

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«05» червня 2024 р.

_____ *Чоломбитько Д. В.*

Кваліфікаційна робота оформлена у відповідності до вимог діючих стандартів та методичних вказівок.

Матеріали кваліфікаційної роботи не містять відомостей, що заборонені для опублікування у відкритих виданнях.

Попередній захист проведено.

Керівник кваліфікаційної роботи



В. В. Безкоровайний

Харківський національний університет радіоелектроніки

Факультет _____ *Комп'ютерних наук*
Кафедра _____ *Системотехніки*
Рівень вищої освіти _____ *другий (магістерський)*
Спеціальність _____ *122 Комп'ютерні науки*
(код і повна назва)
Тип програми _____ *освітньо-наукова*
Освітня програма _____ *Системне проектування*
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ *Чоломбисько Дмитру Володимировичу*
(прізвище, ім'я, по батькові)

1. Тема роботи «*Моделювання процесів розподілу та виконання пакетів робіт при проектуванні технологічних систем*»

затверджена наказом університету від 01 квітня 2024 р. № 259Ст

2. Термін подання студентом роботи до екзаменаційної комісії 11 червня 2024 р.

3. Вихідні дані до роботи Об'єкт дослідження – технологічні системи. Предмет дослідження – процеси розподілу та виконання пакетів робіт в технологічних системах. Предмет розробки – аналітико-імітаційна модель розподілу та виконання пакетів робіт для технологій системного проектування. Функція – оптимізація автоматичного розподілу робіт у технологічних системах. Розмірність задачі: кількість робіт – до 100, кількість виконавців – до 50. Технологічне забезпечення: ІВМ–сумісний персональний комп'ютер.

4. Перелік питань, що потрібно опрацювати в роботі Вступ. Аналіз сучасного стану проблеми розподілу робіт при системному проектуванні технологічних систем. Технологічні системи як об'єкти проектування. Вплив реінжинірингу на розвиток технологічних систем. Проблеми побудови та оптимізації технологічних систем. Математичні моделі та методи задач оптимізації технологічних систем. Технології системного проектування при створенні та вдосконаленні технологічних систем. Постановка мети і задач дослідження. Розробка математичного забезпечення задачі. Постановка задачі розподілу пакетів робіт при системному проектуванні технологічних систем. Аналітична модель розподілу пакетів робіт. Формування критеріїв оптимізації. Критерій оптимізації використання матеріальних ресурсів. Критерій оптимізації часу виконання усього пакету робіт. Критерій оптимізації якості виконання пакету робіт. Згортка критеріїв та побудова цільової функції. Імітаційна модель розподілу пакету робіт. Характеристика систем масового обслуговування (СМО). Застосування СМО при імітації розподілу пакетів робіт. Моделювальний алгоритм розподілу пакетів робіт. Перевірка правильності побудови та точності імітаційної моделі. Розробка програмного забезпечення задачі. Обґрунтування вибору середовища імітаційного моделювання та мови програмування. Обґрунтування вибору СУБД. Архітектура імітаційного програмного продукту. Опис схеми бази даних. Опис внутрішньої структури програмної продукту. Опис клієнтської частини програмної продукту. Результати експериментів. Висновки. Перелік джерел посилання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Кресленики, схеми, плакати та/або комп'ютерні ілюстрації (слайди) на аркушах формату А4, що включаються до тексту пояснювальної записки або складу додатків: схема розподілу пакетів робіт; архітектура програмного засобу; UML-діаграми; часова діаграма обробки подій; екранні форми програми.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата


КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання, аналіз завдання, уточнення плану роботи	22.01.2024	Виконано
2	Аналіз сучасного стану проблеми розподілу та виконання робіт при проектуванні технологічних систем	16.02.2024	Виконано
3	Огляд існуючих математичних моделей і методів розподілу пакетів робіт	04.03.2024	Виконано
4	Розробка математичного забезпечення задачі	15.03.2024	Виконано
5	Розробка програмного забезпечення задачі	03.04.2024	Виконано
6	Проведення експериментальних досліджень	16.04.2024	Виконано
7	Підготовка публікацій за результатами дослідження	22.04.2024	Виконано
8	Оформлення пояснювальної записки	01.05.2024	Виконано
9	Подання закінченої роботи науковому керівникові	28.05.2024	Виконано
10	Усунення зауважень наукового керівника	30.05.2024	Виконано
11	Подання роботи на рецензування	01.06.2024	Виконано
12	Підготовка презентації	07.06.2024	Виконано
13	Попередній захист	08.06.2024	Виконано
14	Подання роботи до екзаменаційної комісії	10.06.2024	Виконано

Дата видачі завдання 22 січня 2024 р.

Студент _____ Чоломбитько Д. В.
(підпис)

Керівник роботи


(підпис)

проф. каф. СТ Безкоровайний В. В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 84 с., 3 табл., 27 рис., 3 додатки, 43 джерела інформації.

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ, ЗАДАЧА ПРО ПРИЗНАЧЕННЯ, ОПТИМІЗАЦІЯ РОЗПОДІЛУ, ПАКЕТ РОБІТ, РОЗПОДІЛ РОБІТ, СИСТЕМНЕ ПРОЄКТУВАННЯ, ТЕХНОЛОГІЧНА СИСТЕМА.

Об'єктом дослідження є технологічні системи (ТС).

Предметом дослідження є процеси розподілу та виконання пакетів робіт в технологічних системах.

Метою дослідження є підвищення ефективності технологій системного проєктування та реінжинірингу технологічних систем за рахунок розробки аналітико-імітаційної моделі процесів розподілу пакетів робіт, що виконуються ними.

Методами дослідження є системний підхід, методи математичного та імітаційного моделювання, теорії багатокритеріального оцінювання та прийняття рішень, структурного аналізу.

У роботі розглянуто поточний стан проблеми моделювання ТС, методи їх проєктування та реінжинірингу, проаналізовано задачу розподілу пакетів робіт, виявлено недосконалість існуючих моделей розподілу пакетів робіт.

Розроблено аналітико-імітаційну модель процесів розподілу та виконання пакетів робіт в ТС, яка дозволяє оптимізувати розподіл стохастичних потоків пакетів робіт за показниками матеріальних (фінансових), часових витрат і якості їх виконання.

Галузями застосування даної моделі може бути технологічні системи на виробничих підприємствах, різноманітні сервісні підприємства.

ABSTRACT

Master's Thesis: 84 pages, 3 tables, 27 figures, 3 appendices, 43 references.

SIMULATION MODELING, ASSIGNMENT PROBLEM, TASKS DISTRIBUTION OPTIMIZATION, TASKS PACKAGE, TASKS DISTRIBUTION, SYSTEMS DESIGN, TECHNOLOGICAL SYSTEM.

The object of research is technological systems (TS).

The subject of research is the processes of distribution and execution of task packages in technological systems.

The purpose of research is to increase the efficiency of system design and reengineering of technological systems. To archive this goal was developed an analytical and simulation model of the distribution processes of task packages that are performed by them.

The research methods are a systematic approach, methods of mathematical and simulation modeling, theories of multi-criteria evaluation and decision-making, system analysis.

The paper examines the current state of the vehicle modeling problem, methods of their design and reengineering, analyzes the problem of distribution of work packages, reveals the imperfection of existing models of distribution of work packages. An analytical and simulation model of the distribution and execution of work packages in the TC has been developed, which allows to optimize the distribution of stochastic flows of work packages according to the indicators of material (financial), time costs and the quality of their execution.

The practical application of this model is technological systems at production enterprises, various service enterprises.

ЗМІСТ

Перелік скорочень	9
Вступ.....	10
1 Аналіз сучасного стану проблеми розподілу робіт при системному проектуванні технологічних систем.....	12
1.1 Технологічні системи як об'єкти проектування	12
1.2 Вплив реінжинірингу на розвиток технологічних систем.....	12
1.3 Проблеми оптимізації технологічних систем	13
1.4 Математичні моделі та методи оптимізації технологічних систем.....	17
1.5 Технології системного проектування при створенні та вдосконаленні технологічних систем	20
1.6 Постановка мети і задач дослідження.....	22
2 Розробка математичного забезпечення задачі	24
2.1 Постановка задачі розподілу пакетів робіт при системному проектуванні технологічних систем	24
2.2 Аналітична модель задачі розподілу пакету робіт	26
2.2.1 Формування критеріїв оптимізації.....	26
2.2.2 Критерій оптимізації використання матеріальних ресурсів.....	28
2.2.3 Критерій оптимізації часу виконання пакету робіт	29
2.2.4 Критерій оптимізації якості виконання пакету робіт.....	31
2.2.5 Згортка критеріїв та побудова цільової функції.....	33
2.3 Імітаційна модель розподілу пакетів робіт	37
2.3.1 Характеристика систем масового обслуговування	38
2.3.2 Застосування СМО при імітації розподілу пакетів робіт	41
2.3.3 Моделювальний алгоритм розподілу пакетів робіт	43
2.3.4 Оцінка точності результатів моделювання	45
3 Розробка програмного забезпечення задачі	48
3.1 Обґрунтування вибору середовища імітаційного моделювання та мови програмування	48
3.2 Обґрунтування вибору СУБД.....	51
3.3 Опис обраної предметної галузі моделювання.....	53
3.4 Архітектура програмного засобу.....	54
3.5 Опис схеми бази даних	55

3.6	Опис внутрішньої структури програмного засобу	58
3.7	Опис клієнтської частини програмного засобу	65
3.8	Проведення імітаційних експериментів	70
	Висновки	79
	Перелік джерел посилання	81
	Додаток А Графічний матеріал кваліфікаційної роботи	85
	Додаток Б Характеристики робіт для імітаційного експерименту	96
	Додаток В Текст програми	98

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних.

СМО – система масового обслуговування.

СУБД – систему управління базами даних.

ТС – технологічна система.

CASE – Computer-aided software engineering.

GPSS – General Purpose Simulation System.

UML – Unified Modeling Language.

ВСТУП

В сучасному світі технології системного проектування відіграють значну роль в підвищенні ефективності технологічних систем (ТС). Наразі, підприємства активно розвиваються в напрямі оптимізації процесів, розробки нових та модифікації старих ТС. Дані дії в комплексі допомагають зменшити витрати, підвищити конкурентоспроможність підприємства та зміцнити його позиції на ринку [1]. Системний підхід до проектування є основним методом проведення як реінжинірингу так і створення нової системи.

При проведенні системного проектування або реінжинірингу ТС часто виникає проблема оптимального розподілу робіт серед виконавців. Також, подібна проблема виникає при самому проектуванні структури майбутньої ТС, коли необхідно вибрати оптимальну конфігурацію обладнання, команд працівників, тощо для виконання деякого переліку замовлень кінцевих користувачів. Розв'язання даних задач є непростим і частіше за все вони вирішуються групою експертів, що значно збільшує час їх виконання та ймовірність появи помилок. Наявність адекватних математичних моделей розподілу стохастичних потоків пакетів робіт серед виконавців дозволила б підвищити ефективність процедур системного проектування та реінжинірингу технологічних систем [2].

Об'єктом дослідження є технологічні системи.

Предметом дослідження є процеси розподілу та виконання пакетів робіт в технологічних системах.

Метою дослідження є підвищення ефективності технологій системного проектування та реінжинірингу технологічних систем за рахунок розробки аналітико-імітаційної моделі процесів розподілу пакетів робіт, що виконуються ними.

Методами дослідження є системний підхід, методи математичного та імітаційного моделювання, теорії багатокритеріального оцінювання та прийняття рішень, структурного аналізу.

Наукова новизна розробленої моделі полягає в розробці аналітико-імітаційної моделі процесів розподілу та виконання пакетів робіт в ТС, яка дозволяє оптимізувати розподіл стохастичних потоків пакетів робіт за показниками матеріальних (фінансових), часових витрат і якості їх виконання.

Практичне використання результатів роботи дозволить скорочувати витрати на створення й експлуатацію ТС, покращувати їх функціональні характеристики за рахунок оптимізації розподілу стохастичних потоків пакетів робіт з урахуванням показників витрат, часу та якості їх виконання.

За матеріалами кваліфікаційної роботи було опубліковано розділ у закордонній колективній монографії, статтю у періодичному фаховому виданні категорії В та тези доповідей у матеріалах чотирьох науково-практичних конференцій.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ РОЗПОДІЛУ РОБІТ ПРИ СИСТЕМНОМУ ПРОЄКТУВАННІ ТЕХНОЛОГІЧНИХ СИСТЕМ

1.1 Технологічні системи як об'єкти проектування

Технологічні системи складаються з сукупності функціонально пов'язаних засобів технологічного оснащення, предметів виробництва та виконавців для виконання в регламентованих умовах виробництва заданих технологічних процесів або операцій [3]. Різноманітні технології виробництва в середині одного підприємства поєднуються в системи та об'єднуються потоками засобів виробництва, при цьому кожен вихідний продукт однієї технології та технологічної системи стає вхідним продуктом для іншої.

Завдяки використанню методів моделювання та оптимізації ТС стають більш гнучкими, швидко адаптуються до змін навколишнього середовища та потреб ринку. Як результат такі системи формують цілісну структуру, що має високу керованість. Кожен компонент такої системи стає направленим на досягання високої економічної ефективності. Проектування та побудова ефективної системи часто є нетривіальною задачею, тому це потребує залучення великої кількості спеціалістів високого класу [2].

1.2 Вплив реінжинірингу на розвиток технологічних систем

Технології реінжинірингу відіграють важливу роль у трансформації застарілих виробничих підприємств, їх технологічних систем та процесів, сприяючи підвищенню їх економічної ефективності та конкурентоспроможності. Процес реінжинірингу включає в себе як впровадження нових технологій, так і кардинальну зміну організаційної структури підприємства, топології її систем [4–5].

Таким чином, під процесом реінжинірингу не варто мати на увазі лише автоматизацію та комп'ютеризацію вже сформованих на підприємстві бізнес-процесів. Подібне спрощене мислення скоріше призведе до нераціонального вкладення часових та матеріальних ресурсів. Нерідко власники чи менеджери підприємств переоцінюють можливості інформаційних технологій до

покращення ефективності технологічних процесів. Реінжиніринг має за мету кардинальну перебудову всієї структури підприємства, впровадження раціоналізації та оптимізації кожного бізнес-процесу згідно з основною метою роботи підприємства [6].

Проведення реінжинірингу часто є циклічною задачею, адже підприємство має адекватно реагувати на постійні зміни у вимогах споживачів в умовах швидких змін зовнішнього середовища. Реінжиніринг рекомендується застосовувати до таких виробничих систем, які вже вичерпали свій ресурс до підвищення ефективності класичними способами менеджменту. Симптомами до проведення реінжинірингу є наявність дуже високого рівня витрат, джерело яких важко пояснюється, втрата клієнтів підприємства через відмову від використання продукції підприємства (наприклад, на підставі високих строків випуску продукції чи її низької якості), стрімке відставання в технологічних процесах від конкурентів при рівних стартових позиціях, нерациональне використання людського потенціалу [5].

З погляду підприємства успішне проведення реінжинірингу його технологічних систем може створити тенденцію до скорочення витрат, збільшення продуктивності праці, підвищення конкурентоспроможності продукції підприємства і, як наслідок, посилення його позицій на ринку.

1.3 Проблеми оптимізації технологічних систем

Методи оптимізації систем зазвичай засновані на загальній схемі, відповідно до якої, рішення завдання створення максимально ефективної системи може бути досягнуто через поетапний системно-алгоритмічний пошук рішень. Спочатку здійснюється пошук та реалізація стратегій, спрямованих на досягнення комплексної оптимальності, включаючи визначення базових критеріїв, розробку методик синтезу, а також аналіз можливостей. Потім відбувається вибір загальних та часткових критеріїв оптимальності, а також синтез всього спектра можливих рішень щодо їх структури та параметрів. Виходячи з цього спектра, шляхом порівняння значень критеріїв оптимальності для різних рішень визначається оптимальне рішення.

Проектування і реінжиніринг ТС завжди супроводжується вибором оптимальної конфігурації її структурних компонентів. Одними із таких

компонентів є структура виробничого обладнання, схема постачальників сировини, працівників різних спеціалізацій та кваліфікацій, тощо. Важливою задачею проектування є формування такої конфігурації, при якій виробничий компонент має максимально можливе завантаження (утилізацію) і створює виробничий продукт за найменший можливий час, ефективно використовуючи матеріальні ресурси при заданому рівні якості. Роль правильної топології технологічної системи також є дуже високою і сприяє зменшенню часових та матеріальних витрат на перехід від одного технологічного процесу до іншого [7].

Структуру системи для спрощення її розуміння відображають методом поділу системи на менші підсистеми та окремі елементи, які пов'язані між собою різноманітними зв'язками. Вона може бути розглядана із різних ракурсів в залежності від мети створення. Структури спрощують розкриття невизначеності складних систем. Найбільшої популярності набули деревоподібні ієрархічні структури, у яких встановлено порядок підпорядкування нижчих за позицією блоків вищим в ієрархії. Наприклад, якщо на підприємстві є декілька окремих пов'язаних виробництв з власним набором технологічних процесів, так ієрархічна структура може мати вигляд, як на рисунку 1.1 [8, с. 44].

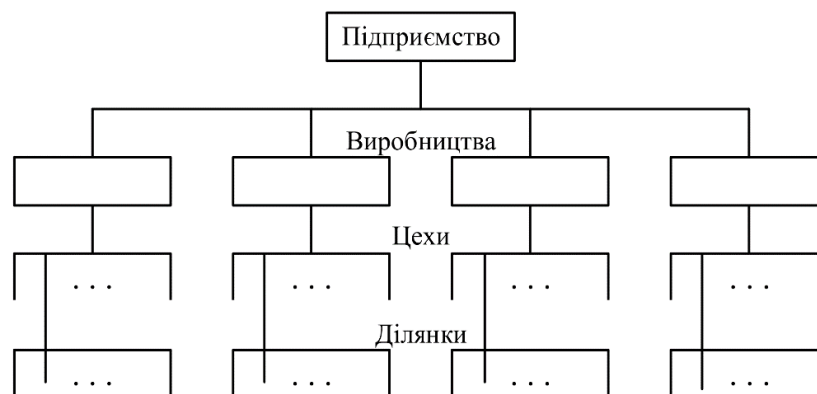


Рисунок 1.1 – Виробнича ієрархічна структура [8, с. 44]

Сучасні підприємства існують в жорстких ринкових умовах, тому однією із найкритичніших проблем є постійне вдосконалення конфігурації та структури підприємства. Через невідповідність ринкового, виробничо-господарського й управлінського потенціалу підприємств мінливим умовам зовнішнього середовища їм набагато важче вижити, що призводить до їх кризи. Багато підприємств, в тому числі і вітчизняних, опираються на структурну оптимізацію

як на один із способів запобігання криз та встановлення стійкості до зовнішніх чинників.

На даний час вже остаточно доведено, що виходячи із постійних змін на ринку із зв'язку з кризами, пандеміями та війнами жодне підприємство не може стабільно існувати незмінним на протязі багатьох років. Правильна структурна конфігурація ТС також повинна забезпечувати високий потенціал до адаптації та надійності системи. Висока степінь адаптованості системи характеризується сильною реакцією виробничої системи до негативних змін внутрішнього чи зовнішнього середовища системи [9].

У зв'язку з цим, при проєктуванні нових технологічних систем та реінжинірингу старих, віддається перевага системам, які є достатньо гнучкими та адаптованими. Їх ключовою особливістю є вміння реагувати на несподівані події завдяки закладеній у них внутрішній міцності та оперативності реагування на зміни. Дана оперативність може бути забезпечена як наявністю висококваліфікованого персоналу для швидкого переналаштування обладнання, так і наявністю повністю автоматизованої лінії, яка може переналаштовувати себе автоматично, згідно із новими вимогами виробництва, появи нової технології, вдосконалення старої тощо [10].

Також варто звернути увагу на надійність системи, яка опирається на надійність кожного окремого її компоненту, наприклад такого як обладнання. Мала кількість стадій виробництва та мала кількість операцій у технології також підвищує показник надійності. Варто зазначити, що не завжди об'єкти проєктування системою є зосередженими системами, в яких легко скоротити кількість виробничих бізнес-процесів та зв'язків між їх компонентами [11].

Враховуючи достатньо високу складність та комплексність сучасних технологічних систем на виробництві, їх у багатьох випадках доцільно розглядати як територіально чи просторово розподілені об'єкти. Оптимізація варіантів побудови таких систем здійснюється з використанням інтелектуальних автоматизованих засобів проєктування. В даній роботі буде розглянуто одну з задач формування оптимальної структури (та топології) розподіленої технологічної системи засобами автоматизації.

З математичної точки зору проблему оптимізації ТС великого масштабу можна розглядати як метaproblem *MetaTask*, яка має декілька ієрархічних рівнів l (рис. 1.2) [12]:

$$MetaTask = \{Task^l\}, Task^l = \{Task_i^l\}, l = \overline{1, n_l}, i = \overline{1, i_l}, \quad (1.1)$$

де n_l – кількість рівнів опису проблеми;

i_l – кількість задач на l -му рівні.

При цьому кожне завдання на даній стадії можна представити як конвертор даних:

$$Task_i^l: In_i^l \rightarrow Out_i^l, l = \overline{1, n_l}, i = \overline{1, i_l}, \quad (1.2)$$

де In_i^l, Out_i^l – вхідні та вихідні дані i -ї задачі на l -му рівні.

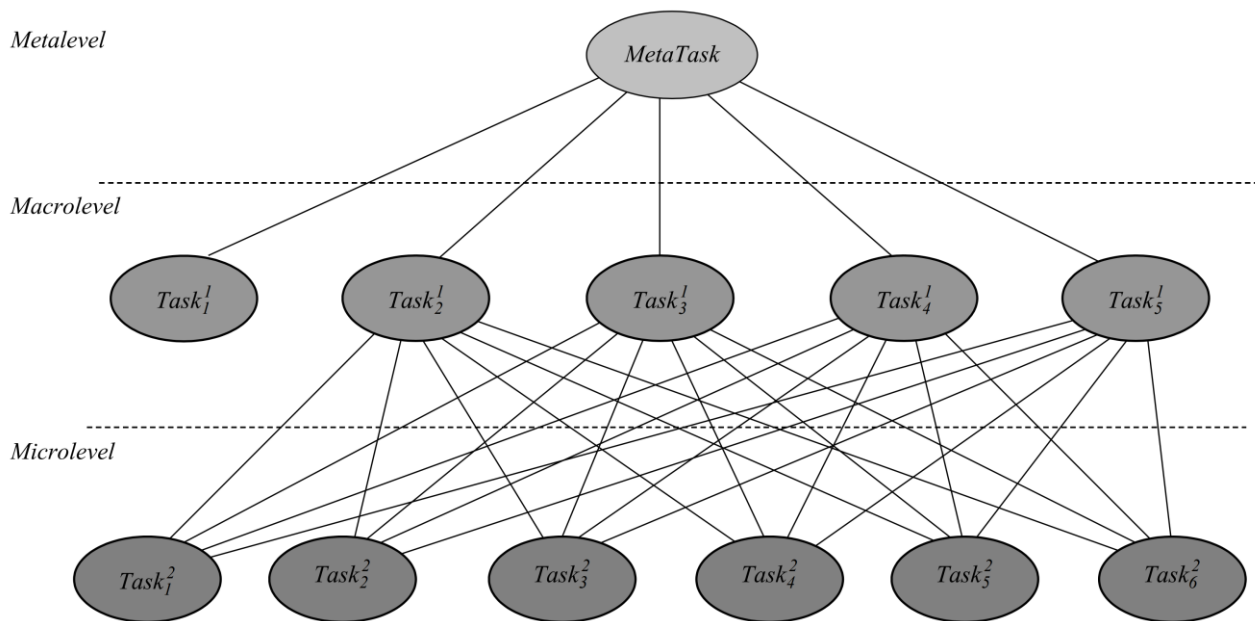


Рисунок 1.2 – Схема декомпозиції проблеми оптимізації ТС [12]

При більш детальному розгляді декомпозиції першого рівня ($l=1$) мета-проблеми *MetaTask* виділимо окремі підзадачі проектування:

- $Task_1^1$: формування вимог до створюваної технологічної системи та створення технічного завдання на її оптимізацію (визначення цілей створення системи, виконуваних нею функцій та обмежень навколишнього середовища);

- $Task_2^1$: системне проектування технологічної системи (визначення структурних та топологічних параметрів системи, проведення оптимізації за

критеріями її ефективності);

– $Task_3^1$: планування розвитку системи (визначення змін у системі для заданих моментів часу);

– $Task_4^1$: адаптація системи (визначення незначущих змін системи при зміні умов функціонування);

– $Task_5^1$: реінжиніринг технологічної системи;

В даній роботі розглядається розробка математичної моделі розподілу та виконання пакетів робіт в ТС, що визначатиме структурні параметри в задачах їх системного проєктування та реінжинірингу ($Task_2^1$ та $Task_5^1$).

1.4 Математичні моделі та методи оптимізації технологічних систем

При ґрунтовному аналізі існуючих способів формування конфігурацій технологічних систем було виявлено декілька, які підходять під суть завдання: задача про призначення, метод ієрархій, методи календарного планування, економічна задача розміщення підприємств та задача зіставлення розкладів. Дані задачі належать до різноманітних класів задач дослідження операцій [9].

Задача про призначення є класичною задачею лінійного програмування. В загальному випадку під задачею лінійного програмування розуміють задачу знаходження мінімуму чи максимуму лінійної функції від n змінних на множині розв'язків системи лінійних нерівностей або лінійних рівнянь. Цю функцію називають цільовою функцією, вона моделює поставлену в задачі мету [13].

Формулюється задача про призначення наступним чином: нехай задано n робіт та n виконавців. Кожен виконавець може виконати кожену роботу, але з різним показником ціни a_{ij} . В якості ціни може бути задано як фактичну платню робітнику за виконання завдання, так і різноманітні суміжні величини, такі як показники часу, витрачених матеріалів, якості тощо. Метою задачі є знаходження оптимального розподілу виконавців по усіх роботах, при цьому кожен виконавець має виконати тільки одну роботу (рис. 1.3) [14-18].

Математична модель класичної задачі про призначення з кількістю робіт, що дорівнює кількості виконавців з матрицею цін може бути подана у вигляді системи (1.3) [14].

		Роботи				
		1	2	...	n	
Робітники	1	a_{11}	a_{12}	...	a_{1n}	1
	2	a_{21}	a_{22}	...	a_{2n}	1

	n	a_{n1}	a_{n2}	...	a_{nn}	1
		1	1	...	1	

Рисунок 1.3 – Табличне подання задачі про призначення

$$\left\{ \begin{array}{l} f(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_{ij} \rightarrow \min_x, \\ a_{ij} > 0, i, j = \overline{1, n}, \\ \sum_{i=1}^n x_{ij} = 1, j = \overline{1, n}; \sum_{j=1}^n x_{ij} = 1, i = \overline{1, n}; x_{ij} \in \{0, 1\}, i, j = \overline{1, n}, \end{array} \right. \quad (1.3)$$

де n – кількість робіт та виконавців;

a_{ij} – витрати (матеріальні, часові) на виконання i -ї роботи j -м виконавцем;

$x = [x_{ij}]$, $i, j = \overline{1, n}$ – матриця призначення i -ї роботи j -му виконавцю (елемент матриці $x_{ij} = 1$ якщо робота призначена, та $x_{ij} = 0$ – якщо ні).

Задача про призначення є окремим спрощеним випадком транспортної задачі, при якому робітники виступають пунктами відправлення, а роботи – пунктами призначення, а всі величини попиту та пропозиції дорівнюють одиниці. Не беручи до уваги методи розв’язання транспортної задачі, частковим випадком якої є задача про призначення, існує ще низка методів, які спрощують розв’язання даної задачі.

Суть угорського алгоритму полягає у покроковому спрощенні матриці призначень за допомогою віднімання мінімальних елементів рядків та стовпців та знаходженні нульових елементів. На місці утворення нульових елементів і є оптимальний розв’язок задачі. Другим методом є метод гілок та меж, який опирається на представлення розподілу робіт у формі множин та проведені її комбінаторної оптимізації.

Дані методи розв'язку справедливо працюють тільки для правильних матриць розподілу, у яких кількість робіт дорівнює кількості виконавців. У випадку не співпадіння, матрицю можна доповнити фіктивними виконавцями чи роботами, які необхідно відкинути після розв'язку задачі [19].

Однією із похідних із задачі про призначення є квадратична задача про призначення. Вона являє собою її комбінацію з задачею про розміщення та транспортною задачею: припустимо, що є виробничі цехи, які можуть бути розміщені на різних ділянках підприємства. Відомі відстані між кожною ділянкою, необхідний об'єм перевезень між цехами, та вартість перевезення одиниці ресурсу. Необхідно знайти такий розподіл, при якому вартість перевезень буде мінімальною. В основі такої задачі лежить квадратична цільова функція [20].

Іншим методом який часто застосовується при вирішенні багатокритеріальної задачі структурного синтезу є метод аналізу ієрархій. Він ідеально підходить під вирішення складних проблем звичним для людини способом. Це пояснюється тим що багато реальних завдань можна вирішити та представити у вигляді ієрархічної структури підлеглих рішень або альтернатив. При побудові структури складних виробничих систем необхідно сформувавши базу альтернатив та встановити для кожної альтернативи множину значень їх параметрів.

При загальному розв'язанні задач даним методом необхідно встановити три речі: загальну мету тобто ціль розв'язання задачі, критерії за якими будуть оптимізуватися альтернативи та встановити самі альтернативи. Критерії варто сформувавши на основі чисельних параметрів альтернатив. Якщо параметри альтернативи задано у вигляді якісної шкали, необхідно перевести її до кількісної, наприклад методом експертної оцінки .

Для формування вхідних даних розв'язувану задачу необхідно структурувати у вигляді ієрархічної структури з декількома рівнями цільових критеріїв альтернатив. Необхідно сформувавши матрицю попарних порівнянь кожного із критеріїв і обчислити оціночний коефіцієнт пріоритету за допомогою шкали Томаса Сааті (рис. 1.4). За допомогою значень даної матриці знаходиться власні вектори кожного критерію та його вага.

На наступному етапі за схожим способом знаходиться матриця попарних порівнянь кожної із альтернатив за кожним критерієм. Це дає можливість

розрахувати власні вектори кожної альтернативи за кожним із критеріїв та вагу альтернативи в критерії. Заключним етапом є розрахунок відносної ваги кожної альтернативи, їх ранжування та вибір найкращої [21].

	A	B	C	D
A	1	5	6	7
B	1/5	1	4	6
C	1/6	1/4	1	4
D	1/7	1/6	1/4	1

Рисунок 1.4 – Схема таблиці попарних порівнянь пріоритетів критеріїв

Схожим класом задач є задачі календарного планування, які використовуються для моделювання випуску продукції деякого об'єму, яка може виготовлятися за декілька стадій, при мінімальній сумарній вартості та витраченого часу. Модифікаціями може бути додавання показників витрат на зберігання та збут продукції чи її часове псування (для продовольчих чи активних хімічних компонентів). Моделі економічної задачі оптимізації розміщення виробництва допомагають при розширенні та реінжинірингу старих підприємств. Вони будуються на основі транспортної задачі та може враховувати декілька типів продукції та сировини. Розв'язання даних задач можливе за допомогою традиційних методів лінійного програмування (наприклад симплекс-методу) [15-16].

Задача зіставлення розкладів є корисною при плануванні використання обладнання та формування змін працівників, що не є прямою вирішуваною задачею при системному проектуванні. Метод покрокового конструювання є основним при розв'язанні цих задач, що дає змогу робити оптимізацію за критеріями оптимального навантаження виконавця протягом дня і робочого тижня, а також його рівномірний розподіл серед різних типів виконавців.

1.5 Технології системного проектування при створенні та вдосконаленні технологічних систем

Зараз спостерігається стрімкий розвиток інформаційних технологій системного проектування, включаючи реалізацію методів моделювання,

створення інформаційних систем різних типів для ТС. Аналіз запитів від бізнесу показує суттєвий попит на виробничі системи, системи управління та прогнозування. Це призвело до необхідності удосконалення методології створення ТС, зокрема, системного підходу та системного проектування.

Системний підхід є незамінним при проектуванні технологічних систем різної складності. Він вважається одним із базисних методів їх створення та без його застосування немислима успішна професійна діяльність практично в будь-якій сфері. Володіння системним аналізом, системним моделюванням і конструюванням, системною практичною діяльністю є найбільшою професійною цінністю сучасного спеціаліста з проектування [1].

На початку розвитку проектування особливо гостро постала проблема постійно зростаючої складності та масштабності підприємств. Враховуючи даний факт, було достатньо складно спроектувати їх одразу в найбільш ефективний спосіб. Але розвиток системного мислення як методу пізнання призвів до значного спрощення розгляду складних реальних виробничих об'єктів, опису їх як абстрактної сукупності взаємодіючих компонентів. Та все ж треба мати на увазі, що будь-який проєкт ТС є ідеалізованою моделлю реального виробничого об'єкту, що обмежена ступенем її деталізації. Подібні моделі подаються у вигляді схем, графіків та документації, які детально описують якою вона має ТС [1, 8].

Системне проектування є прямим практичним застосуванням системного підходу до побудови складних об'єктів. Важливим завданням при системному проектуванні є встановлення мети створення системи, вона повинна коротко але повністю описувати основну суть її існування. На наступних кроках необхідно встановити базові її елементи, зв'язки між ними, знайти місце системи серед оточення, встановити її зв'язки з зовнішніми системами.

Головна суть системного проектування полягає у знаходженні оптимальної конфігурації ТС, тобто структури, типів елементів, їх топології та технології функціонування. Основний метод, який допомагає при розробці систем є декомпозиція. Декомпозиція виконує роль абстрагування, тобто приховування несуттєвих властивостей, зв'язків та відношень між компонентами системи на певному її рівні. По мірі збільшення рівня декомпозиції, деталізація системи зростає. Головне при декомпозиції зберегти ключові риси компонента на верхніх її рівнях.

Сучасні світові тенденції полягають в тому, що вимоги до систем стають все жорсткішими, що в свою чергу робить їх достатньо комплексними. Великі проекти по побудові ТС часто характеризуються складністю опису окремих компонентів, відсутністю прямих аналогів, наявності тісно або хаотично поєднаних у взаємодії компонентів, необхідністю в інтеграції вже існуючих систем в єдине ціле, функціонуванням в умовах невизначеності чинників навколишнього середовища.

Задачею системного проектування є складання такого опису ТС, якого повністю достатньо для її створення при заданих умовах та обмеженнях, на основі первинного опису й алгоритму її функціонування. При цьому варто забезпечити її структурну та функціональну гнучкість для того, щоб була можливість подальшого еволюційного розвитку системи та її адаптації до мінливих вимог зовнішнього середовища.

1.6 Постановка мети і задач дослідження

За результатами огляду й аналізу сучасного стану проблеми розподілу робіт при системному проектуванні виробничих систем було виявлено, що ефективність проектування багато в чому виражається в побудові оптимальної структури виробничої системи. Існуючі методи не можуть в повній мірі задовольнити потреби у створенні оптимально сформованих сучасних підприємств.

Недоліками класичної задачі про призначення є неможливість застосування в умовах динаміки та невизначеності вхідного потоку робіт, неможливість врахування декількох критеріїв оптимізації та наявності специфічних бізнес-правил чи пріоритетності при виконанні робіт.

Проблеми при використанні методу аналізу ієрархій проявляються у вигляді однієї можливої альтернативи як результату при розв'язанні задачі, що унеможливорює призначення декількох можливих альтернатив за один прогін алгоритму. Також його слабкістю є необхідність проведення великої кількості експертних оцінок, що зменшує його автоматизованість.

Інші класи задач, такі як методи календарного планування, зіставлення розкладів та економічної задачі оптимізації розміщення виробництва не можуть прямо задовольнити потреби системного проектування виробничих підприємств

через свою специфічність. Даний клас задач виділяється складністю до формалізації вимог до створюваного календарного плану та розкладу, множинністю цих вимог та проблемою оцінки якості отриманого розв'язку.

Виходячи із вищесказаного було прийнято рішення про розробку аналітико-імітаційної математичної моделі розподілу пакетів робіт при системному проектуванні виробничих систем, яка може працювати в умовах стохастичності вхідного потоку пакетів, дає змогу провести багатокритеріальну оптимізацію та підтримує можливість внесення специфічних бізнес-правил до моделі.

Метою дослідження є підвищення ефективності технологій системного проектування та реінжинірингу технологічних систем за рахунок розробки аналітико-імітаційної моделі процесів розподілу пакетів робіт, що виконуються ними.

Для досягання поставленої мети необхідно:

- визначити базовий метод для розв'язання задачі розподілу пакетів робіт;
- розробити аналітичну модель багатокритеріальної задачі розподілу пакетів робіт;
- розробити імітаційну статистичну модель задачі розподілу стохастичного потоку пакетів робіт;
- розробити програмне забезпечення аналітико-імітаційної моделі процесів розподілу та виконання пакетів робіт;
- провести експериментальне дослідження працездатності розробленої аналітико-імітаційної моделі.

Об'єктом дослідження є технологічні системи.

Предметом дослідження є процеси розподілу та виконання пакетів робіт в технологічних системах.

2 РОЗРОБКА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ

2.1 Постановка задачі розподілу пакетів робіт при системному проектуванні технологічних систем

Структура ТС характеризується топологією, параметрами і технологією функціонування її елементів. При оптимізації технологічної системи необхідно здійснювати оцінку її функціональних і вартісних параметрів у процесі виконання нею заданої множини робіт. Однією з проблем при цьому є оптимальне призначення виконавців на роботи з врахуванням їх кваліфікації та поточного завантаження. Також, при побудові та реінжинірингу виробничих підприємств популярною задачею є формування груп обладнання та команд працівників із розрахунку на замовлення від клієнтів підприємства. Такі замовлення також є пакетами виконуваних робіт, які визначаються характером створюваної продукції.

Вхідною інформацією є дані про необхідні роботи ТС підприємства, про виконавців робіт, про її структурні особливості та специфіку бізнес-процесів.

Вихідною інформацією є дані про призначення виконавців на кожен з робіт пакетів, які надходять протягом певного періоду, а також час, якість і витрати на виконання всього потоку пакетів робіт.

Необхідно розробити аналітико-імітаційну модель розподілу стохастичного потоку пакетів робіт.

Розглянемо більш детально процес надходження, розподілу та виконання пакетів робіт. До ТС у випадкові моменти часу надходять пакети робіт, які необхідно розподілити серед наявних у системі виконавців. При цьому роботи та виконавці мають спеціалізацію: не кожен виконавець може виконати кожену роботу. Система має чітко визначену кількість виконавців r . Список робіт кожного пакету є випадковим.

Виконавцем у виробничій ТС може бути обладнання, працівник або група працівників (бригада). Для кожного виконавця визначені спеціалізація та продуктивність та рівень кваліфікації. Кваліфікація впливає на якість, час виконання робіт пакету, а також на витрати матеріальних, енергетичних чи інших видів ресурсів.

Для оцінки кваліфікації пропонується використати числовий параметр

$\beta \in [0.5; 1.5]$. Його буде використано як фактор впливу на часові та якісні параметри розподілу при їх розрахунку у моделі.

Значення параметра кваліфікації необхідно підібрати емпірично окремо для кожної ТС. Наприклад можна вважати, що в середньому випадку виконавець низької кваліфікації виконує роботу приблизно в півтора рази довше, ніж виконавець середньої кваліфікації, а виконавець високої кваліфікації може виконати роботу швидше приблизно на половину.

Спеціалізація виконавця $k \in K$ відповідає типу задач, які він може виконувати. Для автоматизованих виконавців на виробничих підприємств, таких як роботизовані конвеєрні машини, поняття спеціалізації є особливо критичним.

Нехай у технологічній системі є r виконавців робіт, які характеризуються такими параметрами:

- $l \in \{0, 1\}$: показник того, чи зайнятий на даний час виконавець;
- β : числовий формалізований показник кваліфікації виконавця;
- $k \in K$: тип спеціалізації виконавця.

Позначимо пакет який надходить на виконання у систему як множину з роботами, які можуть повторюватися:

$$Package_c = \{t_1, t_2, \dots, t_i\}, \quad i = \overline{1, n}, \quad (2.1)$$

де c – порядковий номер пакету, який надійшов до системи;

t_i – окрема робота із пакету робіт;

n – кількість робіт в пакеті.

Кожна робота із пакету характеризується такими параметрами:

– c : показник номінальних матеріальних витрат на виконання роботи t_i (дані витрати можуть бути виражені, наприклад, у вигляді розміру витратних матеріалів чи зарплатні виконавця);

– τ : показник номінальних часових витрат на виконання роботи t_i ;

– k : спеціалізація робітника, яка необхідна для виконання роботи;

– d : складність роботи t_i , яка впливає на якість її виконання разом із кваліфікацією робітника;

– p : величина прибутку, яку приносить виконання роботи.

Задачі з пакету робіт мають бути виконані паралельно, але цілком реальною проблемою може стати недостатня кількість виконавців для виконання одразу усіх робіт за один хід розподілу. Тому роботи у пакеті повинні бути відсортовані в порядку їх технологічного виконання та важливості.

2.2 Аналітична модель задачі розподілу пакету робіт

2.2.1 Формування критеріїв оптимізації

Оптимізація при системному проектуванні означає процес досягнення найкращого рішення у певних умовах та певних вхідних параметрах. Важливо відзначити, що визначення «найкращого рішення» може залежати від конкретної особи або групи осіб, які займаються його пошуком.

Виходячи з цього, поняття «кращого рішення» не позбавлене суб'єктивізму, який слід враховувати та мінімізувати при вирішенні оптимізаційних завдань. Зазвичай процес прийняття рішення оцінює системний проектувальник чи спеціаліст із досвідом у цій галузі. Важливим аспектом також є те, що оптимальне рішення досягається у певних умовах: зміна умов може призвести до зміни результату та його оптимальності.

Знайти дійсно «найкраще рішення» часто буває просто неможливо через високу складність процесу оптимізації і велику кількість параметрів моделі. Тому на практиці здебільшого доводиться покращувати вже існуючі моделі [1].

В процесі побудови оптимізаційної задачі необхідно проектувальнику чи науковцеві вирішити такі завдання:

- вибрати та визначити критерії оптимізації;
- визначити проектні параметри;
- побудувати цільову функцію;
- визначити обмеження;
- вибрати метод розв'язання оптимізаційної задачі.

Критерієм оптимізації називають деяку властивість об'єкта проектування. В широкому сенсі – це деякий параметр, який надає змогу порівнювати альтернативи між собою. Критерії мають бути якомога подібнішими до цілей, щоб оптимізація за критеріями відповідала максимальному наближенню до

загальної мети оптимізації. Таким чином, знаходження значення критерію для обраної альтернативи є визначенням ступеня придатності як засобу досягнення мети [1, 22].

При системному проектуванні часто приймають рішення, спираючись на досвід та практику, які формують вибір критеріїв. У більшості ситуацій для оцінки варіантів рішень використовують множину критеріїв, які можуть бути незалежними або взаємозалежними. Проведення розподілу робіт на виробничих підприємствах є непростою задачею, яка зазвичай пов'язана з достатньо великою кількістю критеріїв.

Через те, що мету оптимізації у реальних практичних задачах часто досить важко виразити через один єдиний критерій постає проблема багатокритеріальності. Критерій, як і будь-яка модель є лише приблизним відображенням мети оптимізації, і часто його одного не достатньо для отримання адекватних результатів. Але варто зазначити, що при виявленні критеріїв важлива не їх загальна кількість, а те, наскільки повно вони покривають мету. Тобто кількість критеріїв також варто мінімізувати, адже їх надмірна кількість лише збільшить розрахункову складність моделі і призведе до зростання часу розв'язання задачі. Класичні методи дослідження операцій при розв'язанні задач розподілу робіт «добре» працюють, коли є єдиний критерій, коли критеріїв декілька, ситуація істотно змінюється [22].

У випадку проектування виробничих технологічних систем такими критеріями можуть бути: максимізація завантаженості виконавців, мінімізація часу виконання окремих робіт та/або усього пакету, максимізація якості робіт, максимізація прибутку від усіх робіт, мінімізація фінансових/матеріальних витрат на виконання робіт тощо.

Така велика кількість критеріїв потребує їх фільтрації та оцінки важливості, адже немає потреби врахувати маловажливі критерії.

Використовуючи експертну оцінку, було виявлено, що не всі з запропонованих критеріїв є важливими. Ґрунтовні експертні дослідження показали, що є три основних: мінімізація часу виконання окремих робіт або усього пакету робіт, максимізація якості робіт, мінімізація матеріальних витрат на виконання роботи.

2.2.2 Критерій оптимізації використання матеріальних ресурсів

Матеріальні ресурси є основними у виробничих системах, бо саме з них відбувається виготовлення продукції. Результати останніх етапів господарської діяльності українських підприємств явно показують, що один із ключових факторів їхньої прибутковості – це ефективне та розумне використання матеріальних ресурсів. Це видно з того, що основна частина собівартості промислових товарів та послуг пов'язана з витратами на матеріали, а наші показники в цій галузі відстають від світових стандартів. В умовах сучасної світової економіки питання про те, як ефективно використовувати матеріальні ресурси, стає ще важливішим, оскільки це безпосередньо впливає на обсяги виробництва, доходи та витрати підприємства, його фінансовий стан та конкурентоспроможність на ринку [23].

Матеріальні ресурси відносяться до внутрішніх факторів конкурентоспроможності підприємства, оскільки вартість витрачених на створення товару матеріалів (компонентів), сировини, палива, тощо прямо впливають на можливості підприємства по випуску товару. Неощадливі та неоптимальні витрати сировини призводять до зменшення прибутку, що потенційно зменшує можливості підприємства до розширення та здобування кращих позицій на ринку.

За результатами досліджень розвитку промислових підприємств останніх років можна стверджувати, що одним із найбільш стабільних методів зменшення витрат матеріальних ресурсів є впровадження більш прогресивних норм витрат ресурсів за рахунок вдосконалення технологій виготовлення виробів, більшого використання оперативного та організаційного планування. У випадку прийняття за витрати матеріальних ресурсів заробітної плати працівників, необхідно призначати найбільш вигідну кандидатуру на кожну роботу, уникаючи випадків, коли працівник із найбільшою заробітною платою виконує найпростішу та найдешевшу роботу (при цьому також треба уникати простою робітників) [23, 24].

При формуванні критерію використання матеріальних ресурсів необхідно врахувати такі чинники, як: номінальні матеріальні витрати на виконання роботи та витрати на перехід від попередньої роботи (якщо вона була і виконання робіт пакету виконується в поточному режимі, прикладом може бути

переналаштування та обслуговування обладнання).

Таким чином маємо цільову функцію критерію використання матеріальних ресурсів [14]:

$$k_1(x) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min_x, \quad (2.2)$$

де $c_{ij} = (c_{ij}^0 + c'_{ij})$, $i, j = \overline{1, n}$ – сумарні матеріальні витрати при призначенні i -ї роботи j -му виконавцю;

c_{ij}^0 – витрати на перехід від виконання попередньої роботи;

c'_{ij} – номінальні витрати на виконання роботи;

x_{ij} – змінна призначення i -ї роботи j -му виконавцю (1 – якщо робота була призначена, та 0 – якщо ні).

У випадку проведення системного проектування матеріальні ресурси можуть бути представлені у вигляді витрат на макетування, програмні комплекси та зарплатню. Номінальні витрати можна сформулювати на основі статистичних даних виконання такої роботи в минулому, або на основі розрахунків менеджерів.

Це значення є базовим, яке буде модифікуватися за допомогою кваліфікації безпосереднього виконавця. Витрати на перехід від попередньої роботи можуть бути константними, якщо виконавець не займається цією роботою безпосередньо, або також залежати від виконавця, кінцеве рішення повинно прийматися виходячи з реальної суті досліджуваного об'єкта проектування.

2.2.3 Критерій оптимізації часу виконання усього пакету робіт

Успішне функціонування підприємства в операційному плані залежить від кількох важливих факторів, в тому числі від витрат часу на виконання робіт пакету. Планування операційної діяльності передбачає осмислений вибір інструментів нормування, розробку різнорівневих планів (від оперативних до

стратегічних), прогнозування взаємозв'язків із постачальниками та споживачами, прийняття рішень та контроль їх виконання.

Ефективність обраної операційної стратегії управління часом стає основним показником продуктивності операційної системи підприємства, що робить її критично важливим для загального успіху бізнесу. Порушення термінів створення продукції може викликати проблеми з оптовими покупцями, роздрібними мережами, втратою іміджу підприємства і відповідно зниження його конкурентоспроможності. Вплив порушення строків виконання проєктних робіт призводить до затримки виконання усього пакету робіт, що може мати катастрофічні наслідки [25, 26].

Таким чином, оптимізація часових ресурсів стала одним із найважливіших факторів прогнозованості сьогодні. Показником оперативності підприємств є строки випуску продукції, і вони є суттєвою конкурентною перевагою. Прогнозованість в отриманні певного обсягу об'єму продукції є ключовою, наприклад при заключенні довгострокових контрактів.

Одним із способів представлення прогнозу проведення системного проєктування, випуску продукції, або виконання робіт є діаграми Ганта (рис. 2.1).

Даний спосіб відображення дає змогу достатньо просто та наочно орієнтуватися у планах.

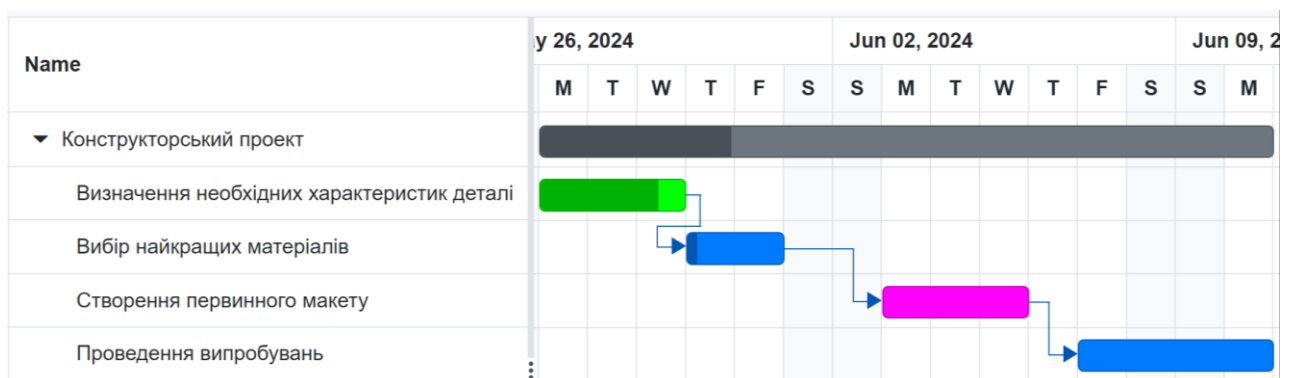


Рисунок 2.1 – Планування виконання проєкту за допомогою діаграми Ганта

Враховуючи фактор паралельного виконання робіт з пакету (при умові, що немає специфічних бізнес-правил, які накладають на це обмеження), за показник оперативності можна прийняти час виконання максимально довгої роботи із

паketу [14]:

$$k_2(x) = \max_i \{ \tau_{ij} x_{ij} \} \rightarrow \min_x, \quad (2.3)$$

де $\tau_{ij} = (\tau_{ij}^0 + \tau'_{ij})$, $i, j = \overline{1, n}$ – сумарний час виконання окремої i -ї роботи j -м виконавцем;

τ_{ij}^0 – час, необхідний для переходу від виконання попередньої роботи;

τ'_{ij} – номінальний час виконання i -ї роботи;

x_{ij} – змінна призначення i -ї роботи j -му виконавцю (1 – якщо робота була призначена, та 0 – якщо ні).

Так само, як із матеріальними ресурсами, показники витрат часових ресурсів можуть залежати від кваліфікації робітника, який був призначений на дану роботу (номінальні часові витрати та час переходу від попередньої роботи).

Номінальні часові витрати на виконання робіт з системного проектування виробничих систем можуть бути розраховані за допомогою експертних оцінок проєктувальників. Це виконується для прогнозування загального часу проведення робіт.

2.2.4 Критерій оптимізації якості виконання пакету робіт

Поняття якості є достатньо складним для формалізації. Воно є різноплановим та багатовимірним та залежить від розглянутого рівня абстракції над об'єктом дослідження.

Кожен товар або послуга, вироблені за допомогою виробничої системи, мають сукупність унікальних властивостей, які відображають їх цінність та відповідають певним потребам людей. Саме під задоволення цих потреб виробляється певний вид продукції.

Одним із основних критеріїв при цьому є саме якість, тобто сукупність властивостей продукції, що зумовлюють міру її придатності задовольняти потреби людини відповідно до свого призначення [27].

Якість продукції виробничих підприємств задається вимогами

технологічної документації і визначається тим, наскільки повно задовольняються потреби споживача у процесі використання товару.

Якість кожного виду виготовленої продукції характеризується множиною показників, серед таких характеристик можуть бути зовнішній вигляд, довговічність, надійність, рівень зношування, ресурс роботи, екологічна безпека, зручність використання та інше.

За рівних умов, таких як ціна, спосіб продажу, реклама та інші фактори, товар із вищою якістю зазвичай має більший попит.

При проведенні системного проектування виробничих систем якість моделі визначають відповідністю виконаного описання тим вимогам, які було висунуто до проектування, відповідністю одержуваних за допомогою моделі результатів.

При цьому у процесі представлення моделей проектованої системи необхідно притримуватися компромісу між детальним описом кожного елемента моделі та доцільністю такого опису, щоб зберегти цілісне уявлення про досліджуваний об'єкт.

Велику роль у забезпеченні якості і, як наслідок, конкурентоспроможності підприємства відіграють стандартизація, сертифікація та системи управління якістю [28].

Для оцінки якості створюваних програмних продуктів використовується стандарт ISO 9126, який класифікує якість з шести структурних характеристик [3]:

- функціональність: відповідність функціональних можливостей створюваного продукту потребам кінцевого користувача;
- надійність: властивість створюваного продукту зберігати свої функції та рівень якості впродовж визначеного періоду часу;
- практичність/зручність використання: набір характеристик, які визначають простоту користування зі створюваним програмним засобом;
- ефективність: відповідність витрачених ресурсів до функціональних можливостей програмного засобу;
- супроводжуваність: набір характеристик, які вказують на простоту розширення та модифікації програмного засобу;
- мобільність: властивість програмного продукту до перенесення з одного виконавчого оточення в інше.

Забезпечення якості є систематичним процесом, який відбувається не лише як технічна функція, яка контролюється відділом технічного контролю (або іншими структурними підрозділами). При реінжинірингу виробничого підприємства умовами для зростання рівня якості продукції також є повне переформатування організаційної структури підприємства, встановлення чітких факторів відповідальності кожного відділу за кінцевий результат.

Достатньо важко ввести єдиний метод формалізації якості виконаної роботи, існує декілька методів, таких як: аналітичний, статистичний, експертний і комбінований.

У моделі ми введемо універсальний показник $q \in [0,1]$, де нижня межа інтервалу буде відповідати роботі з найменшою якістю, а верхня межа інтервалу – роботі, яку виконано найкращіше. При цьому, для потреб моделі визначення даного показника можливе за допомогою експертних оцінок на які впливають кваліфікація виконавця, який призначається.

Таким чином, встановимо цільову функцію критерію оптимізації якості робіт як максимізацію найгіршого показника якості у пакеті [14]:

$$k_3(x) = \min_i \{q_{ij}x_{ij}\} \rightarrow \max_x, \quad (2.4)$$

де q_{ij} , $i, j = \overline{1, n}$ – показник якості виконання i -ї роботи j -м виконавцем;

x_{ij} – змінна призначення i -ї роботи j -му виконавцю (1 – якщо робота була призначена, та 0 – якщо ні).

2.2.5 Згортка критеріїв та побудова цільової функції

Пошук ефективного (оптимального за Парето) розв'язку задачі багатокритеріальної оптимізації є достатньо нетривіальним, а часто і неможливим. Як було зазначено раніше, часткові критерії оптимізації є лише інструментами досягання мети оптимізації, їх значення у кожній альтернативі показують ступінь її відповідності меті за обраним показником.

У практичних випадках задач багатокритеріальної оптимізації неможливо досягти одночасно найкращих значень по усім критеріям. Тобто практично

можлива ситуація, коли одна Парето-оптимальна альтернатива випереджає іншу за одним критерієм, але програє за іншим. Тому важливо мати методи, при застосуванні яких розв'язком задачі буде найкраща альтернатива із множини компромісів. При цьому важливо пам'ятати, що будь-які припущення щодо множини компромісів повинні адекватно відображатися у предметній області об'єкту оптимізації.

Важливим етапом багатокритеріальної оптимізації є нормалізація критеріїв, тобто штучне приведення їх до єдиної розмірності. Критерії за своєю практичною суттю можуть бути кардинально різних розмірностей, використовувати різні шкали та мати різну суть оптимальних значень (наприклад, де мінімум функції часткового критерію є найкращим значенням, чи навпаки).

Проведення нормалізації спрощує аналіз критеріїв та розширює список можливих способів знаходження компромісного рішення. В якості стандарту, критерії приводять в одну шкалу $\xi \in [0,1]$.

Існує декілька методів нормалізації критеріїв в залежності від характеру критеріїв та наявності інформації про їх можливі значення :

– нормалізація за еталоном: $\xi = \frac{f_i^0}{f_{\text{еталон}}}$, $i = \overline{1, n}$; інформацію про еталонне

значення критерію можна сформулювати на основі експертних оцінок, чи достовірних даних (наприклад технічної документації);

– нормалізація за максимальним значенням: $\xi = \frac{f_i^0}{f_{\text{max}}}$, $i = \overline{1, n}$; в такому

разі максимальне наявне значення виступає у ролі еталону;

– лінійна нормалізація за найкращим f^+ і найгіршим f^- значенням:

$\xi = \frac{f_i^0 - f^-}{f^+ - f^-}$, $i = \overline{1, n}$; варто зазначити, що дану функцію можна модифікувати

для більш гнучкого її налаштування: $\xi = \left[\frac{f_i^0 - f^-}{f^+ - f^-} \right]^\alpha$, $i = \overline{1, n}$, де завдяки

коефіцієнту α можна налаштовувати функцію нормалізації (коли $\alpha < 1$ – функція випукла, коли $\alpha = 1$ – функція лінійна, $\alpha > 1$ – функція увігнута) [14].

Для нормування часткових критеріїв задачі оптимізації розподілу пакетів

робіт застосуємо лінійну функцію нормалізації із нормалізацією за найкращим та найгіршим значенням:

$$\xi_l = \left[\frac{k_l(x) - k_l^-}{k_l^+ - k_l^-} \right]^1, \quad l = \overline{1,3}, \quad (2.5)$$

де $l = \overline{1,3}$ – індекс критерію (матеріальних ресурсів, часу та якості).

З практичної природи оптимізаційних задач, очевидно, різні часткові критерії мають різну важливість. Іншими словами, деякі критерії мають пріоритет перед іншими. Тому при побудові цільової функції оптимізації та визначенні сфери можливих рішень необхідно віддавати перевагу більш значущим критеріям.

Таким чином, постає питання про те, як математично визначити пріоритет та ступінь його впливу на вирішення задачі оптимізації.

Наявність пріоритету кожного критерію відкриває можливість для проведення зведення до узагальненого критерію, або згортки критеріїв.

Найбільш поширеними є адитивна (2.6) та мультиплікативна згортки (2.7) [30]:

$$P(x) = \sum_{l=1}^m \lambda_l \xi_l(x), \quad l = \overline{1,m}, \quad (2.6)$$

$$P(x) = \prod_{l=1}^m \xi_l(x)^{\lambda_l}, \quad l = \overline{1,m}, \quad (2.7)$$

де m – кількість критеріїв для згортки.

Дані види згорток мають особливість, при якій достатньо погане значення за одним із критеріїв може бути компенсовано іншими критеріями. На цю особливість треба звертати увагу при побудові обмежень моделі та встановленні пріоритетів для критеріїв, адже це може призвести до вибору альтернатив з потенційно неприйнятними значеннями критеріїв.

Дані методи згортки було модифіковано для розширення спектру можливого застосування функції корисності.

Такими модифікаціями є адитивно-мультиплікативна, ентропійна згортки та згортка за допомогою поліному Колмогорова-Габора (комбінує адитивні та мультиплікативні члени у вигляді добутку пар, трійок, четвірок, і т. д. нормалізованих характеристик альтернатив) [30]:

$$P(x) = \sum_{l=1}^m \lambda_l \xi_l(x) + \sum_{l=1}^m \sum_{i=l}^m \lambda_{li} \xi_l(x) \xi_i(x) + \dots, \quad (2.8)$$

де k – кількість критеріїв для згортки;

λ_{li} – вагові коефіцієнти нормованих критеріїв.

Застосувавши поліном Колмогорова-Габора (2.8) із двома членами, сформуємо функцію-згортку для оптимізаційної задачі розподілу пакетів робіт за критеріями матеріальних ресурсів (2.2), часових ресурсів (2.3) та якості виконання роботи (2.4):

$$P(x) = \sum_{l=1}^3 \lambda_l \xi_l(x) + \sum_{l=1}^3 \sum_{i=l}^3 \lambda_{li} \xi_l(x) \xi_i(x), \quad (2.9)$$

де λ_l , λ_{li} – вагові коефіцієнти нормованих критеріїв ξ_l , при цьому

$$\sum_{l=1}^3 \lambda_l + \sum_{l=1}^3 \sum_{i=l}^3 \lambda_{li} = 1.$$

Таким чином, повна цільова функція для оптимізаційної задачі розподілу пакетів робіт із заданими обмеженнями та функцією-згорткою матиме вигляд:

$$\left\{ \begin{array}{l} P(x) = \sum_{l=1}^3 \lambda_l \xi_l(x) + \sum_{l=1}^3 \sum_{i=l}^3 \lambda_{li} \xi_l(x) \xi_i(x) \rightarrow \max_x \\ \sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1, n}; \quad \sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1, n}; \quad x_{ij} \in \{0, 1\}, \quad i, j = \overline{1, n}. \end{array} \right. \quad (2.10)$$

2.3 Імітаційна модель розподілу пакетів робіт

Моделювання технологічних систем є достатньо комплексною задачею. Велика кількість процесів, компонентів системи, зв'язків між ними накладає на системного проєктувальника велику відповідальність та підвищує ймовірність появи помилок при проєктуванні. Тому поряд із аналітичними методами моделювання існують так звані імітаційні методи. Суттю імітаційного моделювання полягає у створенні імітаційної моделі реальної системи, її цифрового двійника. Дана модель може бути представлена у вигляді логічної схеми, яка імітує властивості та поведінку реальної системи. Так, імітаційні моделі дають змогу багаторазово перевіряти різноманітні гіпотези щодо проєктованого об'єкту без безпосереднього впливу на цей реальний об'єкт [29].

Позитивними якостями імітаційних моделей є:

- можливість отримання інформації щодо функціональних можливостей проєктованого об'єкта, його сильних та слабких сторін, фінального вигляду створюваного продукту;
- можливість швидко та «дешево» масштабувати проєктовану систему для її дослідження без залучення відносно великої кількості реальних ресурсів;
- можливість опрацювання декількох варіантів та альтернатив при виборі фінального вигляду проєкту;
- можливість швидкої зміни та адаптації продукту виходячи з отриманих при моделюванні даних про проєктований об'єкт.

Однак, як і будь-яке моделювання, воно є лише інструментом створення моделей – об'єктів приближених до реальних. Тому варто звернути увагу і на недоліки імітаційного моделювання:

- проведення імітаційного моделювання за обраними алгоритмами дає лише «точкові» чисельні результати, які, очевидно, не можуть покрити усі варіанти використання реального проєктованого об'єкту, що в свою чергу може призвести до викривлення дійсного стану речей;
- достатньо великою проблемою є отримання оцінки ступеня наближення імітаційної моделі до функціонування реального об'єкту; часто, навіть перенесення та врахування максимально великої кількості властивостей не веде до повної відповідності моделі та об'єкту;
- однією із частих неприємних помилок при проведенні імітаційного

моделювання є помилкове формування вхідних даних (особливо тих, які генеруються за допомогою різноманітних генераторів псевдовипадкових чисел) [26].

Існує два типи імітаційних моделей: детерміновані та стохастичні. Останні базуються на випадковій генерації вхідних впливів. Завдяки своїй природі, вони особливо добре допомагають у проектуванні складних виробничих чи економічних системах, в яких достатньо важко чи навіть неможливо повністю визначити внутрішні взаємодії.

Формалізація складних технологічних, економічних чи виробничих систем дозволяє системному аналітику найбільш повно дослідити (або сформувавши) їх внутрішню структуру. Одними із найбільш вивченими засобами формалізації є мережі (системи) масового обслуговування (СМО) та мережі Петрі. Методологія СМО ідеально підходить для опису стадійних виробничих бізнес-процесів.

Мережі Петрі є більш розширеним та потужним класом задач, які можна застосувати для опису найбільш складних підприємств [32].

Враховуючи стохастичний характер надходження пакетів робіт на виконання, було прийнято рішення провести модифікацію аналітичної моделі задачі та привести її до імітаційного виду моделі систем масового обслуговування.

2.3.1 Характеристика систем масового обслуговування

Системи масового обслуговування (СМО) представляють собою математичну модель, яка призначена для обслуговування вхідного потоку заявок які надходять у випадкові моменти часу, і при цьому час обслуговування кожної заявки також є випадковим. Дана математична модель спирається на потужний математичний апарат марківських випадкових процесів. Розподіл Пуассона є найбільш застосованим та найпростішим розподілом для характеру надходження вхідних заявок при моделюванні СМО, зокрема через свою найбільшу випадковість.

У іноземній літературі даний клас задач частіше називають «системами з чергами» і це не дивно, адже СМО застосовуються при моделюванні різноманітних процесів з чергами: обслуговування клієнтів на касі, заправка автомобілів, служби медичного обслуговування (наприклад ті самі «живі» черги

в поліклініках), тощо. На виробництві наразі актуальною проблемою є покращення якості обслуговування, ремонту технічного обладнання та реінжиніринг технологічних систем. Наприклад, недоступність обладнання через тимчасові або постійні несправності безпосередньо впливає на значні витрати на його обслуговування і призводить до втрати прибутку для підприємства в цілому, тому для моделювання виробничих процесів часто застосовують СМО (при цьому, несправність обладнання також моделюється засобами СМО) [32].

При моделюванні важливо отримати корисну інформацію про об'єкт моделювання, тому нижче наведено основні та найбільш критичні показники, які можна виявити засобами СМО:

- середня кількість заявок, яку може обробити СМО за одиницю часу;
- середній час очікування заявки в черзі;
- середня та максимальна кількість заявок в черзі;
- коефіцієнт завантаженості каналу.

У формальному вигляді, кожна СМО характеризується такими обов'язковими структурними компонентами, як:

- джерело заявок;
- вхідний потік заявок;
- накопичувач (черга);
- канал (пристрій обробки заявки);
- вихідний потік заявок.

Джерело заявок є зовнішньою системою по відношенню до СМО з якої і надходять вхідні заявки. В ідеальних системах таке джерело має бути нескінченним генератором вхідного потоку заявок, але якщо дану умову виконати неможливо, в джерело треба завантажити достатньо велику кількість заявок, щоб повністю покрити час імітаційного моделювання.

Вхідний потік заявок характеризується часом надходження заявки до системи та їх кількістю. Якщо декілька пов'язаних заявок надходить до системи, то їх називають пакетом. Максимальна кількість заявок в пакеті може бути обмеженою та не обмеженою. В загальному випадку заявки можуть надходити у фіксовані моменти часу (за детермінованим законом) чи у випадкові моменти часу (за ймовірнісним законом). В останньому випадку проміжки часу між часом надходження заявок можуть описуватися різноманітними законами розподілу:

нормальним, рівномірним, експоненціальний, тощо. Важливим правилом по відношенню до властивостей заявки є те, що в СМО заявка не може бути знищена, поєднана із іншою заявкою або розділена на декілька субзаявок.

Після надходження заявки у систему, є два можливих шляхи її подальшої обробки. Якщо на цей момент канал зайнятий, то заявка переходить до накопичувача заявок. Порядок розміщення у черзі зазвичай є по часу отримання. Як тільки канал звільняється, із накопичувача забирається наступна заявка. Існує декілька правил, по яким заявка може бути вивільнена із черги.

Найбільш поширеним та простим є правило FIFO – заявка яка першою надійшла до черги першою і буде вивільнена. Іншим правилом є LIFO – остання із заявок в черзі буде вивільнена першою. Також існують і інші екзотичні правила, в залежності від реальної практичної суті системи [32].

Заявка може мати пріоритет обслуговування, що дає можливість ігнорувати її положення в черзі по відношенню до заявок з меншим пріоритетом. Пріоритет заявки може бути відносним та абсолютним. Так звані заявки з абсолютним пріоритетом можуть навіть переривати виконання іншої заявки з меншим пріоритетом в каналі заради виконання даної. Заявка, яка знаходиться у черзі характеризується часом перебування у черзі, однією із основних задач оптимізації є зменшення цього часу за рахунок пришвидшення обробки або збільшення кількості каналів.

СМО може мати один або декілька обслуговуючих каналів. Збільшення кількості каналів підвищує загальну продуктивність системи. В залежності від реального пристрою, який моделюється каналом, необхідно враховувати його динамічні параметри, які можуть впливати на його продуктивність. Наприклад, якщо модельованим пристроєм є виробниче обладнання, воно може перегріватися, що спричинить зменшення його виробничої потужності.

Вихідний потік заявок формується із обслужених та не обслужених заявок в каналах СМО. При поєднанні декількох простих СМО утворюються багатофазні СМО, де вхідним потоком наступної фази є потік попередньої.

Розглядаючи модель СМО у загальному вигляді передбачається, що у початковий момент часу імітаційного моделювання система повинна перебувати в невизначеному стані, або повністю пуста. Також, аналіз СМО має значення лише при її функціонуванні протягом достатньо великого проміжку часу, адже це збільшує точність отриманих результатів і зменшує варіативний вплив

випадкового потоку вхідних заявок [32, 33].

2.3.2 Застосування СМО при імітації розподілу пакетів робіт

Для модифікації аналітичної моделі розподілу пакетів робіт технологічної системи під імітаційне моделювання було застосовано математичний апарат СМО. Як було зазначено раніше, дана модель імітаційного моделювання добре підходить під стохастичний характер отримання пакетів робіт системою.

В результаті було створено трифазну багатоканальну СМО з трьома пріоритетами обслуговування. Схему СМО зображено на рисунку 2.2, вона складається з джерела вхідних пакетів заявок G , черги $Q_{1,1}$ та каналу розподілу заявок $C_{1,1}$, клапанів перед чергами $Q_{2,n}$ та каналами виконання заявок $C_{2,n}$ та черги $Q_{3,1}$ і каналу агрегації $C_{3,1}$ [14].

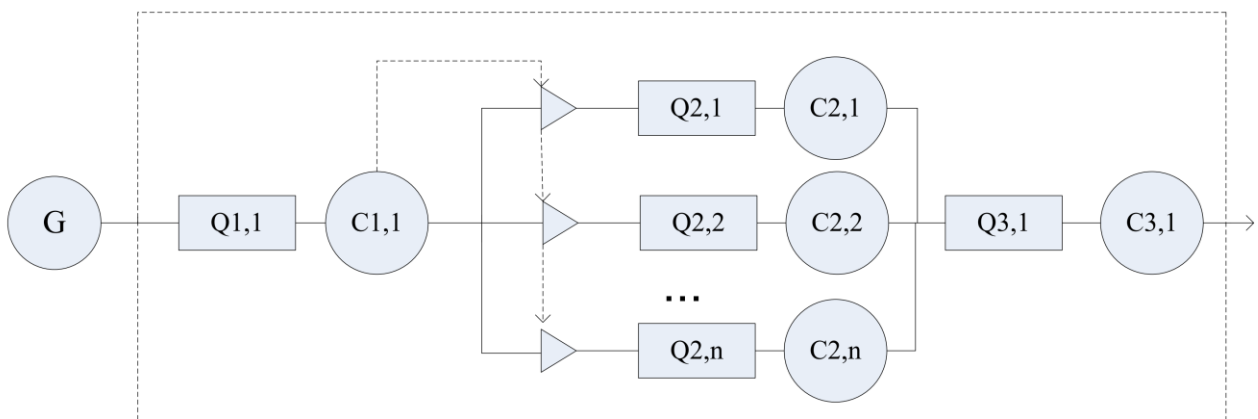


Рисунок 2.2 – Схема трифазної СМО для процесу розподілу та виконання пакетів робіт

Канал розподілу заявок $C_{1,1}$ імітує проведення розподілу заявок із отриманого пакету серед вільних в системі каналів обробки $C_{2,n}$ використовуючи оптимізаційну модель (2.10).

При цьому обробка пакетів робіт виконується за схемою FIFO, адже це стандартна практика для більшості виробничих технологічних систем. Час розподілу залежить від кількості заявок в пакеті. Якщо на момент отримання пакету робіт вільних каналів обробки немає, то пакет відправляється у чергу і не

розподіляється.

Канали виконання заявок $C_{2,n}$ імітують виконання робіт пакету. Канал є цифровим двійником виконавця роботи, тому імітований час виконання роботи залежить від встановленого каналу рівня кваліфікації β і номінального часу виконання роботи τ'_{ij} . При цьому, канали можуть незалежно один від одного виконувати заявки із різних пакетів.

Канал агрегації заявок $C_{3,1}$ потрібен для розрахунку якості виконаних заявок (згідно критерію (2.5), і можливої відправки заявки на перевиконання при занадто низькій якості. У формальному вигляді появу недостатньо якісно виконаних заявок можна подати як випадкову подію із ймовірністю p . Після потрапляння до накопичувача даного каналу усіх заявок пакету і успішного проходження усіх перевірок, заявки агрегуються (збираються) знову в пакет, який виходить із системи.

Таким чином, на стадії розподілу заявок та їх агрегації необхідно витратити додаткові ресурси на їх виконання (часові та матеріальні), тому модифікуємо критерії оптимізації таким чином, щоб на етапі імітаційного моделювання була можливість врахувати дані витрати.

Тоді критерій матеріальних ресурсів (2.2) матиме вигляд:

$$k_1(x) = c_{\Delta} + \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min_x, \quad (2.11)$$

де c_{Δ} – сумарні витрати матеріальних ресурсів на першій та третій фазі.

Критерій витрат часових ресурсів (2.3) матиме вигляд:

$$k_2(x) = \tau_{\Delta} + \max_i \{ \tau_{ij} x_{ij} \} \rightarrow \min_x, \quad (2.12)$$

де τ_{Δ} – сумарні витрати часових ресурсів на першій та третій фазі.

Далеко не всі пакети заявок чи окремі роботи є рівнозначними, тому застосуємо механізм пріоритетів СМО для першочергового обслуговування деяких пакетів.

Пропонується ввести гнучкий відносний пріоритет для заявок a^w .

Введемо 3 значення пріоритетів:

- $a^w = 0$ для низького пріоритету;
- $a^w = 1$ для стандартного пріоритету;
- $a^w = 2$ для високого пріоритету.

Зважаючи на ймовірнісний характер виникнення проблем із якістю, це може спричинити до затримки виконання усього пакету на досить тривалий час. Якщо заявка була переміщена до черги розподілу в самий кінець, час виконання залежить від розміру черги. Завдяки введенню пріоритетів на виконання робіт можна надати такій заявці найбільший пріоритет $a^w = 2$, і вона буде розподілена для повторного виконання при наступному розподілі [17].

2.3.3 Моделювальний алгоритм розподілу пакетів робіт

Розглянемо більш детально імітаційний алгоритм СМО, який буде використано для моделювання розподілу пакетів робіт у технологічній системі.

Початок імітаційного моделювання. На першому етапі необхідно ввести до імітаційної системи усю вхідну інформацію про:

- вид та параметри закону розподілу вхідного потоку пакетів заявок для формування часу отримання пакетів заявок та їх кількості;
- вид та параметри закону розподілу властивостей кожної окремої заявки (витрат матеріальних ресурсів c_{ij} , часових витрат τ_{ij} та рівня якості q_{ij});
- номінальні витрати на проведення розподілу та агрегації пакету заявок c_{Δ} та τ_{Δ} ;
- значення вагових коефіцієнтів λ_l , λ_{li} для цільової функції розподілу (2.10);
- значення граничного рівня якості q_{\min} для виявлення браку;
- початкове значення лічильника часу та умов зупинки моделювання.

Крок 1. На основі закону розподілу джерело G повинно сформувати новий пакет та встановити для нього час надходження до системи.

Крок 2. Отримання каналом розподілу заявок $C_{1,1}$ нового згенерованого

пакету заявок в точний момент часу. Якщо канал розподілу зайнятий, чи немає вільних каналів для виконання заявок $C_{2,n}$, то пакет повністю переміщується до накопичувача $Q_{1,1}$. Якщо канал розподілу $C_{1,1}$ вільний, і є хоча б один вільний канал виконання заявок $C_{2,n}$, то виконується розподіл в якому беруть участь усі канали $C_{2,n}$.

Крок 3. Канал $C_{1,1}$ виконує розподіл за час, вказаний у параметрі τ_{Δ} . Решта не розподілених заявок з пакету переміщується до накопичувача (черги) $Q_{1,1}$. Заявки, які було розподілено до каналів $C_{2,n}$, які на даний час зайняті, переміщуються до черг $Q_{2,n}$. Канал $C_{1,1}$ звільняється.

Крок 4. Кожен із зайнятих каналів $C_{2,n}$ імітує виконання роботи, яку відображає заявка за розрахунковий час, який було зіставлено на основі імітованих даних кваліфікації виконавця та номінального часу виконання роботи.

Крок 5. Виконана заявка з каналу $C_{2,n}$ переходить до третьої фази СМО і переміщується до каналу агрегації $C_{3,1}$, якщо він вільний, або до черги $Q_{3,1}$, якщо ні. Канал $C_{2,n}$ бере наступну розподілену заявку із черги $Q_{2,n}$, якщо вона там присутня. Канал першої фази СМО $C_{1,1}$ виконує наступний розподіл для заповнення черги даного виконавчого каналу (перехід на крок 3).

Крок 6. Канал агрегації $C_{3,1}$ отримує усі заявки із пакету та обчислює якість усього пакету на основі критерію $k_3(x) = \min_i \{q_{ij}x_{ij}\} \rightarrow \max_x$. Якщо розрахункова якість виявляється нижчою за встановлений ліміт q_{\min} , то виконаній заявці з даним рівнем встановлюється найвищий пріоритет $a^w = 2$ та вона переміщується до черги $Q_{1,1}$ на повторний розподіл та виконання. Якщо рівень якості є прийнятним, то пакет заявок вважається виконаним та канал $C_{3,1}$ звільняється.

Крок 7. Запис статистичної інформації про виконання даного пакету до сховища даних. Цикл повторюється з кроку 1.

Крок 8. Виконання статистичного аналізу проведеного імітаційного моделювання, знаходження таких параметрів як:

- загальній кількості заявок та пакетів;
- максимального та середнього розміру черги $Q_{1,1}$, $Q_{1,n}$, $Q_{3,1}$;
- максимального, мінімального та середнього часу перебування заявки в черзі;
- мінімальний, середній та максимальний час між двома пакетами заявок;
- мінімального, середнього та максимального часу обслуговування однієї заявки та всього пакету в цілому.

2.3.4 Оцінка точності результатів моделювання

В реальних практичних задачах недостатньо лише побудувати аналітичну та імітаційну моделі для отримання адекватних та якісних результатів. Так як будь-яка модель є лише відображенням реального об'єкту моделювання, необхідно постійно перевіряти ступінь відповідності моделі цілям моделювання.

В загальному випадку опираються на два критерії правильності побудови моделі: ступеню відповідності моделі реальному об'єкту та точності результатів моделювання.

Перший критерій часто називають верифікацією, тобто перевіркою, наскільки точно було переведено об'єкт моделювання у логічні структури моделі, його модельний опис. Математична модель повинна математично та логічно, з встановленим рівнем точності, відповідати моделюваному об'єкту, процесу, чи явищу. Після проведення логічного моделювання, на подальших етапах розробки імітаційного продукту, варто сконцентрувати увагу на якості програмної реалізації, яка є ключовою частиною для проведення імітаційного моделювання, чи не було втрачено ті чи інші критичні деталі при таких кожних таких перетвореннях.

Другий критерій являє собою перевірку адекватності роботи самої моделі, чи відповідає набір вихідних даних моделі очікуваному результату. В більшості випадків моделі є достатньо великими та комплексними, тому варто сформулювати набори для кожного окремого логічного вузла моделі і проводити попереднє тестування окремо для кожного з них. Даний підхід дозволить спростити тестування та складнощі з виявлення модуля, який дає помилку.

Коли тестування кожного окремого вузла завершено, необхідно провести

тестові прогони усієї моделі. Для цього необхідно сформувавши достатньо великий набір тестових вхідних даних з встановленими еталонними результатами для подальшого порівняння з виходами моделі.

Також, в такі набори варто включати спеціальні групи даних для проведення аналізу чутливості моделі до зміни вхідних даних, особливістю цих груп є достатньо невелика різниця в даних, і при цьому вона не повинна спричиняти велику різницю на виході моделі. Обґрунтування адекватності моделі доводить, що вона працює із встановленим рівнем точності, який задовольняє обрану мету імітаційного моделювання [29].

Враховуючи, що розроблена імітаційна модель трифазної СМО (рис. 2.2) у якості вхідного потоку заявок приймає випадковий потік заявок із заданим законом розподілу, необхідно провести достатньо велику кількість прогонів імітаційного моделювання із різними випадковими числами.

Точність проведення моделювання докорінним чином залежить від кількості проведених над моделлю експериментів та їх тривалості. Особливу увагу також варто приділити початковим умовам при старті моделювання. Зазвичай моделювання починається при умовах, що модель є «пустою»: усі канали є не завантаженими і усі черги є пустими.

Визначити достатню кількість прогонів (експериментів) імітаційної моделі можна на основі обраної точності використовуючи методи оцінки точності статистичного імітаційного моделювання.

Нехай у якості аналізованого параметру буде обрано математичне сподівання деякого параметра моделі:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad (2.13)$$

де x_i – значення аналізованого параметра;

N – кількість експериментів.

Тоді на основі формули оцінки математичного сподівання (2.13) можна встановити, що похибка аналізу становить $\varepsilon = |x_{\text{факт}} - \bar{x}|$, де $x_{\text{факт}}$ – фактичне значення аналізованого параметра.

При проведенні імітаційного моделювання у якості достовірності оцінки

використовується ймовірність α , що значення похибки ε не перевищить граничне задане значення ε^* : $p = \left[\left| x_{\text{факт}} - \bar{x} \right| \leq \varepsilon^* \right] = \alpha$.

Використовуючи задані умови, сформуємо співвідношення для визначення необхідної кількості експериментів N^* для досягання встановленого рівня точності ε^* (2.14).

На початкових етапах імітаційного моделювання проводиться деяка невелика кількість тестових прогонів (наприклад 10-20 із набору) для визначення рівня дисперсії по аналізованих параметрах. Потім, на основі отриманих даних необхідно обрати параметр із найбільшою дисперсією та визначити мінімально необхідну кількість експериментів N^* на його основі [18].

$$\varepsilon = \frac{t_{\alpha} \sigma}{\sqrt{N}}, \quad N^* = \frac{t_{\alpha}^2 \sigma^2}{(\varepsilon^*)^2}, \quad (2.14)$$

де t_{α} – квантиль нормального розподілу ймовірностей заданого рівня достовірності α ;

σ – середньоквадратичне відхилення оцінки \bar{x} .

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ

3.1 Обґрунтування вибору середовища імітаційного моделювання та мови програмування

Розвиток високопродуктивних обчислювальних систем значно збільшив можливості імітаційного моделювання, швидкість отримання результатів та потенційну їх точність. Застосування методів паралельних обчислень також сприяє оптимізації використання комп'ютерних ресурсів при проведенні моделювання.

Вибір мови програмування залежить від типу імітаційної моделі, її складності, специфічності вимог по розгортанню програмного продукту, необхідності підтримки користувацького інтерфейсу. Програми імітаційного моделювання розробляються як за допомогою спеціалізованих мов, так і із застосуванням мов загального призначення. Спеціалізовані мови імітаційного моделювання поділяють за способом встановлення внутрішнього системного часу: неперервні, дискретні та комбіновані.

Дискретні мови програмування розроблені для моделювання дискретних процесів (хоча деякі із них і можуть мати додаткові модулі для підтримки роботи у режимі неперервних процесів). Їх ключовою особливістю є наявність засобів для керування процесом моделювання. Для цього в імітаційній системі створюються динамічні списки подій різних типів:

- основні, які використовуються для переведення систему з одного стану в інший (наприклад отримання надходження заявки у канал СМО);
- допоміжні, які використовуються для виконання сервісних функцій імітаційної систем (наприклад оновлення порядку черги заявок після вивільнення однієї із них).

Враховуючи подійний принцип роботи розроблюваної імітаційної моделі СМО, для моделювання описаного в даній роботі найкраще підходять спеціальні дискретні мови імітаційного програмування, або мови загального призначення. При цьому основна роль розробника імітаційного продукту полягає у створенні подій та встановленні їх у заданому логічному порядку у модельному часі [29].

Спеціальна мова імітаційного програмування GPSS (General Purpose

Simulation System) була розроблена Джеффри Гордоном коли він працював у ІВМ. Вона була створена с метою побудови імітаційних моделей середньої складності для Q-схем і проведенню експериментів з ними. Її використовують при моделюванні потоків робіт на виробничих підприємствах, потоку заявок комп'ютерних та сервісних систем.

За своєю природою це об'єктна мова програмування для побудови систем за допомогою декларативного підходу. Елементами даної мови є транзакти (наприклад вхідні заявки) та блоки (оператори). Згенеровані транзакти обробляються набором блоків, вони являють собою правила роботи симуляції, які було задано розробником. Її остання реалізація для операційної системи Windows GPSS World принесла набір нових функцій, таких як підтримка скриптової мови розширення PLUS, можливості відображення статистики та графіків для аналізу вихідних параметрів моделювання [34].

Перевагами мови GPSS є висока ефективність при моделюванні СМО, автоматичне збирання статистичної інформації під час проведення моделювання та достатньо проста блочна структура програм, що значено спрощує роботу розробника. Негативною стороною даної мови моделювання є її застарілість, що продукує проблеми із сумісністю з сучасним обладнанням, обмеженість можливостей вводу та виводу даних та специфічність побудови програм, що спричинює достатньо високий поріг входу для розробника.

Arena Simulation Software являє собою комплексний програмний продукт для проведення дискретних симуляцій різної складності. Дане програмне забезпечення використовує мову симуляцій SIMAN, та має достатньо широкі можливості по своїй адаптації для побудови імітацій із різних предметних областей. Процес створення моделі є інтерактивним, він відбувається у візуальному редакторі при взаємодії з яким кожен елемент реального об'єкту моделювання може бути виражений у вигляді модуля з деякими встановленими параметрами. Такі модулі збираються в імітаційну систему, а при виконанні моделювання по даній системі проходять сутності. Такий підхід спрощує створення імітаційної моделі та знижує поріг входу для розробника.

Arena має вбудований інтерфейс збору статистичних даних, які можуть бути виведені в докладні звіти, допомагаючи в аналізі та оптимізації змодельованої системи. Дане програмне забезпечення ідеально підходить під моделювання складних бізнес-процесів, яке не обмежується жодною

конкретною галуззю та може використовуватися для моделювання систем як у виробничому секторі, так і в секторі послуг. Сучасні модулі дозволяють проводити інтеграцію із багатьма Enterprise-системами, наприклад з сервісами Microsoft [35].

Існуючими проблемами із даним програмним продуктом є складність із ліцензуванням та інтеграцією. Arena постачається виключно на комерційній основі та має обмеження з приводу гнучкості та можливості інсталяції до існуючих корпоративних систем.

Розглядаючи мови загального користування, можна виділити таку відому та популярну мову як Java. Це високорівнева об'єктно-орієнтована мова програмування з понад 30-ти річною історією розвитку. Однією із ключових особливостей, завдяки вона завоювала свою популярність це принцип кросплатформеності, тобто можливості запуску створених на даній мові програм на будь-якому обладнанні.

Хоча Java і не створювалася напряму з метою використання для побудови імітаційних моделей математичного програмування, але вона має дуже сильний вбудований математичний апарат у вигляді спеціалізованих типів даних та функцій, який також можна легко розширити за допомогою додаткових Open Source бібліотек Apache. Таким чином за допомогою даної мови програмування можна розв'язувати задачі лінійної оптимізації, регресійного аналізу та статистики. Також дана мова програмування має прості та надійні бібліотеки для побудови backend застосунків (такі як Web-сервери Spring, Jersey), що значно спрощує інтеграцію програмного продукту та надає можливості для побудови під нього користувацького інтерфейсу у вигляді веб-форм.

На жаль в Java не існує візуальних редакторів для створення імітаційних моделей, як в Arena, що накладає обмеження та потребує залучення спеціалістів із розробки програмного забезпечення спеціально на цій мові програмування (так само як для GPSS). Але дана незручність повністю компенсується майже нескінченними можливостями по розробці спеціалізованої імітаційної моделі у такому вигляді, який це потребує бізнес.

Провівши ґрунтовний аналіз існуючих рішень для створення імітаційних моделей та з огляду на перелічені фактори, було прийнято рішення про написання імітаційної моделі з використанням мови програмування Java.

3.2 Обґрунтування вибору СУБД

Для проведення імітаційного моделювання необхідно обрати таку СУБД (систему управління базами даних), яка може витримувати високі навантаження і надавати можливості до простого розширення, адже в процесі моделювання може виявитися, що не було враховано деякі важливі параметри.

На ринку зараз існують сотні СУБД різних типів: реляційні, документо-орієнтовані, графові, «ключ-значення». Реляційні є класичними та ідеально підходять під задачі моделювання. Через свою попередньо визначену логічну структуру, вони надають можливість зручно зберігати як вхідні дані до моделювання, так і його результати. Завдяки схожому інтерфейсу доступу до даних і мови SQL існує багато готових бібліотек для взаємодії з різними базами даних для мови програмування Java. Підтримка транзакцій, індексації та масштабування є мінімальним стандартом і вони присутні у всіх найкращих СУБД даного класу.

Із реляційних СУБД найпопулярнішими вважаються: Oracle EE, MySQL, Microsoft SQL Server та PostgreSQL (рис. 3.1) [36].

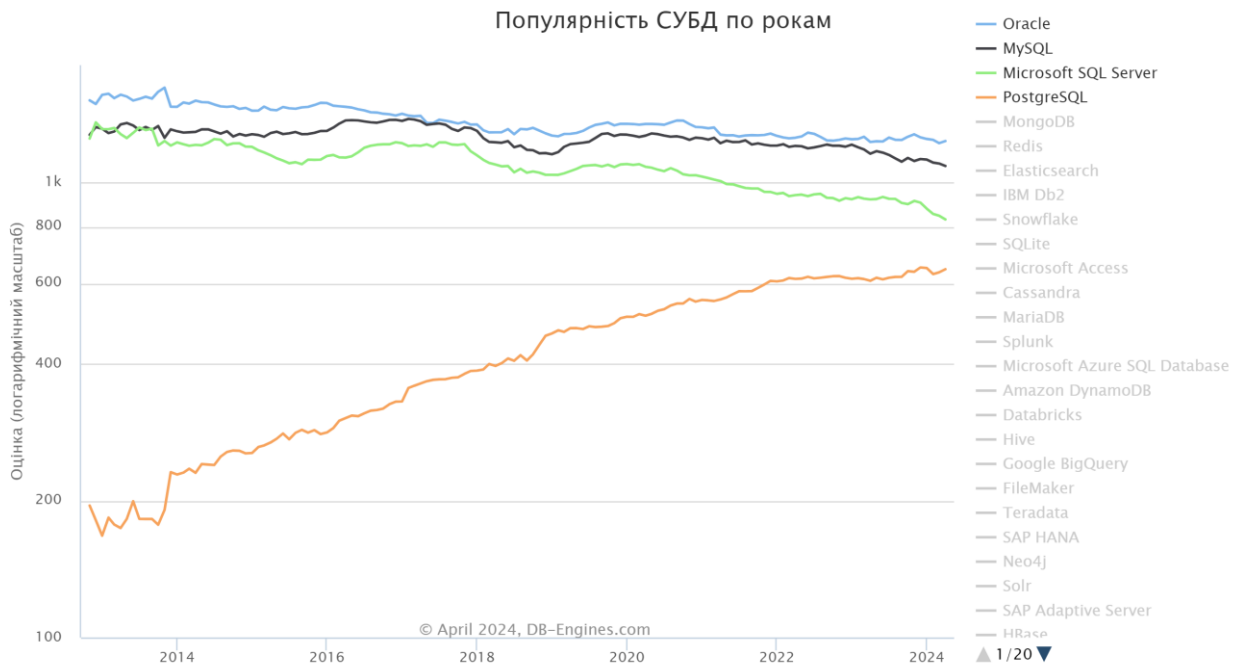


Рисунок 3.1 – Графік популярності СУБД на початок 2024 року [36]

Oracle EE є однією із найстаріших комерційних СУБД, яка і досі є лідером галузі. Вона надає найбільш широкий перелік функцій для задоволення вимог корпоративних середовищ будь-якого масштабу. Вона легко може обробляти великі масиви даних з високою продуктивністю та масштабованістю, має вбудовані модулі для швидкого розгортання у кластері для забезпечення високої доступності, реплікації та відновлення після відмови. Із мінусів є складність налаштування, високий поріг входу для обслуговування (потребує наявності спеціалістів високого класу) та висока вартість комерційної ліцензії [37].

Microsoft SQL Server є комерційною СУБД від Microsoft. Її характерними особливостями є орієнтованість на застосування у комплексі з іншими продуктами даної компанії, у даної СУБД є інтеграція з іншими продуктами як Microsoft Access та Azure на Windows машинах, а також підтримує інтеграцію із сервісами штучного інтелекту. Для формування запитів до бази даних використовується своя власна SQL мова T-SQL, що вимагає наявності спеціаліста який має досвід її використання [38].

MySQL є відкритою СУБД, яку також розробляє компанія Oracle. Вона є лідером серед не комерційних СУБД через свою простоту, як налаштування так і роботи, а також швидкості роботи при невеликій завантаженості. Вона має обмежений набір базових типів даних, один спосіб індексації даних (індекс b-tree). Із її сильних сторін є висока продуктивність при операціях зчитування даних та просте горизонтальне масштабування.

В останні роки особливу популярність набирає відкрита СУБД PostgreSQL. Ключовим фактором, який відрізняє її від аналогів є можливість тонкого налаштування кожного параметру СУБД під конкретний варіант використання. В ній існує декілька варіантів індексації, велика кількість вбудованих типів даних, стандартизація під SQL, та наявність модульної платформи розширення, завдяки якій PostgreSQL можна доповнити різноманітними модулями, які додають нові типи даних та змінюють логіку роботи з таблицями [39].

Провівши дослідження по наявним на ринку СУБД було прийнято рішення щодо використання відкритої СУБД PostgreSQL через свою відносну простоту, відкритість стандартам SQL та наявності великої кількості доповнюючих модулів, які можуть значно спростити та розширити взаємодію з БД.

3.3 Опис обраної предметної галузі моделювання

В якості цільової предметної галузі моделювання було обрано станцію технічного обслуговування автомобілів (автосервіс, СТО). Дана предметна галузь є достатньо хорошим прикладом технологічної системи, яку необхідно оптимізувати засобами системного підходу. При цьому застосування моделей оптимізації може бути застосовано як на етапі створення (проектування) підприємства з ремонту автомобілів, так і на етапі його функціонування, з метою підвищення ефективності виконання робіт та збільшення його економічних показників.

На етапі проведення системного проектування підприємства автосервісу виникає низка фундаментальних проблем, які можуть суттєво вплинути на успішність кінцевого результату:

- вибір місця з множини аналогів під виробничі та ремонтні приміщення;
- вибір та топологія обладнання, такого як підйомники, мийки, піскоструминні та лакофарбові камери, тощо; прикладом погано спроектованої топології обладнання є розміщення двох підйомників занадто близько один до одного так, що використання одного із них блокує використання іншого і навпаки;
- вибір кваліфікованого персоналу для виконання робіт.

Вносячи зміни до критеріїв моделі (2.10) відповідно до параметрів вказаних проблем, розроблена у розділі 2 математична модель може бути застосована для проведення системного проектування такого підприємства.

У випадку проведення оптимізації вже існуючого автосервісу, перш за все необхідно визначити його існуючу внутрішню структуру. З формальної точки зору кожне підприємство функціонує згідно Закону України від 28 листопада 2014 року № 615 «Про затвердження Правил надання послуг з технічного обслуговування і ремонту колісних транспортних засобів» [40]. Для даної роботи було проведено абстрактний аналіз одного із підприємств даної галузі, яке має інформаційний ресурс у мережі Інтернет.

Бізнес-процес по обслуговуванню автомобілів починається із звернення клієнта до адміністратора автосервісу особисто або через мережу Інтернет. Автосервіс має чіткий список послуг, які він може надавати. У кожній послугі є встановлені нормативи витрат матеріальних та часових ресурсів, які беруться з

загальної бази даних дилерських автомайстерень або із статистики по цьому сервісу.

Автосервіс встановлює для кожної послуги норму прибутку, який буде отримано у випадку її виконання. Якщо клієнт знає які послуги йому потрібні, їх можна замовити одразу. Якщо ж проблема несправності автомобіля невідома, то спершу необхідно виконати діагностику. В будь-якому випадку, після підписання договору про виконання послуг формується пакет (список) послуг на виконання, який передається робітникам сервісу (виконавцям). Робітники виконують послуги згідно своєї спеціалізації із встановленим рівнем кваліфікації. Після закінчення виконання послуги спеціальний працівник (або бригадир) перевіряє якість виконаної роботи. Якщо вона задовільна, вона приймається, якщо ні – бригадир змушує працівника виправити помилку або навіть виконати послугу повторно, в залежності від конкретного випадку.

Після виконання усіх послуг транспортний засіб передається клієнту на перевірку. При задовільній відповіді від клієнта та оплаті договір та всі послуги вважаються виконаними в повному обсязі.

3.4 Архітектура програмного засобу

Виходячи із сказаного в описі предметної галузі підприємства автосервісу видно, що вона працює за схемою багатофазної багатоканальної СМО, як було зображено на рисунку 2.2. Для тестування імітаційної моделі та розробки програмного продукту буде використано описані дані про послуги та працівників сервісу. Користувачем продукту є адміністратор (керуючий) автосервісу, який буде його використовувати для розподілу робіт, планування навантаження, моделювання розширення підприємства та коригування пріоритетів витрат (матеріальних та часових ресурсів або якості). Запускатися даний продукт повинен на будь-якому комп'ютері сервісу, як звичайний веб-додаток.

З точки зору архітектури програмний продукт складається із чотирьох основних компонентів (рис. 3.2):

- веб-форми для запуску моделювальних сесій, перегляду результатів та корекції вхідних даних;
- клієнтського веб-серверу, який приймає запити від веб-форми та взаємодіє з базою даних та імітаційним ядром;

- імітаційного ядра, яке і виконує імітаційне моделювання згідно аналітичної моделі (2.10) і рис 2.2;
- бази даних для збереження вхідних параметрів та результатів моделювання, в даний імітаційний програмний продукт також буде перенесено дані про роботи та виконавців із вже існуючої бази даних сервісу.

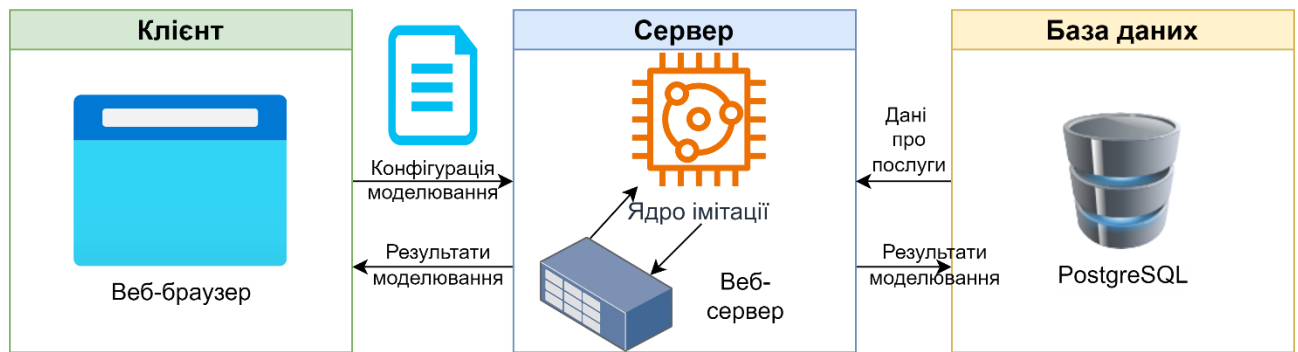


Рисунок 3.2 – Архітектура імітаційного програмного засобу

Було визначено, що найзручніше для адміністратора автосервісу переглядати результати моделювання у вигляді Microsoft Excel файлів, тому було прийнято рішення про формування звітів саме в цьому форматі, а самі файли зберігати у спеціально виділеній таблиці БД.

Дана архітектура побудована таким чином, що її дуже просто розширювати новими модулями імітації. Завдяки прихованій структурі, клієнт у вигляді веб-браузеру не має прямого доступу до бази даних, що автоматично дає певний рівень безпеки внутрішніх даних.

3.5 Опис схеми бази даних

При розробці імітаційної програми необхідно ретельно підійти до процесу синтезу бази даних. Створювана БД повинна відповідати вимогам до повторюваності експериментів, тобто повинна зберігати повну інформацію про вхідні параметри моделювання та отримані результати. Це дозволить проводити аналізи ефективності проведених експериментів та їх адекватність.

Для моделювання баз даних було використано методологію проєктування IDEF1X. Вона призначена для побудови логічної структури реляційної бази даних, яка не залежить від конкретної СУБД, а лише має декларативний опис

проектованих сутностей, атрибутів та зв'язків між ними. Сутності являють собою об'єкти, інформацію про які необхідно зберігати у системі, атрибути відображають властивості та параметри цієї сутності. Особливий атрибут, який ідентифікує сутність (тобто дає змогу відокремити одну сутність від іншої) називають первинним ключем. Такий ключ має бути у кожній сутності. Зв'язки є логічними співвідношеннями між сутностями. Таким чином одна сутність може «посилатися» іншу. Зв'язки можуть розширювати множину атрибутів сутності, збільшуючи сукупну інформацію про неї [41].

Розглянемо створену логічну модель даних для програмного продукту проведення імітаційного моделювання автосервісу (рис. 3.3).

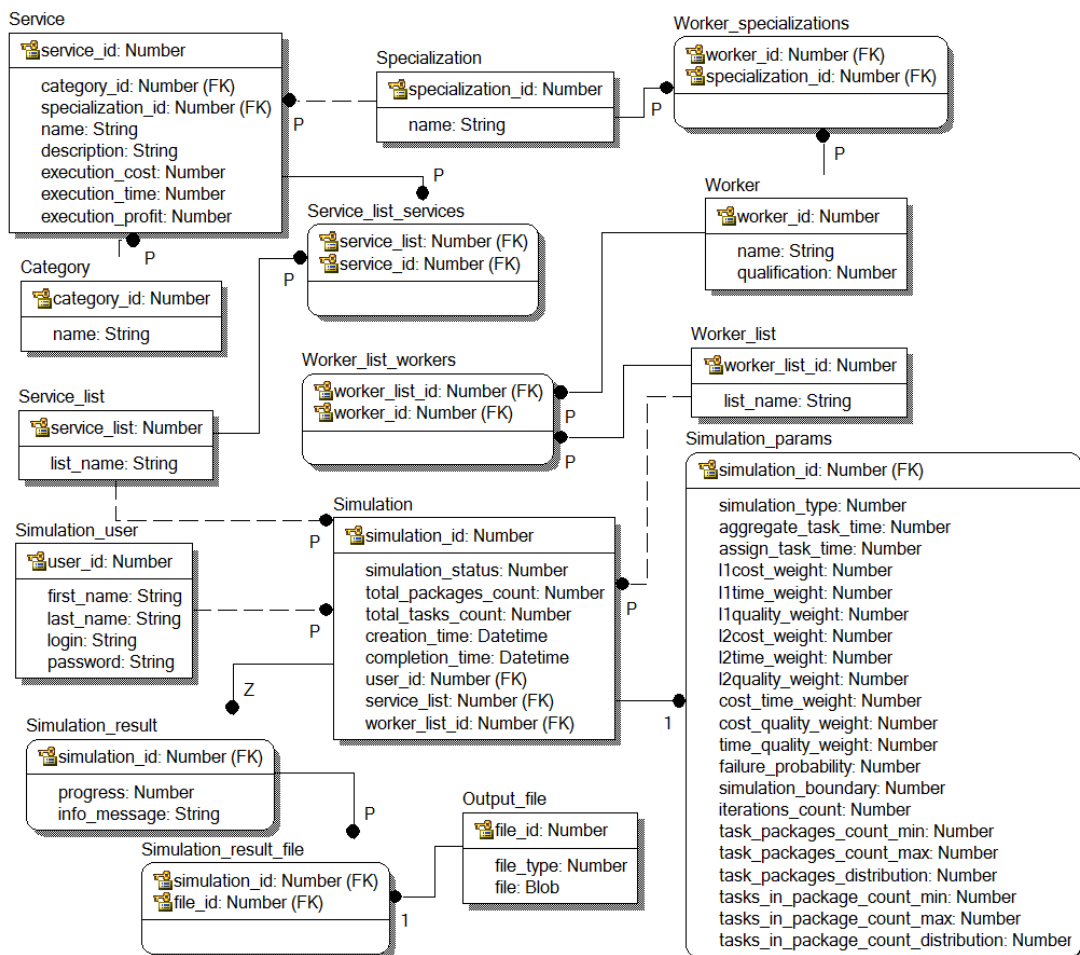


Рисунок 3.3 – ER-діаграма логічної моделі даних

Дана модель складається з 15 сутностей, кожна з яких виконує важливу роль у точному збереженні моделювальної інформації для її подальшого аналізу. Нижче приведено опис основних із них.

Service – дана сутність призначена для збереження інформації про послуги, які надає автосервіс. Дані послуги трансформуються у роботи, які буде додано до пакетів робіт при їх генерації під час проведення імітаційного моделювання. Наповнення записів даної сутності даними можливе завдяки наявності повної інформації у власника автосервісу (такої як норма витрат часу та матеріальних ресурсів, а також рівень прибутку).

Worker – сутність призначена для збереження даних про робітників підприємства, головним чином з них формується список виконавців, яких буде задіяно для проведення моделювання. Робітники характеризуються рівнем своєї кваліфікації, яка впливає на параметри виконання роботи (час, витрати та якість) та списком спеціалізацій, який дає право на виконання тієї чи іншої роботи.

Service List – сутність яка відображає точний список послуг, з яких буде сформовано роботи при проведенні імітаційного моделювання.

Worker List – сутність яка відображає точний список робітників, з яких буде сформовано виконавців при проведенні імітаційного моделювання.

Simulation – це головна сутність моделі, яка призначена для збереження ключової інформації про сесію імітаційного моделювання. Вона поєднує в собі зв'язки як з вхідними даними моделювання (список послуг, список робіт, параметрами моделі, інформацію про власника сесії), так зв'язки з вихідними даними – результатом та списком файлів. Додатково до неї було додано поля для відслідковування поточного статусу виконання імітаційного моделювання («Очікує запуску», «В роботі», «Успішне», «Помилкове»).

Simulation Params – допоміжна залежна від *Simulation* сутність, яка створена для збереження усіх вхідних числових параметрів моделювання: кількості ітерацій, часової межі моделювання, вагових коефіцієнтів для матеріальних, часових витрат та якості згідно формули полінома Калмагорова-Габора, номінального часу проведення розподілу 1-ї роботи, номінального часу проведення агрегації 1-ї роботи, ймовірності виникнення проблем із якістю виконаної роботи, параметрів генерації пакетів робіт та самих робіт всередині пакетів при моделюванні.

Simulation Result – залежна від *Simulation* сутність, яка призначена для збереження даних про прогрес виконання моделювання, його вихідних файлів та інформації про помилки.

Output File – це сутність призначена для збереження інформації про вихідні

дані імітаційного моделювання у вигляді Microsoft Excel файлів. Завдяки тому, що СУБД PostgreSQL підтримує бінарний тип даних (BYTEA), вихідні файли можна зберігати прямо в базі даних та не виділяти під це сховище інших типів (таких як наприклад FTP).

Дана логічна модель даних була побудована за допомогою CASE-засобу «Allfusion ErWin Data Modeler», який надає безпрецедентно широкий набір інструментів з проєктування IDEF1X діаграм. Його особливістю є можливість одночасної побудови і фізичної моделі, яка буде виконана із врахуванням особливостей обраної СУБД. Переглянути фізичну схему даних, реалізовану безпосередньо в СУБД PostgreSQL можна у додатку А.

3.6 Опис внутрішньої структури програмного засобу

Розглянемо більш детально створювану моделювальну програму. Як було зазначено, її головною задачею є реалізація схеми імітаційного моделювання (схеми СМО), змодельованої на рис. 2.2. Але при цьому спектр функцій, які вона повинна надавати є значно ширшим.

Згідно з вимогами, зазначеними у підрозділі 3.4, даний програмний продукт повинен бути доступним із мережі та надавати доступ до своїх ресурсів за допомогою екранних форм. Це вкладається у стандартну парадигму клієнт-серверних програм побудованих за допомогою мови програмування Java, у яких є серверна оболонка для виклику процедур та ядро, яке виконує такі процедури.

Для спрощення розробки було використано декілька готових рішень та фреймворків. У якості базового фреймворку для програмного продукту було використано Spring Framework. Це дуже потужний фреймворк для побудови складних програм бізнес-рівня. Він має безліч готових рішень для роботи з БД, створення клієнт-серверних програм, моніторингом програми, розгортанням її на хостингу, автоматичної конфігурації, тощо. Для роботи із статисткою було використано математичне забезпечення бібліотеки Apache Math. Для отримання статистично надійних генераторів було використано бібліотеку Apache RNG. Для побудови інтерфейсу клієнтської частини програми було використано сучасну JavaScript бібліотеку React із комплектом компонентів Bootstrap 5.

Завдяки тому, що мова програмування Java є об'єктно-орієнтованою, уся внутрішня структура програми будується за допомогою класів. При

проектуванні об'єктно-орієнтовних програм прийнято використовувати діаграми класів, як основних блоків системи. Класи показують, які дані має кожен логічний блок та набір операцій, які він може з ними виконувати.

Діаграми класів є однією із найбільш використовуваних діаграм нотації UML (Unified Modeling Language). Вона є логічним поданням системи загалом, або її конкретної частини. На діаграмі відображають ґрунтовну інформацію про те, як система повинна функціонувати та який у неї внутрішній устрій, які у ній є залежності та взаємозв'язки між елементами. Як схема, вона відображає статичну структуру моделі системи в термінології класів об'єктно-орієнтованого програмування.

Завдяки тому, що багато із сьогоденних мов програмування є об'єктно-орієнтованими, UML діаграма класів стала простим та зрозумілим інструментом для моделювання програмного забезпечення не тільки для розробників, але й для бізнес-аналітиків та менеджерів, адже загальні риси створюваних компонентів залишаються незмінними незалежно від кінцевої реалізації. Також дана особливість використовується спеціалізованими CASE-засобами для автоматичної генерації програмного коду по сформованій діаграмі [42].

Розглянемо більш детально кілька найважливіших діаграм класів для імітаційного ядра, які було побудовано при створенні програмного продукту. Як видно зі схеми багатоканальної СМО рисунку 2.2, основою імітаційної моделі є пакети робіт, роботи та канали, які їх обробляють. Дані сутності є ключовими в роботі системи, тому їх було змодельовано в першу чергу (рис. 3.4).

Клас «Робота в пакеті» (ExecutionTask) має такі атрибути:

- id: унікальний ідентифікатор для збереження роботи в системі при проведенні моделювання;
- name: оригінальна назва послуги, яку було трансформовано у роботу при додаванні до пакету робіт;
- specialization: необхідна спеціальність спеціаліста для виконання даної роботи;
- executionTime, executionCost: числові параметри номінальних витрат матеріальних ресурсів та часу при виконанні роботи;
- executionProfit: числовий параметр прибутку, який отримує підприємство при виконанні даної роботи;
- priority: пріоритет даної роботи (використовується для черги розподілу).

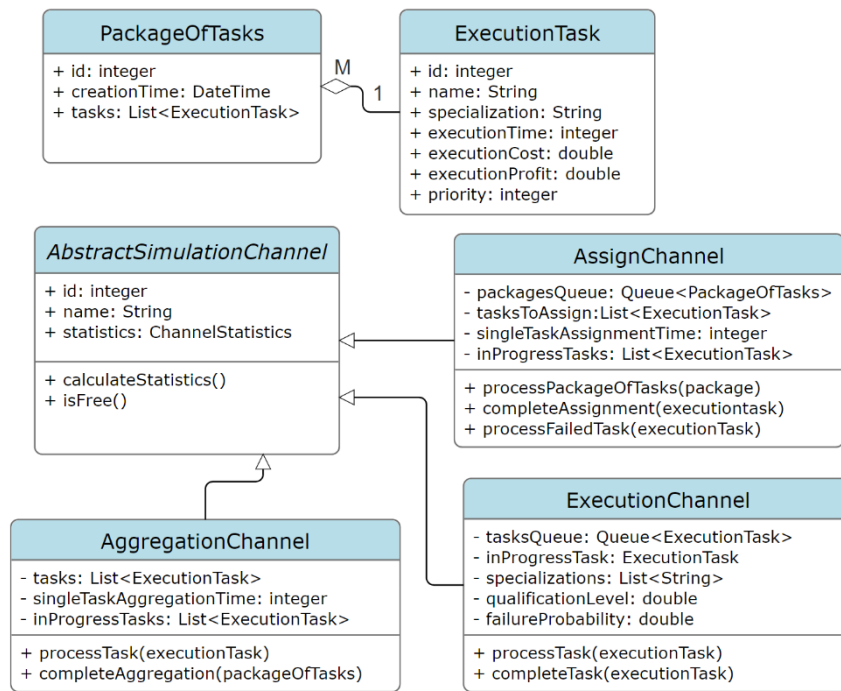


Рисунок 3.4 – Діаграма класів основних сутностей моделювання

Клас «Пакет робіт» (PackageOfTasks) містить в собі деяку встановлену генератором кількість робіт та має атрибути:

- id: унікальний ідентифікатор для збереження пакету в системі при проведенні моделювання;
- creationTime: час створення пакету в системі для відслідковування його шляху та часу його виконання;
- tasks: список робіт пакету, не допускається повторення робіт в рамках одного пакету.

Пакети робіт, які будуть надходити до системи, генеруються на етапі формування вхідних даних імітаційного алгоритму всередині подій надходження пакету, тому вони повністю визначенні, коли імітація запускається.

Для моделювання каналів було створено спеціальні сутності на основі базового класу AbstractSimulationChannel, який надає базові атрибути для ідентифікації та найменування каналу, а також для збору статистичних даних. У всіх типах каналів збираються такі дані, як час надходження роботи в канал, час її виходу, параметри розміру черги та кількість опрацьованих заявок. Це дозволяє отримати інформацію по інтенсивності використання каналу, яка на кінцевому етапі моделювання записується до вихідного результуючого файлу.

Клас «Розподільчий канал» (AssignChannel) представляє собою реалізацію

розподільчих каналів СМО, він має такі атрибути та операції:

- `packagesQueue`: атрибут, який є реалізацією черги пакетів робіт для розподілу у каналі; тут зберігаються тільки ті пакети робіт, жодної роботи із котрих ще не розподілили;
- `tasksToAssign`: активна черга робіт, які готові до наступного розподілу; сюди переміщуються усі роботи з пакету робіт, з якого було розподілено хоча б одну роботу;
- `singleTaskAssignmentTime`: числовий параметр моделювання який задає час, необхідний для розподілу однієї роботи;
- `inProgressTasks`: список робіт, які розподіляються у даний момент часу; дане значення пусте, якщо канал вільний;
- `processPackageOfTasks`: операція яка викликається при отриманні каналом нового пакету робіт; внутрішня логіка вирішує що треба зробити з пакетом, одразу додати всі його роботи до списку активних робіт `tasksToAssign` якщо канал вільний, чи помістити до окремої черги `packagesQueue`, якщо зайнятий або є попередні пакети.
- `completeAssignment`: операція, яка викликається подією закінчення виконання розподілу задач; внутрішня логіка видаляє всі пакети зі списку поточних `inProgressTasks` та бере наступні за потребою;
- `processFailedTask`: операція повторної обробки роботи, яка не пройшла перевірку якості.

Клас «Виконавчий канал» (`ExecutionChannel`) представляє собою модель виконавчих каналів СМО. Даний клас має в собі специфічні під дану задачу атрибути та операції:

- `tasksQueue`: черга заявок для виконання даним каналом, для виконання найбільш ефективних розподілів розподільчий канал призначає нову задачу виконавчому каналу тільки при умові що його черга пуста (тобто канал одразу після закінчення виконання поточної задачі перейде до призначеної без виконання сторонніх задач);
- `inProgressTask`: задача, яка виконується в даний момент часу, дане поле пусте якщо канал вільний;
- `specializations`: список спеціальностей, якими володіє канал для виконання робіт (трансформується з робітників автосервісу);

- `qualificationLevel`: числове значення кваліфікації даного виконавця-каналу;
- `failureProbability`: числове значення в діапазоні $[0;1]$, яке встановлює ймовірність неправильного виконання роботи, яку необхідно буде переробити;
- `processTask`: операція яка запускається після отримання події про призначення нової роботи на виконання від каналу розподілу;
- `completeTask`: операція яка виконується після отримання події про виконання завдання;

Клас «Канал агрегації» являє собою модель для реалізації агрегативних каналів, які є фінальною точкою обробки робіт та збору їх знову до пакету. Він має такі атрибути та операції:

- `tasks`: список робіт, які очікують агрегацію; даний список може містити роботи з різних пакетів, якщо виконавчі канали обробляли декілька пакетів паралельно;
- `singleTaskAggregationTime`: числовий параметр каналу, який показує необхідні часові витрати на агрегацію однієї роботи;
- `inProgressTasks`: список робіт, які агрегуються назад у пакет в даний момент часу;
- `processTask`: процедура, яка викликається кожен раз при завершенні виконання роботи у виконавчому каналі та переході роботи до агрегативного каналу; якщо отримана робота є останньою для формування пакету робіт, то виконується агрегація пакету (якщо канал в цей момент часу вільний);
- `completeAggregation`: фінальна процедура, яка виконується після отримання події про закінчення агрегації пакету; на даному етапі формується статистика про проходження пакету усіх етапів моделювання.

Управління процесом імітаційного моделювання було засновано на подійному принципі. Згідно даного принципу імітаційна система веде хронологічний облік подій, кожна з яких запланована на чітко встановлений моделювальний час. Для кожної події система має окремий обробник подій, який викликається при її настанні. Реакцією на деяку подію в системі може бути планування однієї чи декількох нових подій на інший час. В цілому, задачею розробника є ідентифікація та реалізація набору подій, які змінюють стан системи та визначенні логічних зв'язків між ними [29].

Діаграму класів для основних сформованих подій зображено на

рисунку 3.5. Вона включає в себе 8 класів, які наслідуються від базового класу `AbstractSimulationEvent`. Даний клас має лише одне поле для збереження інформації про час появи події в системі.

Подія `NewTasksPackageEvent` відповідає за надходження нового пакету робіт до системи. Набір всіх подій даного типу формується на етапі генерації вхідних даних до імітаційної системи і дані події є опорними при формуванні усіх інших подій. Після отримання даної події обробник передає отриманий пакет робіт до розподільчого каналу, який вже може сформувати подію `AssignTasksEndEvent` про їх розподіл.

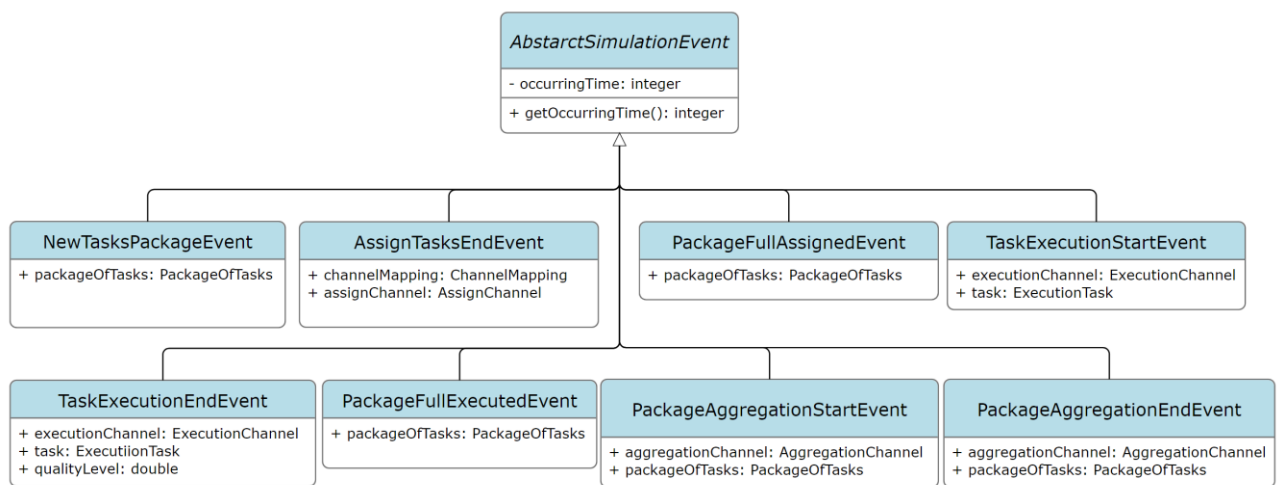


Рисунок 3.5 – Діаграма класів для сутностей подій моделювання

Подія `AssignTasksEndEvent` відповідає за закінчення розподілу встановленого набору робіт з пакетів, які були отримані раніше. В одному розподілі можуть бути роботи з різних пакетів для збільшення завантаженості виконавчих каналів. У атрибуті `channelMapping` записуються пари з виконавчого каналу та призначеної йому роботи. Дана подія породжує набір подій `TaskExecutionStartEvent` згідно призначенням. Якщо для отриманого раніше пакету робіт всі роботи було розподілено, то породжується подія `PackageFullAssignedEvent`.

Подія `PackageFullAssignedEvent` є утилітарною для збору статистики по часу розподілу робіт конкретного пакету та збереження даної статистики.

Подія `TaskExecutionStartEvent` відповідає за перенесення окремої роботи до виконавчого каналу згідно призначення та виконання розрахункових дій щодо

часу його блокування. Дана подія породжує подію з завершенням виконання роботи `TaskExecutionEndEvent`.

Подія `TaskExecutionEndEvent` відповідає за встановлення стану роботи як виконаної та збору статистики щодо виконання (час виконання, рівень якості, перебування в чергах, тощо). Після цього робота переноситься на виконання до агрегативного каналу, де буде оцінено її якість. Говорячи про встановлення рівня якості виконаної роботи, то це значення будується на основі параметру `failureProbability` самого каналу (яке встановлюється на етапі формування конфігурації моделювання) та його кваліфікації. Якщо дана робота була останньою виконаною з пакета, то породжується подія `PackageFullExecutedEvent`.

Подія `PackageFullExecutedEvent` є утилітарною для збору корисної статистики щодо проведення виконання робіт з конкретного пакету.

Подія `PackageAggregationStartEvent` відповідає про початок агрегації отриманих робіт з пакету. Дана подія породжується системою коли усі роботи з деякого пакету було виконано та всі вони пройшли успішно перевірку на якість. Із розрахунку на час виконання агрегації ця подія породжує нову подію `PackageAggregationEndEvent` про її завершення

Подія `PackageAggregationEndEvent` є фінальною для пакету робіт. На її етапі виконується повний збір та збереження статистики по обробці пакету системою.

Розглянемо роботу подій у моделювальній системі на основі невеликого прикладу (рис 3.6). Нехай до системи в момент часу 500 надійшов пакет із п'ятьма роботами, при цьому система вже має інші роботи для виконання.

В той самий момент часу 500 у системі з виконавчих 5-ти каналів є 3 вільних, які підходять під виконання отриманих робіт. Таким чином, розподільчим каналом запускається розподіл трьох робіт, який коштує 50 умовних одиниць часу за кожну роботу (150 за всі). Про завершення розподілу повідомляє подія `AssignTasksEndEvent`, яка з'являється у момент часу 650. Розподілені роботи T1, T2, T3 одразу переміщуються до виконавчих каналів, про що сигналізує три відповідних події `TaskExecutionStartEvent`.

Припустимо для прикладу, що в цей момент часу інші 2 виконавчі канали вже звільнилися (дані події на рисунку не відображено), тоді одразу ж запускається розподіл решти робіт з пакету витрачаючи 100 одиниць часу

сумарно. Про закінчення цього розподілу сигналізує подія AssignTasksEndEvent, яка з'явилася в момент часу 750. Роботи T4 та T5 також переміщуються на виконання. Так як пакет був повністю розподілений, то також породжується подія PackageFullAssignedEvent.

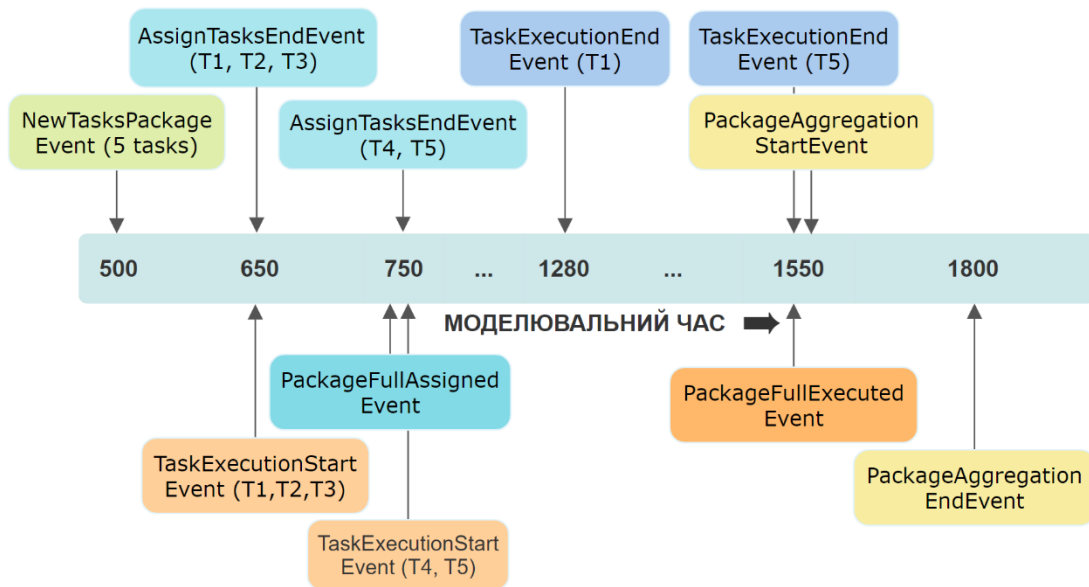


Рисунок 3.6 – Часова діаграма обробки подій при отриманні нового пакету робіт

Виконавчі канали займаються обробкою робіт T1-T5 згідно своїх параметрів та по виконанню продукують події TaskExecutionEndEvent. Коли остання робота буде виконана та перевірена на якість, виконання вважається закінченим, що продукує подію PackageFullExecutedEvent. Потім запускається агрегація робіт знову до пакету, яка триває із розрахунку 50 одиниць часу на кожну роботу (в сумі 250 часових одиниць). Із появи події PackageAggregationEndEvent виконання пакету робіт системою завершується.

3.7 Опис клієнтської частини програмного засобу

Побудова адекватної імітаційної моделі є важливою частиною з розробки імітаційного продукту, але без можливості зручно та просто конфігурувати та запускати моделювання такий продукт потенційно витрачає більше часу адміністратора, що нівелює його корисність. Коли внутрішнє програмне

забезпечення та інструменти розроблені з урахуванням потреб користувачів, співробітники можуть працювати ефективніше та результативніше. Це може призвести до підвищення продуктивності всередині автосервісу.

Для вирішення цієї проблеми біло прийнято рішення про розробку користувацького інтерфейсу на базі веб-форм. Можливості сучасних мов програмування, в тому числі HTML 5, CSS 3, JavaScript, надають можливість будувати складні багатосторінкові веб-сайти достатньо швидко, дозволяючи концентрувати увагу на вирішенні бізнес-задач, а не на створенні окремих компонентів UI (User Interface). Для створення клієнтського інтерфейсу було використано фреймворк React. Даний фреймворк надає можливість будувати інтерфейс із застосуванням компонентного підходу, тобто побудові сторінки як набору окремих логічних частин, кожна з яких є окремим будівельним блоком із своєю внутрішніми даними та поведінкою. Також було використано готову бібліотеку CSS стилів Bootstrap 5, що на порядки збільшує швидкість розробки за рахунок виключення написання сотень строк однотипного коду.

Розглянемо найважливіші екранні форми програмного імітаційного продукту. Для створення нового прогону моделювального алгоритму було розроблено спеціальну сторінку (рис. 3.7, рис. 3.8) для внесення усіх бажаних параметрів моделювання, таких як межі моделювання, його тип, кількість ітерацій, вагові коефіцієнти, ймовірність відмови при виконанні роботи (проведення неякісної роботи), кількості пакетів робіт та самих робіт у пакетах, а також списків робіт та виконавців.

Після визначення усіх необхідних параметрів необхідно натиснути на кнопку «Запустити моделювання». Дана дія запустить перевірку усіх встановлених параметрів та відправить сформовану конфігурацію на сервер. Після цього буде здійснено автоматичний перехід на сторінку очікування виконання моделювання (рис. 3.9).

Після завершення моделювання, буде здійснено автоматичний перехід на сторінку перегляду результатів (рис. 3.10). На даній сторінці для кожного створеного або проведеного моделювання можна переглянути його вхідні параметри, задіяну кількість ітерацій, згенеровану сумарну кількість пакетів робіт та самих робіт, а також завантажити архів з результатами, якщо моделювання було завершено успішно. У випадку помилки буде показано її короткий зміст.

ІС Автосервіс + Нове моделювання Результати Списки робіт Виконавці ДЧ Дмитро Чоломбисько Вийти

Заповніть будь ласка поля для створення нової імітації

Параметри моделі:

Верхня межа моделювання:

100000

Ця межа буде використана для визначення максимального часу отримання пакету робіт.

Тип моделювання:

Комплексне
 Одиничне

Кількість експериментів для комплексного моделювання:

100

Більша кількість циклів підвищить точність проведених досліджень, але збільшить час проведення імітації.

Вагові коефіцієнти моделі:

Матеріальних витрат:	0,2	Часу:	0,15	Якості:	0,15
Квадрату матеріальних витрат:	0,2	Квадрату часу:	0,15	Квадрату якості:	0,05
Напряму матеріальні витрати та час:	0,05				

Рисунок 3.7 – Екранна форма створення нового моделювання (частина 1)

Вірогідність відмови:

0,05

Ймовірність виникнення роботи низької якості, яку необхідно переробити.

Параметри пакетів робіт:

Кількість пакетів робіт в імітації:

від 10 до 30 за Рівномірним законом

Кількість робіт в одному пакеті:

від 10 до 30 за Рівномірним законом

Для побудови пакетів використовувати список робіт:

Стандартний список робіт (56)

Параметри виконавців:

Використовувати список виконавців:

Стандартний список виконавців (10)

Запустити моделювання

Рисунок 3.8 – Екранна форма створення нового моделювання (частина 2)

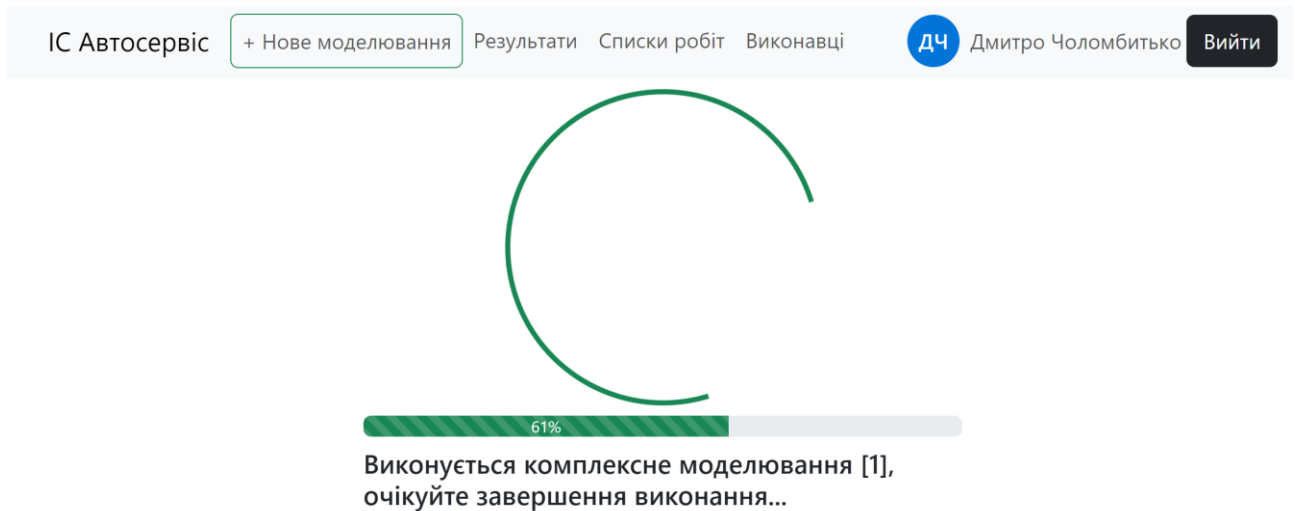


Рисунок 3.9 – Екранна форма прогресу проведення моделювання

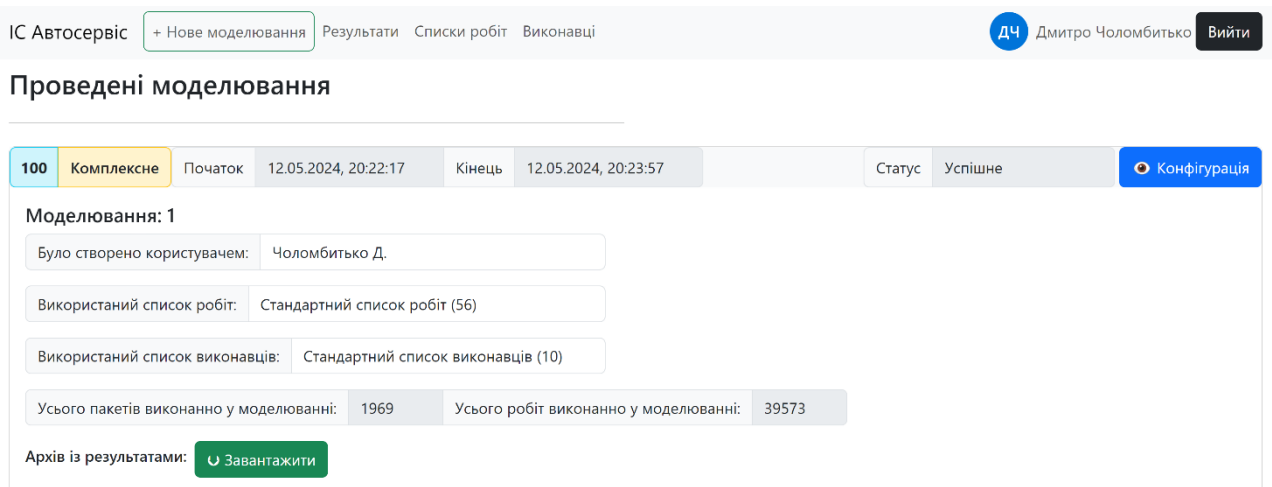


Рисунок 3.10 – Екранна форма перегляду результатів моделювання

Загалом, схему взаємодії клієнтської частини програмного засобу із серверною можна представити у вигляді діаграми послідовності, зображеної на рис. 3.11. На стороні браузера користувач (адміністратор сервісу) формує конфігурацію нового прогону імітаційного моделювання. Ця конфігурація валідується та відправляється на серверну сторону, де потрапляє на обробку до

контролеру `SimulationController`. Дана конфігурація зберігається як новий екземпляр моделювання у базі даних за допомогою методів класу `SimulationDAO`, після чого до браузера відправляється ідентифікатор нового моделювання для відслідковування результатів. Паралельно із цим, в асинхронному режимі запускається саме моделювання. Якщо воно є комплексним, то відбувається циклічний виклик процедур з генерації вхідних даних, які потім передаються на вхід до нового екземпляра імітаційного ядра `Simulator` де воно запускається на виконання для отримання результатів. Після проведення усіх циклів моделювання усі вихідні дані зберігаються у БД.

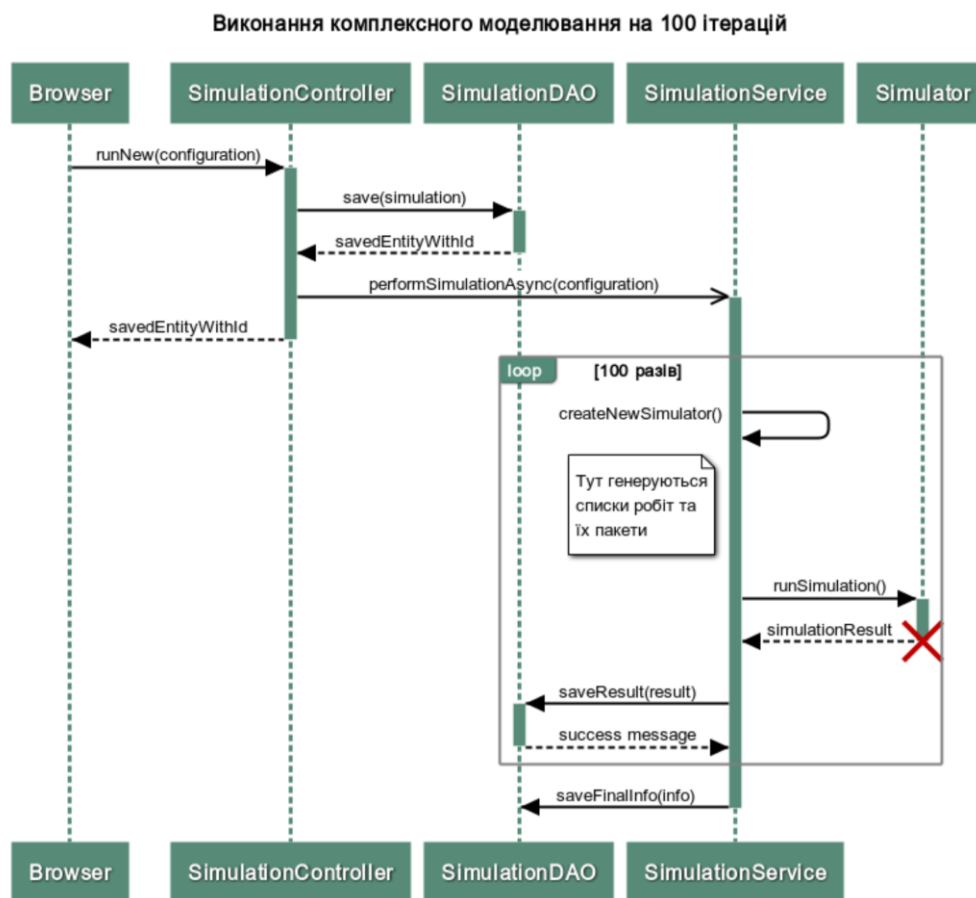


Рисунок 3.11 – Діаграма послідовності для процесу проведення моделювання

Рисунки користувацького інтерфейсу для створення та вибору списків робіт, списків виконавців та спеціальностей не є достатньо важливими для безпосереднього висвітлення у роботі, тому їх наведено у додатку А.

3.8 Проведення імітаційних експериментів

Для зручності обробки результатів моделювання, яке може бути запущено за допомогою користувачького інтерфейсу на рис. 3.8 було прийнято рішення про використання у якості вихідного формату даних Excel файли.

Їх основною перевагою є зрозумілість як для команди розробки продукту моделювання, так і для кінцевих користувачів у ролі адміністраторів та власників підприємства.

Додатковими корисними особливостями є можливість побудови графіків по великим масивам даних та можливість використання великої кількості вже вбудованих у редактор формул. Візуалізація інформації значно спрощує її сприйняття та збільшує продуктивність при аналізі результатів моделювання.

Кінцевий результат моделювання може бути завантажено у вигляді архіву із Excel файлами. При цьому, в залежності від типу моделювання в архіві буде присутнім або 1 файл з описом одиничного моделювання, або список файлів при комплексному моделюванні. Також, при комплексному моделюванні присутній спеціальний файл результату агрегації, в якому присутні статистичні дані про усі ітерації моделювання.

В файлі з описом одиничного моделювання присутній опис вхідних даних моделювання, загальної інформації, такої як вагові коефіцієнти моделі, межі моделювання, кількість каналів (рис. 3.12), статистики роботи кожного каналу та параметрів обробки пакетів.

У файлі результату агрегації присутня статистична інформація по кожному окремому параметру моделювання (час обробки пакету, час перебування в черзі, розмір черги, якості виконання роботи, витратах на виконання роботи, прибутку).

Детальні рисунки сторінок з цих Excel файлів наведено у графічному додатку А.

Для проведення експериментів було визначено такі параметри конфігурації моделювання, щоб максимально завантажувати виконавців, при цьому щоб не виникало перевантаження та накопичення пакетів в черзі (таблиця 3.1). Така конфігурація дозволить максимально показати можливості моделі по розподілу робіт, а також показати можливості розробленого програмного продукту до аналізу завантаженості виконавців, розміру черги,

продуктивності, тощо. Параметри робіт для даного моделювання наведено окремо у табличному додатку Б.

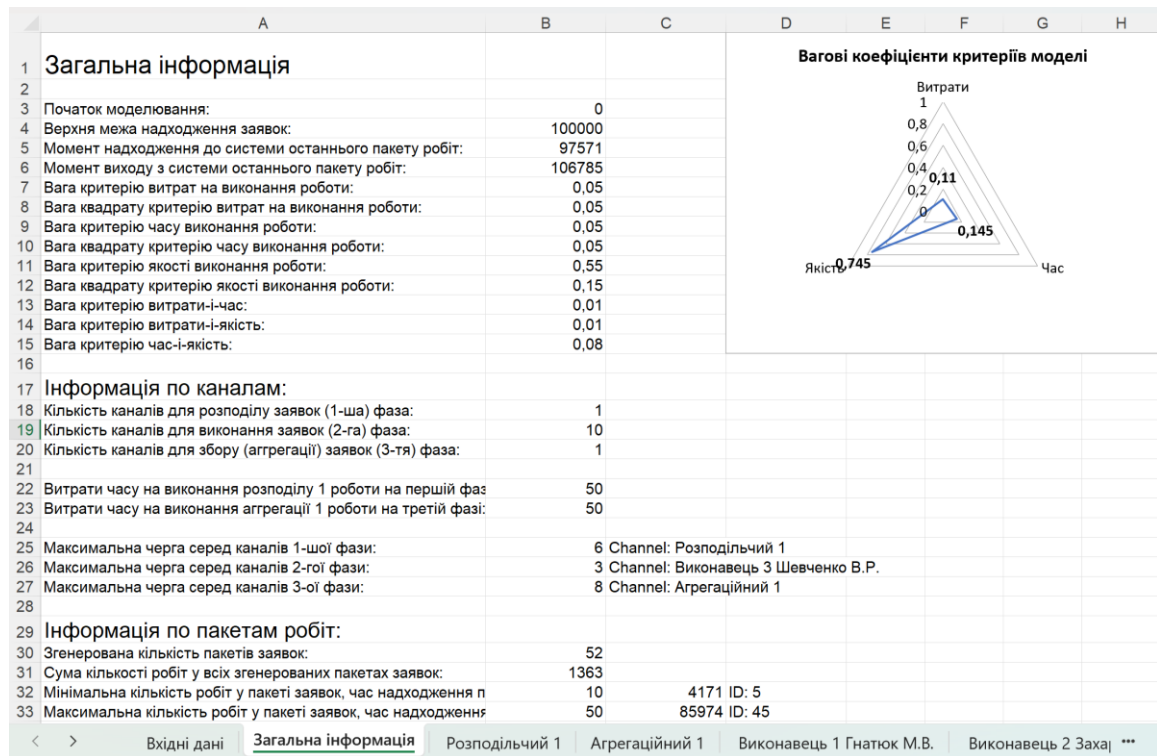


Рисунок 3.12 – Excel сторінка опису параметрів моделювання

Таблиця 3.1 – Конфігурація для проведення імітаційного моделювання

Параметр	Значення та закон розподілу
Кількість ітерацій для досягання необхідної точності, кількість циклів	300
Верхня межа надходження пакетів робіт, час. од.	20 000
Кількість пакетів робіт за 1 цикл моделювання, шт	від 25 до 60, нормальний
Кількість робіт у 1 пакеті робіт, шт	від 15 до 40, нормальний
Кількість різноманітних спеціалізацій, шт	2
Кількість різноманітних робіт для формування пакетів робіт, шт	56 (28 спеціалізації 1 та 28 спеціалізації 2), див дод. Б
Кількість розподільчих каналів, шт	1
Кількість каналів-виконавців, шт	10
Кількість каналів агрегації пакетів робіт, шт	1

В якості опорного вагового коефіцієнту було обрано максимізацію якості виконання робіт (рисунок 3.13). Більш детальну схему розподілу коефіцієнтів для занесення до полінома Колмогорова-Габора наведено у таблиці 3.2.

Таблиця 3.2 – Вагові коефіцієнти критеріїв моделювання

Параметри вагових коефіцієнтів	Значення
Вага критерію витрат на виконання роботи:	0,02
Вага квадрату критерію витрат на виконання роботи:	0,02
Вага критерію часу виконання роботи:	0,02
Вага квадрату критерію часу виконання роботи:	0,02
Вага критерію якості виконання роботи:	0,55

Продовження таблиці 3.2 – Вагові коефіцієнти критеріїв моделювання

Вага квадрату критерію якості виконання роботи:	0,25
Вага критерію витрати-і-час:	0,01
Вага критерію витрати-і-якість:	0,01
Вага критерію час-і-якість:	0,1

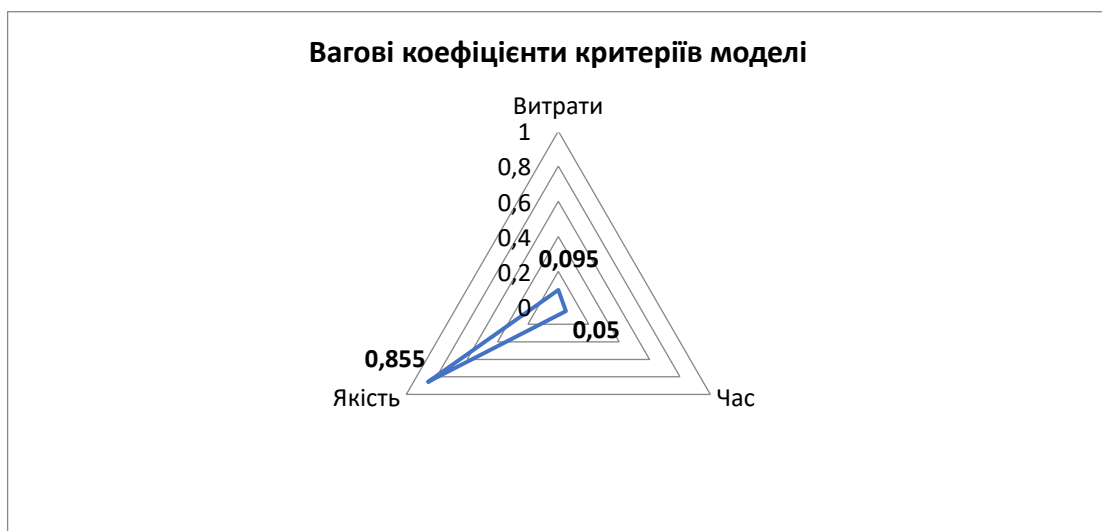


Рисунок 3.13 – Візуалізація вагового коефіцієнту спрямованого на якість

Для отримання більш широкого бачення ситуації з розподілу робіт та

впливу різної кваліфікації виконавців на результати моделювання, було прийнято рішення про використання виконавців усіх кваліфікацій (рисунок 3.14). Менше значення на графіку відповідає гіршому значенню кваліфікації.

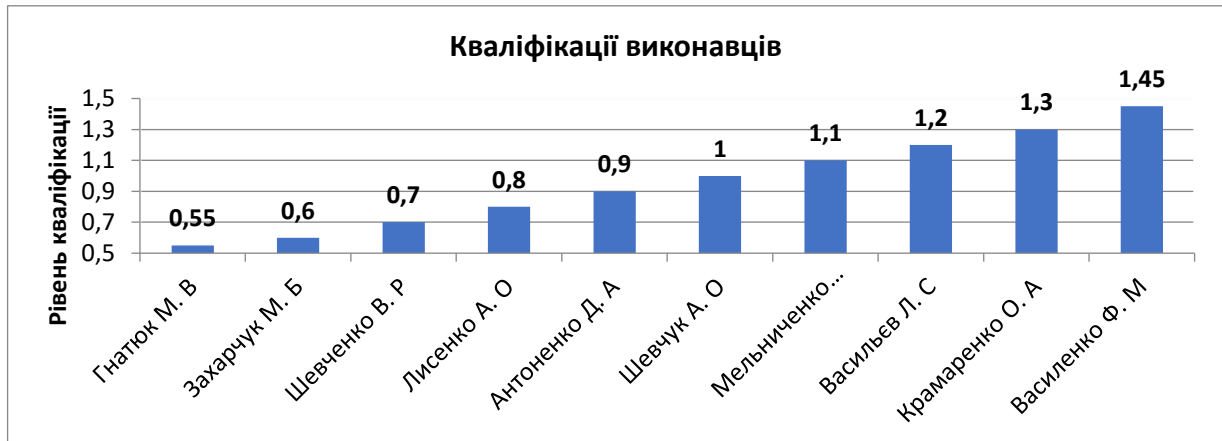


Рисунок 3.13 – Графік кваліфікації обраних для імітації виконавців

Після завершення імітаційного моделювання маємо такі результати, що наведено у таблиці 3.3.

Таблиця 3.3 – Результати проведення моделювання

Вимірювана величина	Результат
Кількість оброблених пакетів, шт	12706
Кількість оброблених робіт, шт	348766
Середній час виконання 1 пакету, час. од.	1591 ± 72
Середній розмір черги пакетів каналу розподілу	2,82 ± 0,09
Середній час перебування в черзі, час. од.	708 ± 51
Середній прибуток з виконаного пакету, ум од	23575 ± 666
Середня якість виконання роботи, %	66,2 ± 2,1

Для багатьох згаданих у таблиці 3.3 параметрів в агрегаційному звіті наведено більш детальну статистику, яка дає змогу провести ґрунтовний аналіз характеристики та використати отриману інформацію для прийняття управлінських рішень. Характеристика середньої кількості виконаних робіт кожним окремим виконавцем (рис. 3.14) дає змогу оцінити його важливість та

вплив його кваліфікації на продуктивність підприємства.

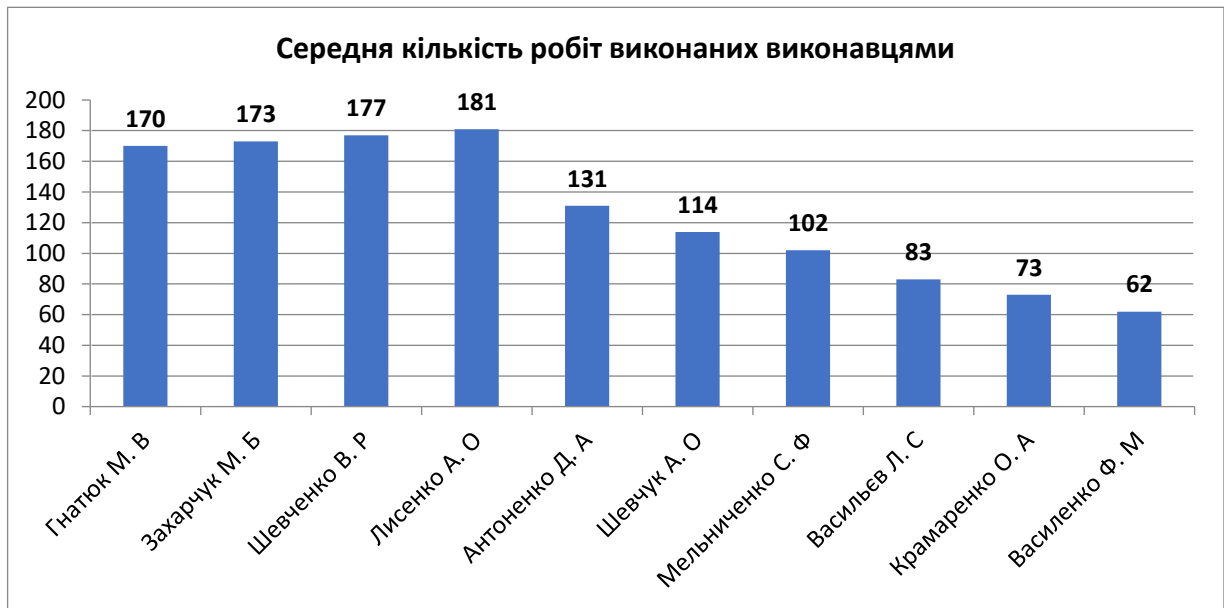


Рисунок 3.14 – Графік розподілу середньої кількості виконаних робіт по виконавцям

З графіку видно, що найбільш продуктивними є виконавці з низькою та середньою кваліфікацією, що може пояснюватися тим, що її рівня достатньо для виконання на гарному рівні якості більшості з отримуваних задач, але при цьому за менші витрати часових та матеріальних ресурсів ніж у висококваліфікованих виконавців.

При цьому, розподіл виконаних робіт по різних спеціалізаціях є майже однаковим, що вказує на достатньо якісний рівень генерації вхідних пакетів для їх балансування.

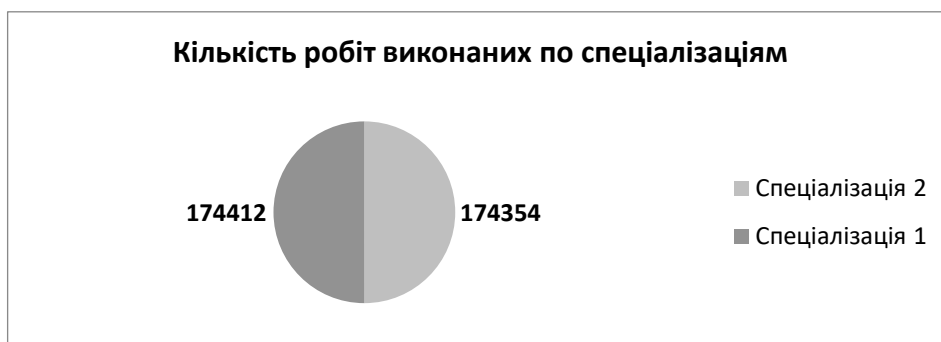


Рисунок 3.15 – Графік розподілу робіт по спеціальностям

Час виконання роботи суттєво впливає на ефективність підприємства, аналіз даного показника надає можливість більш точно проводити календарне планування по виконанню клієнтських замовлень. Із графіку на рис. 1 видно, що найчастіше послуги в пакеті при заданих умовах будуть виконані за проміжок часу в 1000-1500 одиниць, що становить приблизно 17 середніх робіт. Часові одиниці було узято як приблизне наближення до виконання робіт у хвилинах.

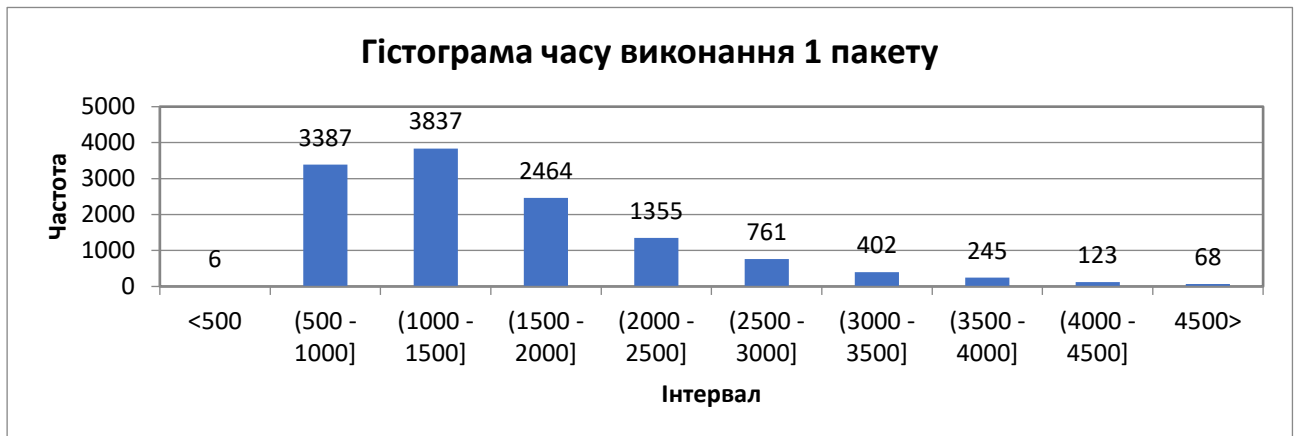


Рисунок 3.16 – Графік розподілу часу виконання пакету з врахуванням черги

При цьому аналіз черги (рис 3.17) показує, що при виконанні умови високої завантаженості виконавців, вона може сягнути 4-6 пакетів робіт у 22% випадках. Даний показник є не сильно високим, але це все одно може призвести до невдоволення клієнтів через занадто великий проміжок очікування. Подолати дану проблему можна за допомогою запису до черги очікування, чи зміни конфігурації виконавців (але це може призвести до більших матеріальних витрат на збільшення штату).

В даній роботі було прийнято, що на прибуток від виконання пакету робіт та на загальний прибуток підприємства особливим чином впливає якість виконаної роботи. Якість залежить від можливостей клієнта за неї додатково платити та кваліфікації робітника, який дану роботу виконує.

Таким чином було прийнято, що за виконання роботи із застосуванням високих стандартів якості, підприємство встановлює ціну в декілька разів вищу ніж за роботу, яка була виконана з низькою якістю. Також було встановлено, що імітація різних запитів на якість від клієнтів також генерується випадковим чином за рівномірним законом розподілу.

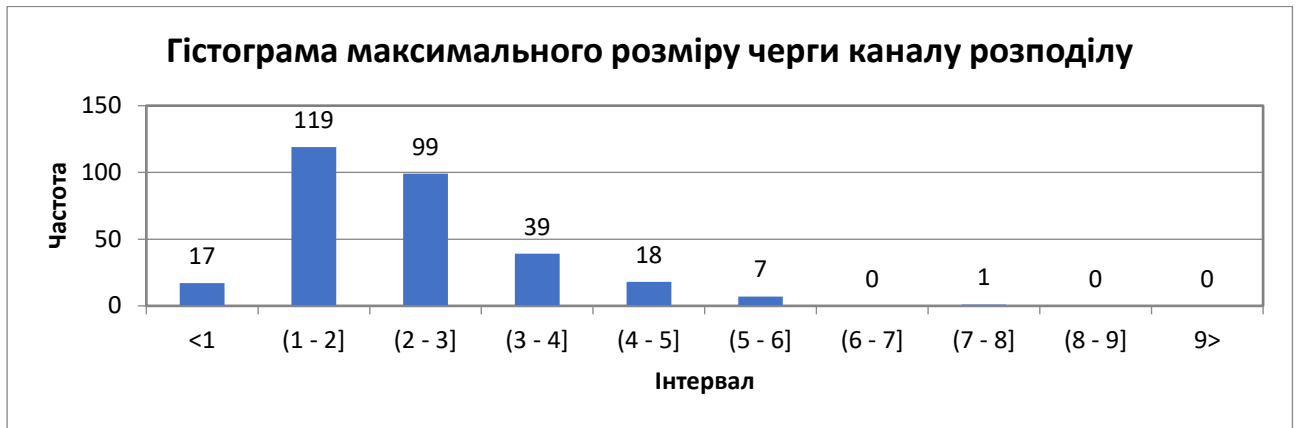


Рисунок 3.17 – Розподіл максимального розміру черги каналу розподілу

Для даного моделювання розподіл пакетів робіт за доходом є піковим для інтервалу від 20000 до 24000 умовних одиниць (рис 3.18), але даний розподіл є цікавим у динаміці порівняно до інших вагових коефіцієнтів. У даній конфігурації моделювання було обрано саме якість як основний напрям оптимізації.

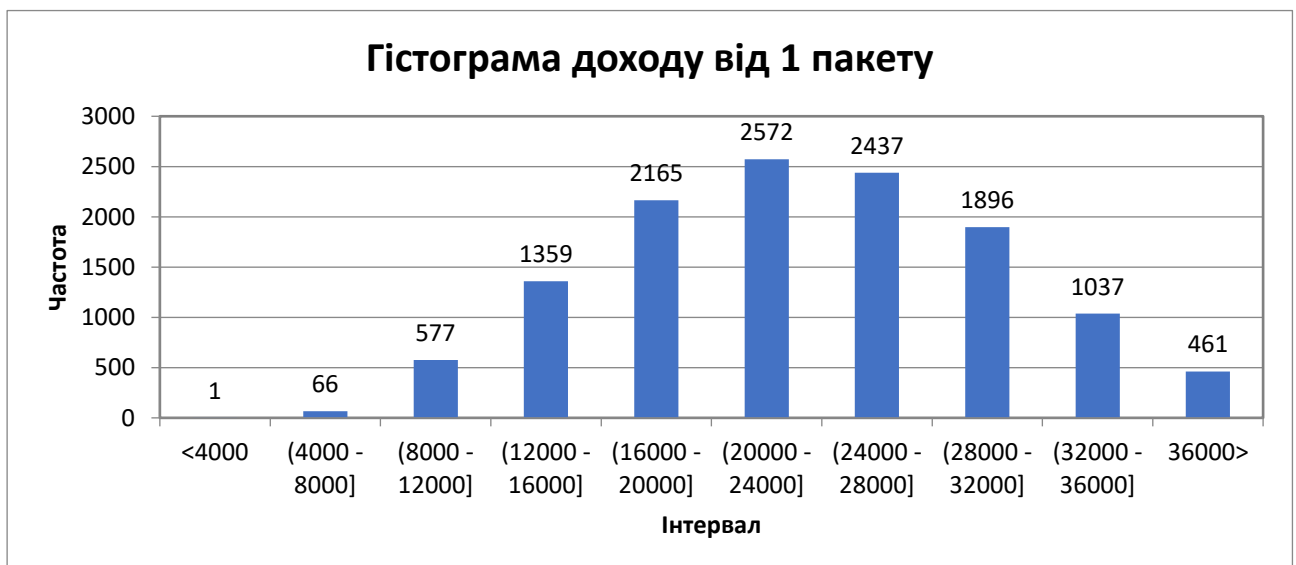


Рисунок 3.18 – Графік розподілу доходу від виконаного пакету робіт при оптимізації якості виконання роботи

Для порівняння було обрано напрям оптимізації за часом виконання роботи (сумарний коефіцієнт часу = 0,855). Якраз при швидкій та некваліфікованій роботі якість є найбільш вразливою характеристикою. Усі інші параметри моделювання із таблиці 3.1 було збережено такими самими.

Результат для порівняння розподілу доходу наведено на рис. 3.19. Середній показник прибутку зменшився на 18% (19901 проти 23575) і середнє значення якості впало на 11% (58,9 проти 66,2).



Рисунок 3.19 – Графік розподілу доходу від виконаного пакету робіт при оптимізації часу виконання роботи

При цьому варто зазначити, що середній час виконання 1 пакету робіт зменшився на 17% (1365 проти 1591), що позитивно впливає на можливу кількість виконаних робіт та збільшення об'єму виконаних замовлень.



Рисунок 3.20 – Графік розподілу часу виконання 1-го пакету при орієнтуванні на час виконання пакету

Таким чином, даний програмний продукт дозволяє проводити комплексні дослідження для визначення ефективної структури і критичних точок підприємства, напрямків його можливого розвитку.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було отримано рішення науково-практичного завдання підвищення ефективності технологій системного проєктування та реінжинірингу технологічних систем (ТС) за рахунок розробки аналітико-імітаційної моделі процесів розподілу пакетів робіт, що виконуються ними.

Проведено аналіз сучасного стану проблеми оптимізації технологічних систем, методів їх проєктування та реінжинірингу. За результатами аналізу була встановлена актуальність задачі багатокритеріального розподілу пакетів робіт для оптимізації структурної конфігурації ТС.

Для реалізації розробки за базові було обрано метод аналізу ієрархій, задачі про призначення, календарного планування та оптимізації розміщення виробництва. Виходячи з обмеженості обраних методів було прийнято рішення щодо розширення класичної задачі про призначення та розробці аналітико-імітаційної моделі для її розв'язання.

В ході розробки математичної моделі було обрано та формалізовано локальні критерії для оптимізації розподілу за часом, витратами та якістю розподілу та виконання пакетів робіт. З врахуванням стохастичного характеру потоку пакетів і часу їх виконання запропонована формалізація процесу функціонування ТС з використанням апарату імітаційного статистичного моделювання систем масового обслуговування.

Розроблений моделювальний алгоритм реалізовано в програмному середовищі Java у вигляді web-додатка для локальної мережі та мережі Інтернет. У якості цільової предметної області було використано галузь сервісного обслуговування автомобілів. За результатами проведених експериментів були визначені оцінки показників розподілу робіт по виконавцям та спеціалізаціям, середнього часу виконання пакетів робіт, максимального розміру черг та прибутку.

Наукова новизна розробленої моделі полягає в розробці аналітико-імітаційної моделі процесів розподілу та виконання пакетів робіт в ТС, яка дозволяє оптимізувати розподіл стохастичних потоків пакетів робіт за показниками матеріальних (фінансових), часових витрат і якості їх виконання.

Отримані результати можуть бути застосовані для оптимізації розподілу

пакетів робіт серед виконавців на різноманітних виробничих підприємствах, в яких проводиться реінжиніринг чи системне проектування бізнес-процесів технологічних систем. Реалізовані компоненти орієнтовані на їх спільну взаємодію та повну автоматизацію процесу, що суттєво відрізняє їх від існуючих на ринку систем-конкурентів.

Практичне використання результатів роботи дозволить скорочувати витрати на створення й експлуатацію ТС, покращувати їх функціональні характеристики за рахунок оптимізації розподілу стохастичних потоків пакетів робіт з урахуванням показників витрат, часу та якості їх виконання.

За матеріалами кваліфікаційної роботи було опубліковано розділ у закордонній колективній монографії [14], статтю у періодичному фаховому виданні категорії В «Вісник Харківського національного автомобільно-дорожнього університету» [18], тези доповідей у матеріалах чотирьох науково-практичних конференцій [15–17, 43].

Напрямами подальших досліджень можуть бути вдосконалення аналітико-імітаційної моделі для врахування залежності між роботами пакетів та розробка методів для розв'язання задач оптимізації розподілу пакетів робіт великої розмірності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Чорней Н. Б., Чорней Р. К. Теорія систем і системний аналіз. посіб. Київ : МАУП, 2005. 256 с.
2. Посилкіна О. В. Аналіз діяльності підприємства. Харків : НФаУ, 2019. 398 с.
3. ДСТУ 2470-94. Надійність техніки. Системи технологічні. Терміни та визначення. Чинний від 1995-01-01. Вид. офіц. Київ. 16 с.
4. Nguyen M.-N., Do N.-H. Re-engineering assembly line with lean techniques. *Procedia CIRP*. 2016. Vol. 40. P. 590–595. URL: <https://doi.org/10.1016/j.procir.2016.01.139> (date of access: 28.03.2024).
5. Білявський В. М. Реінжиніринг в системі управління міжнародною конкурентоспроможністю підприємства // Вісник Харківського національного університету імені В. Н. Каразіна. Серія : Економічна. - 2017. - Вип. 93. - С. 7-17.
6. Буй Ф. Л., Беляєв Л. В. Методи і технології реінжинірингу інформаційних систем. *Інформаційні технології в економіці і управлінні*. 2019. № 1. С. 145–152.
7. Вислоух С. П., Піпко А. В. Моделювання та оптимізація завантаження обладнання виробничих систем з використанням структурно-логічного методу. *Вісник НТУУ КПІ*. 2007. № 34. С. 117–124.
8. Соколов С. В. Теорія систем і системний аналіз. Суми : Сум. держ. ун-т, 2020. 171 с.
9. Ладієва Л. Р. Оптимізація технологічних процесів : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2023. 160 с.
10. Research on the modelling and development of flexibility in production system design phase driven by digital twins / L. Zhang et al. *Applied sciences*. 2022. Vol. 12, no. 5. P. 2537. URL: <https://doi.org/10.3390/app12052537> (date of access: 28.03.2024).
11. Дудукалов Ю. В. Методи оптимізації технологічних систем технічного обслуговування та ремонту засобів транспорту. *Вісник ХНАДУ*. 2011. № 53. С. 24–32.
12. Beskorovainyi V., Imanhulova Z. Technology of large-scale objects system optimization. *ECONTECHMOD: an international quarterly journal*. 2017. Vol. 06, no. 4. P. 3–8.

13. Лавров Є. А., Перхун Л. П. Математичні методи дослідження операцій : навч. посіб. Суми : Сум. держ. ун-т, 2017. 212 с.
14. Bezkorovainyi V., Bezuhla H., Cholombytko D. Mathematical models of the cyclic work package distribution task // Innovative integrated computer systems in strategic project management: Collective monograph edited by I. Linde. European University Press. Riga: ISMA, 2022. P. 7-15. URL: <https://mmp-conf.org/documents/archive/monography2022.pdf> (дата звернення 24.05.2024 р.).
15. Безкоровайний В.В., Безугла Г.Є., Чоломбитько Д.В. Формалізація динамічної багатокритеріальної задачі розподілу пакетів робіт // Міжнародна науково-практична конференція «Інтелектуальні інформаційні системи в управлінні проектами та економіці в умовах воєнного стану»: Праці. Харків: ХНУРЕ, 2022. С. 37-39.
16. Безкоровайний В. В., Безугла Г. Є., Чоломбитько Д. В. Математична модель багатокритеріальної задачі розподілу комплексів робіт // Міжнародна науково-практична конференція “Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку” / Збірник тез доповідей (м. Харків, 15 березня 2022 р.). Харків. 2022. С. 60–62.
17. Чоломбитько Д. В., Безкоровайний В. В. Модифікація моделі процесу розподілу та виконання пакетів робіт при оптимізації технологічних систем // Комп’ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві. Матеріали всеукр. наук.-практ. конф. здобувачів вищої освіти і молодих учених. Харків, ХНАДУ, 2023. С. 12–15. URL: <https://dspace.khadi.kharkov.ua/handle/123456789/17765> (дата звернення: 29.12.2023).
18. Безкоровайний В. В., Чоломбитько Д. В. Моделювання процесу багатокритеріального розподілу та виконання пакетів робіт при проектуванні технологічних систем. *Вісник Харківського національного автомобільно-дорожнього університету*. 2024. Т. 1, № 104. С. 7–14.
19. Burkard R. E. Selected topics on assignment problems. *Discrete applied mathematics*. 2002. Vol. 123, no. 1-3. P. 257–302. URL: [https://doi.org/10.1016/s0166-218x\(01\)00343-2](https://doi.org/10.1016/s0166-218x(01)00343-2) (date of access: 28.03.2024).
20. Hospital layout design renovation as a Quadratic Assignment Problem with geodesic distances / C. Cubukcuoglu et al. *Journal of building engineering*. 2021. Vol. 44. P. 102952. URL: <https://doi.org/10.1016/j.jobe.2021.102952> (date of access:

28.03.2024).

21. Творошенко І. С. Технології прийняття рішень в інформаційних системах : навч. посіб. Харків : ХНУРЕ, 2021. 120 с. URL: <https://doi.org/10.30837/978-966-659-294-4> (дата звернення: 11.02.2024).

22. Теслюк В. М., Загарюк Р. В. Методи багатокритеріальної оптимізації. Львів : Вид-во Нац. ун-ту "Львів. політехніка", 2012. 65 с.

23. Волошина С. В. Економічний механізм управління ефективністю використання матеріальних ресурсів підприємства. *Економіка і суспільство*. 2016. № 3. С. 159–165.

24. Урманов Ф. Ш. Особливості підвищення ефективності використання матеріальних ресурсів підприємства. *Формування ефективної моделі розвитку підприємства в умовах ринкової економіки*. 2016.

25. Дунда С., Рибачук-Ярова Т., Тюха І. Тайм-менеджмент як напрям підвищення ефективності операційної стратегії підприємства. *Економіка та суспільство*. 2022. № 42. URL: <https://doi.org/10.32782/2524-0072/2022-42-68> (дата звернення: 28.03.2024).

26. Вітлінський В. В. Моделювання економіки : навч. посіб. Київ : КНЕУ, 2003. 411 с.

27. Биба В. В., Теницька Н. Б. Якість продукції як чинник конкурентоспроможності підприємства. *Економіка і суспільство*. 2017. № 12. С. 171–177.

28. Задоров В. Б. Підхід до створення технології попереднього системного проектування комп'ютерних інформаційних систем підприємств. *Управління розвитком складних систем*. 2010. № 1. С. 56–63.

29. Томашевський В. М. Моделювання систем: підруч. для студентів ВНЗ. Київ : Вид. група ВНУ, 2007. 352 с.

30. Киричок Т. Ю. Алгоритм розв'язання багатокритеріальної задачі вибору показника зношування банкнот за допомогою функції корисності. *Наукові вісті НТУУ "КПІ"*. 2013. № 1. С. 68–76.

31. Beskorovainyi V. Combined method of ranking options in project decision support systems. *Innovative technologies and scientific solutions for industries*. 2020. No. 4 (14). P. 13–20. URL: <https://doi.org/10.30837/itssi.2020.14.013> (date of access: 28.03.2024).

32. Пустовойтов П. Є. Математичні моделі мереж зв'язку : навч. посіб.

Харків : ХНУРЕ, 2019. 105 с.

33. Барабаш О. В., Колумбет В. П. Дослідження систем масового обслуговування на основі імітаційного моделювання з урахуванням мультиагентного підходу. *Інфокомунікаційні та комп'ютерні технології*. 2023. Т. 2, № 04. С. 115–121. URL: <https://doi.org/10.36994/2788-5518-2022-02-04-12> (дата звернення: 11.02.2024).

34. GPSS – Вікі ЦДУ. *Вікі ЦДУ*. URL: <https://wiki.cusu.edu.ua/index.php/GPSS> (дата звернення: 27.05.2024).

35. Arena Simulation Newsletter and Blog. *Arena Simulation*. URL: <https://info.arenasimulation.com/blog> (дата звернення: 27.05.2024).

36. DB-Engines Ranking. *DB-Engines*. URL: <https://db-engines.com/en/ranking> (дата звернення: 27.05.2024).

37. Explore Oracle Database Solutions for Maximum Efficiency. *Oracle | Cloud Applications and Cloud Platform*. URL: <https://www.oracle.com/database/technologies> (дата звернення: 27.05.2024).

38. What is SQL Server? - SQL Server. *Microsoft Learn: Build skills that open doors in your career*. URL: <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server> (дата звернення: 27.05.2024).

39. MySQL vs PostgreSQL | What are the differences?. *StackShare*. URL: <https://stackshare.io/stackups/mysql-vs-postgresql> (дата звернення: 27.05.2024).

40. Про затвердження Правил надання послуг з технічного обслуговування і ремонту колісних транспортних засобів. *Верховна Рада України*. URL: <https://zakon.rada.gov.ua/laws/show/z1609-14> (дата звернення: 01.03.2024).

41. Мінухін С. В., Беседовський О. М., Знахур С. В. Методи і моделі проектування на основі сучасних CASE-засобів : навч. посіб. Харків : Вид. ХНЕУ, 2008. 272 с.

42. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Pearson Education, Limited, 2018.

43. Чоломбитько Д. В. Моделювання розподілу робіт при системному проектуванні та реінжинірингу виробничих технологічних систем. *28-й Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті»* : Зб. матеріалів форуму, м. Харків, 16–18 квіт. 2024 р. Харків, 2024. С. 305–306.