Models for Condition/Event Systems. In Proceedings of 1991 American Control Conference, vol. 3, pp. 2899-2904, Boston, MA, 1991.
90. Norris, J. R. Markov Chains. Cambridge University Press. 1998. 237 pp.
91. Seneta, E. Non-negative Matrices and Markov Chains. Springer. 2006. – 279pp.
92. Bowman, G. R., Pande, V. S., & Noé, F. (2014). An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation. Springer.
93. Machida, F., & Maciel, P. R. M. (2020). Markov Chains and Petri Nets. Y Handbook of Software Aging and Rejuvenation: Fundamentals, Methods, Applications, and Future Directions (гл. 5). World Scientific Publishing doi: 10.1142/9789811214578_0005
94. Helske, J., Helske, S., Saqr, M., López-Pernas, S., & Murphy, K. (2023). A Modern Approach to Transition Analysis and Process Mining with Markov Models. In Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2023).

95. Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4), 541–580.
96. Воробйова О. М. Технічні засоби автоматизації: навч. посіб. / О. М. Воробйова, Ю. В. Флейта. - Одеса : ОНАЗ ім. О. С. Попова, 2018. - 208 с.
97. M. Hasan, M. Hossain Anik, S. Chowdhury, S. Alam Chowdhury, T. Islam Bilash, and S. Islam, “Low-Cost Appliance Switching Circuit For Discarding Technical Issues Of Microcontroller Controlled Smart Home,” *International Journal of Sensors and Sensor Networks*, vol. 7, no. 2, pp. 16–22, 2019.
98. Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Tiezhen Wang, et al. Tensorflow lite micro: Embedded machine learning for tinyml systems. *Proceedings of Machine Learning and Systems*, 3 : 800-811, 2021
99. Huang HPan YXia WZou XYang DShi LDu HEeckhout LSmaragdakis GLiang KSampson AKim MRossbach C(2025)Simplifying and Accelerating NOR Flash I/O Stack for RAM-Restricted MicrocontrollersProceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 210.1145/3676641.3716272
100. Narkthong, N., Duan, S., Ren, S., & Xu, X. (2024) MicroVSA: An Ultra-Lightweight Vector Symbolic Architecture-based Classifier Library for Always-On Inference on Tiny Microcontrollers. In the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Vol. 2, (pp. 730–745), Association for Computing Machinery, NewYork, NY,USA.<https://doi.org/10.1145/3620665.3640374>
101. ISTQB Certified Tester AI Testing (CT-AI) Syllabus. Version 1.0. International Software Testing Qualifications Board, 2023. [Електронний ресурс]. — Режим доступу: <https://www.istqb.org/downloads/send/83-ct-ai-syllabus/298-istqb-ct-ai-syllabus-v1-0-pdf.html>

ДОДАТКИ

Додаток 1. Таблиці переходу станів вдосконаленої моделі, розробленої на основі ММ, з застосуванням лічильника ітерацій досягнення станів моделі та без застосування такого лічильника

Таблиця Д-1.1.

Таблиця переходу станів вдосконаленої моделі, розробленої на основі ММ, з
застосуванням лічильника ітерацій досягнення станів моделі

	«01»	«02»	«03»	«04»	«05»	«06»	«07»	«08»
«01»	–	1/1	–	–	–	–	–	–
«02»	case iteration of: 1: 2/12 2: 5/12 3: 4/12 4: 1/12 5: 1/12	–	case iteration of: 1: 6/12 2: 1/12 3: 1/12 4: 2/12 5: 1/12	case iteration of: 1: 1/12 2: 1/12 3: 1/12 4: 3/12 5: 1/12	case iteration of: 1: 2/12 2: 4/12 3: 5/12 4: 3/12 5: 2/12	case iteration of: 1: 1/12 2: 1/12 3: 1/12 4: 3/12 5: 7/12	–	–
«03»	–	1/1	–	–	–	–	–	–
«04»	–	1/1	–	–	–	–	–	–
«05»	–	1/1	–	–	–	–	–	–
«06»	–	case iteration of: 1: 1/7 2: 4/7 3: 1/7	–	–	–	–	case iteration of: 1: 3/7 2: 2/7 3: 3/7	case iteration of: 1: 3/7 2: 1/7 3: 3/7
«07»	–	–	–	–	–	case iteration of: 1: 2/5 2: 1/5	–	case iteration of: 1: 3/5 2: 4/5
«08»	–	–	–	–	–	case iteration of: 1: 1/5 2: 1/10	case iteration of: 1: 1/5 2: 1/10	–

Таблиця Д-1.2

Таблиця переходу станів покращеної моделі, розробленої на основі ММ, без застосування лічильника ітерацій

	«01»	«02»	«03»	«04»	«05»	«06»	«07»	«08»
«01»	–	T2:=2/12 T3:=6/12 T4:=1/12 T5:=2/12 T6:=1/12	–	–	–	–	–	–
«02»	T1:=1/1 T2:=1/12 T3:=5/12 T4:=1/12 T5:=4/12 T6:=1/12	–	T1:=1/1 T2:=3/12 T3:=2/12 T4:=2/12 T5:=3/12 T6:=2/12 T7:=1/1	T1:=1/1 T2:=1/12 T3:=2/12 T4:=1/12 T5:=3/12 T6:=5/12 T8:=1/1	T1:=1/1 T2:=2/12 T3:=2/12 T4:=3/12 T5:=2/12 T6:=3/12 T9:=1/1	T1:=1/1 T2:=1/12 T3:=2/12 T4:=4/12 T5:=4/12 T6:=1/12 T10:=1/5 T11:=3/5 T12:=2/5	–	–
«03»	–	T1:=1/1 T2:=3/12 T3:=2/12 T4:=2/12 T5:=3/12 T6:=2/12	–	–	–	–	–	–
«04»	–	T1:=1/1 T2:=1/12 T3:=2/12 T4:=1/12 T5:=3/12 T6:=5/12	–	–	–	–	–	–
«05»	–	T1:=1/1 T2:=2/12 T3:=2/12 T4:=3/12 T5:=2/12 T6:=3/12	–	–	–	–	–	–
«06»	–	T1:=1/1 T2:=1/12 T3:=2/12 T4:=4/12 T5:=4/12 T6:=1/12 T10:=1/5 T11:=3/5 T12:=2/5	–	–	–	–	T1:=1/1 T2:=1/12 T3:=1/12 T4:=5/12 T5:=4/12 T6:=1/12 T13:=2/6 T15:=4/6	T1:=1/1 T2:=1/12 T3:=1/12 T4:=5/12 T5:=4/12 T6:=1/12 T14:=3/8 T16:=2/8

	«01»	«02»	«03»	«04»	«05»	«06»	«07»	«08»
«07»	—	—	—	—	—	T1:=1/1 T2:=1/12 T3:=1/12 T4:=5/12 T5:=4/12 T6:=1/12 T10:=8/12 T11:=3/12 T12:=1/12	—	T16:=1/1 5
«08»	—	—	—	—	—	T1:=1/1 T2:=1/12 T3:=1/12 T4:=5/12 T5:=4/12 T6:=1/12 T10:=8/12 T11:=2/12 T12:=2/12	T13:=2/5 T15:=3/5	—

Додаток 2. Граф досяжності станів розробленої моделі системи РБ, створеної на основі моделей Петрі-Маркова та доповнених функціональними компонентами

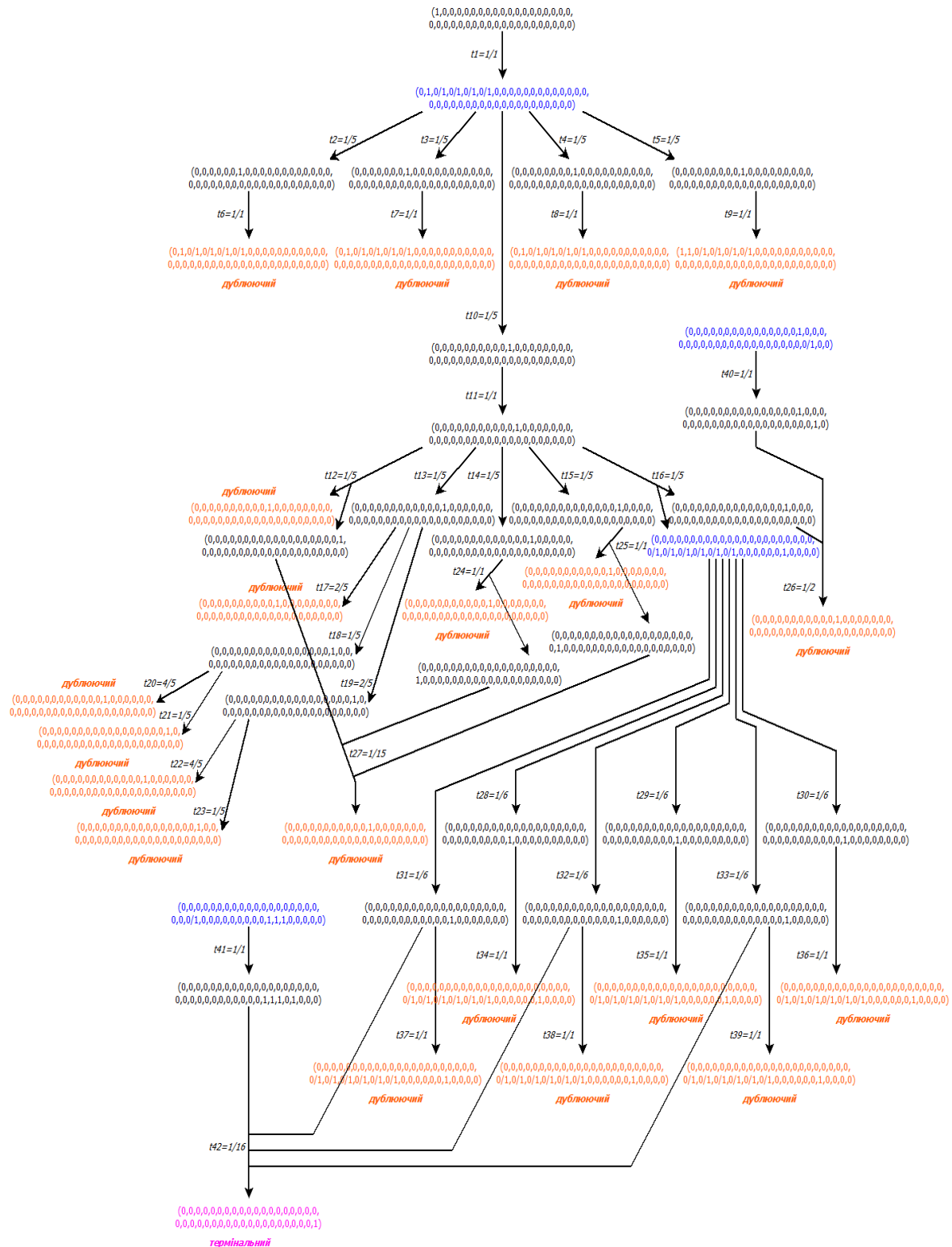


Рис. Д-2.1. Граф досяжності розробленої моделі системи РБ

Додаток 3. Акти впровадження результатів дисертаційного дослідження

ЗАТВЕРДЖУЮ

Проректор з наукової роботи
Національного університету

«Львівська політехніка»

д.т.н., проф. Іван ДЕМІДОВ

«24» 23 2025 р.

АКТ

про використання результатів дисертаційної роботи
Береговської Христини Василівни

«Інформаційна технологія управління адаптивною системою розумного будинку»,
представленої на здобуття наукового ступеня доктора філософії

за спеціальністю 122 «Комп'ютерні науки»,

при виконанні науково-дослідної роботи за темою:

«Інтелектуальні інформаційні технології багаторівневого управління

енергоефективністю регіону»

Комісія у складі голови - начальника НДЧ д.т.н., ст. досл. Романа НЕБЕСНОГО та членів: зав. відділу науково-організаційного супроводу наукових досліджень к.т.н. Галини ЛАЗЬКО, завідувача кафедри автоматизованих систем управління д.т.н. проф. Василя ТЕСЛЮКА та заст. начальника планово-фінансового відділу Ірини ФАСТ – цим актом підтверджують, що результати дисертаційної роботи Береговської Христини Василівни представленої на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» використані при виконанні науково-дослідної роботи, яка виконувалася за рахунок коштів загального фонду державного бюджету за темою: «Інтелектуальні інформаційні технології багаторівневого управління енергоефективністю регіону» (номер державної реєстрації 0117U004450).

Зокрема, Христиною БЕРЕГОВСЬКОЮ було розроблено метод побудови декомпозиційних моделей систем «Розумного будинку» на базі мереж Петрі-Маркова, доповнених функціональними компонентами (Розділ 3. Методи синтезу та навчання моделей систем «Розумний будинок» на базі моделей Петрі-Маркова та доповнених функціональними компонентами).

Голова комісії:

Начальник науково-дослідної частини,
д.т.н., ст. досл.

Роман НЕБЕСНИЙ

Члени комісії:

Зав. відділу науково-організаційного
супроводу наукових досліджень

Галина ЛАЗЬКО

В.о. заступника начальника
планово-фінансового відділу

Ірина ФАСТ

Завідувач кафедри
автоматизованих систем управління, д.т.н., проф.

Василь ТЕСЛЮК



ЗАТВЕРДЖУЮ
Проректор з наукової роботи
Національного університету
«Львівська політехніка»
д.т.н., проф. Іван ДЕМІДОВ
«04» _____ 2025 р.

АКТ
про використання результатів дисертаційної роботи
Береговської Христини Василівни
«Інформаційна технологія управління адаптивною системою розумного будинку»
на основі мереж Петрі-Маркова»,
представленої на здобуття наукового ступеня доктора філософії
за спеціальністю 122 «Комп'ютерні науки»,
при виконанні науково-дослідної роботи за темою:
«Методи та засоби інтелектуального вимірювання параметрів руху та визначення
просторової орієнтації наземних мобільних робототехнічних платформ»

Комісія у складі голови - начальника НДЧ д.т.н., ст. досл. Романа НЕБЕСНОГО та членів: зав. відділу науково-організаційного супроводу наукових досліджень к.т.н. Галини ЛАЗЬКО, завідувача кафедри автоматизованих систем управління д.т.н. проф. Василя ТЕСЛЮКА та заст. начальника планово-фінансового відділу Ірини ФАСТ – цим актом підтверджують, що результати дисертаційної роботи Береговської Христини Василівни представленої на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» використані при виконанні науково-дослідної роботи, яка виконувалася за рахунок коштів загального фонду державного бюджету за темою: «Методи та засоби інтелектуального вимірювання параметрів руху та визначення просторової орієнтації наземних мобільних робототехнічних платформ» (номер державної реєстрації 0124U000822).

Зокрема, Христиною БЕРЕГОВСЬКОЮ було розроблено модель для системи управління, яка, на відміну від існуючих, використовує теорію стохастичних мереж Петрі та дає змогу врахувати випадкові процеси та стани складових розумного будинку (Розділ 2. Моделі адаптивної системи «Розумний будинок» на базі мереж Петрі-Маркова та доповнених функціональними компонентами).

Голова комісії:

Начальник науково-дослідної частини,
д.т.н., ст. досл.

Роман НЕБЕСНИЙ

Члени комісії:

Зав. відділу науково-організаційного
супроводу наукових досліджень

Галина ЛАЗЬКО

В.о. заступника начальника
планово-фінансового відділу

Ірина ФАСТ

Завідувач кафедри
автоматизованих систем управління, д.т.н., проф.

Василь ТЕСЛЮК

ЗАТВЕРДЖУЮ

Проректор з наукової роботи
Національного університету
«Київська політехніка»
д.т.н., проф. Іван ДЕМИДОВ
2025 р.

АКТ

про використання результатів дисертаційної роботи
Береговської Христини Василівни

«Інформаційна технологія управління адаптивною системою розумного будинку»,
представленої на здобуття наукового ступеня доктора філософії
за спеціальністю 122 «Комп'ютерні науки»,
при виконанні науково-дослідної роботи за темою:
«Методи та засоби нейронечіткого управління
групою мобільних робототехнічних платформ»

Комісія у складі голови - начальника НДЧ д.т.н., ст. досл. Романа НЕБЕСНОГО та членів: зав. відділу науково-організаційного супроводу наукових досліджень к.т.н. Галини ЛАЗЬКО, завідувача кафедри автоматизованих систем управління д.т.н. проф. Василя ТЕСЛЮКА та заст. начальника планово-фінансового відділу Ірини ФАСТ – цим актом підтверджують, що результати дисертаційної роботи Береговської Христини Василівни представлена на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» використані при виконанні науково-дослідної роботи, яка виконувалася за рахунок коштів загального фонду державного бюджету за темою: «Методи та засоби нейронечіткого управління групою мобільних робототехнічних платформ» (номер державної реєстрації 0123U101688).

Зокрема, Христиною БЕРЕГОВСЬКОЮ було розроблено модель взаємодії власника та адаптивної системи розумного будинку, яка ґрунтується на теорії мереж Петрі (Розділ 2. Моделі адаптивної системи «Розумний будинок» на базі мереж Петрі-Маркова та доповнених функціональними компонентами), удосконалена модель дає змогу здійснювати декомпозицію системи на довільному рівні, в тому числі на рівні функціональних елементів системи – сенсорів та актуаторів.

Голова комісії:

Начальник науково-дослідної частини,
д.т.н., ст. досл.

Роман НЕБЕСНИЙ

Члени комісії:

Зав. відділу науково-організаційного
супроводу наукових досліджень

Галина ЛАЗЬКО

В.о. заступника начальника
планово-фінансового відділу

Ірина ФАСТ

Завідувач кафедри
автоматизованих систем управління, д.т.н., проф.

Василь ТЕСЛЮК

ЗАТВЕРДЖУЮ

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

к.т.н., доц. Олег ДАВИДЧАК

« 09 » 06 2025 р.



АКТ

**про впровадження в навчальний процес результатів дисертаційної роботи
Береговської Христини Василівни**

Цей акт складено про те, що результати дисертаційної роботи Береговської Христини Василівни представлені на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» на тему «Інформаційна технологія управління адаптивною системою розумного будинку» впроваджено у навчальний процес кафедри автоматизованих систем управління Національного університету «Львівська політехніка».

Впровадження результатів дисертаційної роботи полягає у їхньому використанні при викладанні навчальних дисциплін як окремих розділів лекційних курсів, так і в циклах лабораторних робіт.

Для викладання дисципліни «Основи смарт-технологій і систем» для студентів освітньо-кваліфікаційного рівня «бакалавр», що навчаються за спеціальністю 122 «Комп'ютерні науки», використано наступні результати дисертаційної роботи:

– аналіз моделей та методів розроблення систем «Розумний будинок» (Розділ 1. Аналіз моделей, методів і засобів інформаційних технологій управління системами «Розумний будинок»);

– аналіз програмного забезпечення для роботи з системами «Розумний будинок» (Розділ 1. Аналіз моделей, методів і засобів інформаційних технологій управління системами «Розумний будинок»);

– загальна структура автоматизованого програмно-апаратного комплексу (Розділ 4. Розроблення засобів інформаційної технології адаптивного управління системою «Розумний будинок» на основі моделей Петрі-Маркова і доповнених функціональними компонентами та результати дослідження).

Для викладання дисципліни «Моделювання процесів і смарт-систем» для іноземних студентів освітньо-кваліфікаційного рівня «бакалавр», що навчаються за спеціальністю 122 «Комп'ютерні науки», використано наступні результати дисертаційної роботи:

– метод синтезу моделей систем розумного будинку, створених на базі моделей Петрі-Маркова та доповнених функціональними компонентами (Розділ 3. Методи синтезу та навчання моделей систем «Розумний будинок» на базі моделей Петрі-Маркова та доповнених функціональними компонентами);

– методу навчання моделей систем розумного будинку, створених на базі моделей Петрі-Маркова та доповнених функціональними компонентами (Розділ 3. Методи синтезу та навчання моделей систем «Розумний будинок» на базі моделей Петрі-Маркова та доповнених функціональними компонентами).

Професор кафедри АСУ,
д.т.н., проф.

Завідувач кафедри АСУ,
д.т.н., проф.

В.о. директора ІКНІ,
к.т.н., доц.

Іван ЦМОЦЬ

Василь ТЕСЛЮК

Віктор ХАВАЛКО