

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА
ДИЗАЙНУ

ФАКУЛЬТЕТ МЕХАТРОНИКИ ТА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна робота

на тему:

**Аналіз та дослідження складних систем із застосуванням методів
моделювання**

Рівень вищої освіти другий (магістерський)
Спеціальності 122 Комп'ютерні науки
Освітня програма Комп'ютерні науки

Виконав: студент групи МгІТ-1-23
Горделадзе П.М.

Науковий керівник:
к.т.н., доц. Астістова Т.І.

Рецензент
к.т.н., доц. Мелькик Г.В.

Київ 2024

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій

Кафедра комп'ютерних наук

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки

Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

_____Наталія ЧУПРИНКА

«_____»_____2024_року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТА

Горделадзе Павла Мерабовича

1.Тема роботи: Аналіз та дослідження складних систем із застосуванням методів моделювання, науковий керівник роботи Астісова Тетяна Іванівна, к.т.н., доц., затверджені наказом закладу вищої освіти від 03.09.2024 року, № 118-уч.

2. Строк подання студентом роботи 22.11. 2024 р.

3. Вихідні дані до роботи:

Розробка кафедри комп'ютерних наук

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)

РОЗДІЛ 1. Аналіз предметної області; РОЗДІЛ 2. Алгоритми, методи та програмні засоби моделювання складних систем; РОЗДІЛ 3. Програмне забезпечення процесу моделювання .

Додатки - програмні коди системи, презентація.

5 . Дата видачі завдання: 03. 09. 2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	20.08.2024	
2	Розділ 1 Аналіз предметної області	01.09.2024	
3	Розділ 2. Алгоритми, методи та програмні засоби моделювання складних систем	10.10.2024	
4	Розділ 3. Програмне забезпечення процесу моделювання	25.10.2024	
5	Висновки	29.10.2024	
6	Оформлення кваліфікаційної роботи (чистовий варіант)	10.10.2024	
7	Подача кваліфікаційної роботи науковому керівнику для відгуку	12.11.2024	
8	Подача кваліфікаційної роботи для рецензування	18.11.2024	
9	Перевірка кваліфікаційної роботи на наявність ознак плагіату	20.11.2024	
10	Подання кваліфікаційної роботи на затвердження завідувачу кафедри	22.11.2024	

Студент

Павло ГОРДЕЛАДЗЕ

Науковий керівник роботи

Тетяна АСТІСТОВА

АНОТАЦІЯ

Горделадзе П.М. Аналіз та дослідження складних систем із застосуванням методів моделювання

Кваліфікаційна робота за спеціальністю 122 - «Комп'ютерні науки». – Київський національний університет технологій та дизайну, Київ, 2023 рік.

Темою кваліфікаційної роботи є Аналіз та дослідження складних систем із застосуванням методів моделювання

Для того щоб розв'язати поставлені задачі проведено аналіз наукових та прикладних досліджень у галузі моделювання складних систем, застосовано методи аналітичного аналізу літературних джерел та порівняльних характеристик методів і засобів моделювання складних систем; моделювання систем, теорія мереж Петрі, програмування модулів різних видів систем різними мовами програмування та пакетами програмування, експеримент і тестування – при апробації запропонованого методу і засобу.

На основі проведеного дослідження обґрунтована доцільність застосування імітаційного моделювання в прикладних пакетах при моделюванні різних видів складних систем, розглянути спеціальні моделюючі системи, реалізація графів Петрі за допомогою побудованої схеми та мови програмування Python, бібліотеки pm4py, моделювання СМО з використанням системи GPSS .

Інші методи, такі як математичні, фізичні та інші, є мало застосовуваними для моделювання складних систем. Причинами цього є складність реалізації даних методів та коштовність. Сформульовані основні вимоги до прикладних пакетів і критерії їх вибору при проведенні моделювання.

Ключові слова: Python, мереж Петрі, системи GPSS, CPN Tools, СМО, бібліотека pm4py.

ANNOTATION

Gordeladze P.M. Analysis and research of complex systems using modeling methods

Qualification work on specialty 122 - "Computer science". - Kyiv National University of Technology and Design, Kyiv, 2023.

The topic of the qualification work is Analysis and research of complex systems using modeling methods.

In order to solve the set problems, an analysis of scientific and applied research in the field of modeling complex systems was carried out, methods of analytical analysis of literary sources and comparative characteristics of methods and means of modeling complex systems were applied; modeling of systems, the theory of Petri nets, programming of modules of various types of systems in various programming languages and programming packages, experiment and testing - when approving the proposed method and means.

On the basis of the conducted research, the expediency of using simulation modeling in application packages for modeling various types of complex systems is substantiated, consider special modeling systems, the implementation of Petri graphs using a constructed scheme and the Python programming language, the pm4py library, simulation of CMO using the GPSS system.

Other methods, such as mathematical, physical and others, are rarely used for modeling complex systems. The reasons for this are the complexity of implementing these methods and their cost. The basic requirements for application packages and criteria for their selection during modeling are formulated.

Keywords: Python, Petri nets, GPSS systems, CPN Tools, SMO, pm4py library.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Постановка завдання	10
1.2.Складні системи та їх специфіка	12
1.2.1. Поняття системи.....	12
1.2.2. Ознаки та системні властивості систем.....	14
1.2.3. Принципи управління системою.....	16
1.2.4. Характерні особливості складних систем.....	18
1.3. Різновиди моделювання	20
1.4 Системний аналіз і етапи імітаційного моделювання складних.....	22
1.5 Проектування і розробка імітаційних моделей складних об'єктів.....	27
Висновки до першого розділу.....	28
РОЗДІЛ 2. АЛГОРИТМИ , МЕТОДИ ТА ПРОГРАМНІ ЗАСОБИ	
МОДЕЛЮВАННЯ СКЛАДНИХ	29
2.1. Огляд програмних засобів імітаційного моделювання.....	29
2.2. Поняття про мережі Петрі	31
2.3. Алгоритм представлення системи засобами мереж Петрі	35
2.4. Методи моделювання систем масового обслуговування.....	39
Висновки до другого розділу.....	42
РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ МОДЕЛЮВАННЯ	
3.1.Програмне забезпечення комп'ютерного експерименту.....	43
3.2. Програмні засоби моделювання мереж Петрі.....	44
3.3. Моделювання системи в CPN Tools	49
3.4. Розробка програмних модулів на мові GPSS та Python	54
Висновки до третього розділу.....	66
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69
ДОДАТКИ.....	73

ВСТУП

Актуальність теми. Для дослідження технічних, економічних, екологічних, соціальних, інформаційних систем, найбільш ефективним способом як на етапі їх проектування, так і в процесі експлуатації, є моделювання. Можливості моделювання систем далеко не вичерпані, тому постійно з'являються найновіші методи та технології моделювання.

Завдяки розвитку комп'ютерних технологій і інформатики стало простіше вирішувати складні завдання, для вирішення яких потрібні великі тимчасові і фінансові витрати. Використовуючи моделювання, можна спростити їх рішення. Найбільш зручним способом моделювання складних систем є імітаційне комп'ютерне моделювання процесів реального світу.

Створення моделі – кропіткий і творчий процес, що вимагає від дослідника не тільки глибоких теоретичних знань з різних математичних та технічних дисциплін, але й творчого підходу до розв'язання задач, вміння генерувати певні евристики, що відповідають глибинній суті досліджуваного об'єкта.

На сьогодні ученими та інженерами досліджено та розроблено багато методів математичного та концептуального моделювання, які дають змогу відображати сутності предметної області та процеси, які у ній протікають. В наш час відомі моделі при моделюванні комп'ютерних систем, побудові моделей технічних систем та виробничих процесів, систем автоматизованого управління виробничою діяльністю підприємств та галузей промисловості. З використанням моделювання розв'язана велика кількість наукових та технічних задач оптимальної організації функціонування складних систем.

Якщо взяти, як приклад розробку моделей в галузі комп'ютерних наук, то можна побачити, що враховуючи стрімкий розвиток комп'ютерних мереж, технологій IoT, використання «хмарних» сервісів, масивів великих даних (Big Data) необхідно провести додаткові дослідження щодо застосування апарату мереж Петрі при моделюванні комп'ютерних систем з метою адаптації його до

сучасних вимог та потреб в галузі комп'ютерних наук.

Наявна дійсність висуває нові завдання – моделювання багаторівневих ієрархічних систем із стохастичними, нечіткими, хаотичними та іншими властивостями, які притаманні більшості складних систем.

Об'єктом дослідження виступає процес моделювання складних систем на основі апарату мереж Петрі та інших програмних засобів.

Мета роботи. Мета дипломної роботи магістра полягає у дослідженні складних систем із застосуванням різних методів моделювання.

Завдання. Завдання роботи полягає в аналізі сучасних програмних засобів, які застосовуються для моделювання складних систем.

Методи дослідження. Для того щоб розв'язати поставлені задачі застосовано методи: аналітичного аналізу літературних джерел та порівняльних характеристик методів і засобів моделювання складних систем; теорія мереж Петрі, програмування модулів різних видів систем різними мовами програмування та пакетами програмування; експеримент і тестування – при апробації запропонованого методу і засобу.

Наукова новизна отриманих результатів. Проведено аналіз наукових та прикладних досліджень у галузі моделювання складних систем. На основі проведеного дослідження обґрунтована доцільність застосування імітаційного моделювання в прикладних пакетах при моделюванні різних видів складних систем, розглянути спеціальні моделюючі системи, реалізації графів Петрі за допомогою побудованої схеми та мови програмування Python.

Інші методи, такі як математичні, фізичні та інші, є мало застосовуваними для моделювання складних систем. Причинами цього є складність реалізації даних методів та коштовність. Сформульовані основні вимоги до прикладних пакетів і критерії їх вибору при проведенні імітаційного моделювання.

Апробація результатів.

Результати дослідження апробовано на VII міжнародній науково – практичній конференції «Мехатронні системи: інновації та інжиніринг MSIE-2024» на базі

Київського національного університету технологій та дизайну, м. Київ - 07 листопада 2024 р. у вигляді тез конференції.

1. Astistova T.I., Hordeladze P.M. Overview of existing solutions for controlling and monitoring smart home devices Матеріали VII міжнародній науково – практичної конференції «Мехатронні системи: інновації та інжиніринг MSIE-2024» КНУТД, м. Київ- 07 листопада 2024 р

2. . Астісова Т. І., Горделадзе П.М. Огляд програмних засобів імітаційного моделювання. Матеріали VII міжнародній науково – практичної конференції «Мехатронні системи: інновації та інжиніринг MSIE-2024» КНУТД, м. Київ- 07 листопада 2024 р

.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Постановка завдання

Моделювання, зокрема математичне та імітаційне, представляє собою важливий напрямок науково-технічного прогресу. З появою комп'ютерної техніки та інформаційних технологій воно широко застосовується в усіх сферах людської діяльності для створення та дослідження технічних, технологічних, соціально-економічних систем тощо. У наш час розроблені моделі багатьох виробничих процесів, систем управління виробничою діяльністю підприємств та галузей промисловості. За допомогою моделювання розв'язані наукові та технічні задачі оптимальної організації функціонування складних систем.

Дослідженню проблем моделювання у різних предметних областях присвячено праці як українських, так і закордонних учених, зокрема, А.Г. Івахненко, В.М. Томашевського, І.В. Стеценко, Ю.М. Теслі, А.А. Тимченко, Р. Шеннона, В. Кельтона, А. Лоу, В.Е. Котова та ін. Виходячи з результатів досліджень науковців, найбільш ефективним та універсальним методом при моделюванні складних систем є методи імітаційного моделювання за допомогою мережі Петрі.

Мережі Петрі використовуються при моделюванні паралельних, асинхронних, розподілених, не детермінованих та інших типів систем. Різноманітність задач, для яких застосовуються мережі Петрі, викликала розробку численних модифікацій даних мереж при моделюванні апаратних пристроїв у прикладних дослідженнях та техніці, процесів життєдіяльності у біології та медицині, інформаційних та управляючих потоків у програмних системах тощо.

Якщо провести аналіз публікацій, то використання імітаційного моделювання розглядалося протягом багатьох років безліччю дослідників – від Ф. Мартина до В. Кельтона. Запропоновані в 1962 р. в докторській дисертації Карла Петрі «Зв'язок автоматів», де були сформульовані основні поняття теорії

зв'язку асинхронних компонент обчислювальної системи, вони одержали подальший розвиток у роботах таких вчених як Т. Мурата, К. Йенсен, В. Котов, А. Слепцов тощо [1].

Сьогодні в Європі проблемами моделювання систем на базі мереж Петрі займаються провідні наукові центри в Данії (University of Aarhus) та Німеччині (University of Hamburg), які проводять теоретичні дослідження, розробляють програмні засоби моделювання та впроваджують стандарти стосовно використання мереж Петрі [1,2].

На Україні проблемами розробки та моделювання мереж Петрі займається Зайцев Д.А. [3,4]. Сучасна класифікація мереж Петрі та приклади їх застосування для розв'язання прикладних задач наведені в роботі [5].

Алгоритм процесу моделювання представлено на рисунку 1.1.

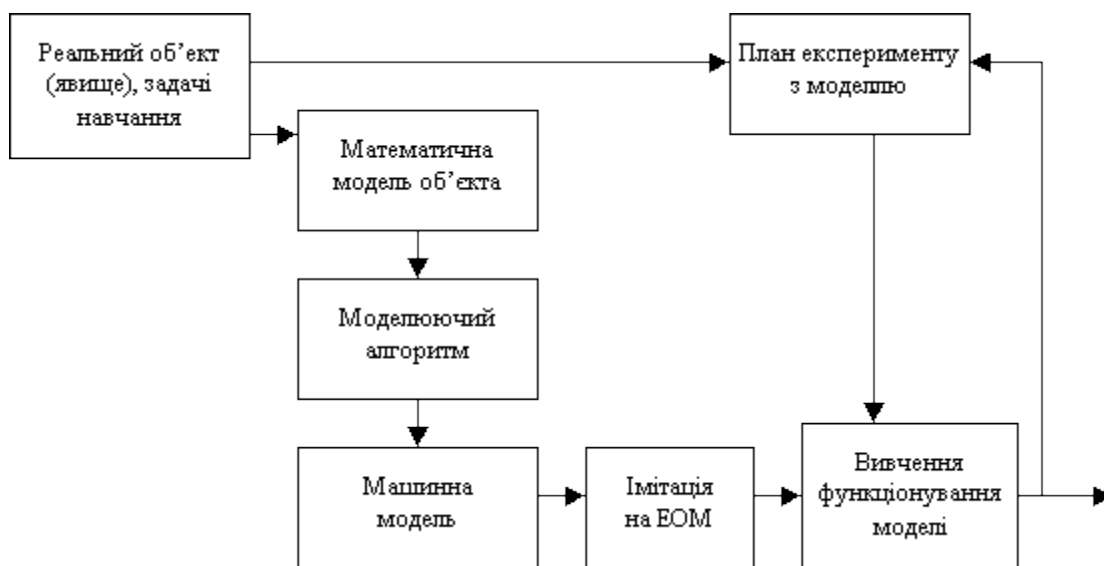


Рисунок 1.1- Алгоритм процесу моделювання

В процес моделювання можна виділити шість етапів.

1. Перший етап - визначається мета та задача моделювання.
2. Другий етап - вербальний опис систем. Це розробка концептуальній моделі системи, теоретичної бази моделювання та формальної моделі.

Концептуальна модель описує набір змінних системи, формулювання цілей та задачі дослідження. Далі обирають інструментальні засоби та теоретичну базу для побудови моделі. Формалізацію систем можна проводити засобами мереж масового обслуговування та засобами мереж Петрі.

У формальній моделі міститься також інформація, яким чином будуть знайдені вихідні змінні моделі в результаті моделювання. Наприклад, якщо в якості теоретичної бази моделювання обрані засоби мереж масового обслуговування, то формальна модель представляється зображенням мережі масового обслуговування, що складена за умовою задачі, з указуванням числових значень вхідних змінних і параметрів, а також формули розрахунку вихідних змінних моделі, що являються ціллю моделювання.

3. Третій етап - створення моделі, її реалізація за допомогою обраного програмного забезпечення та перевірка алгоритму моделювання на відповідність задуму моделювання (верифікація моделі).

4. На четвертому етапі виконується дослідження моделі. Якщо дослідження моделі відповідає цілі моделювання, то результати моделювання вважаються корисними. Після проведеного експерименту потрібно зробити статистичну обробку результатів

5. П'ятий етап – проводиться оцінка точності, стійкості та чутливості результатів моделювання.

6. Шостий етап - підводяться підсумки та висловлюються думки щодо напрямків подальшого дослідження об'єкта моделювання.

У ході процесу моделювання іноді доводиться повертатися до попередніх етапів і уточнювати або пост, щоб отримати потрібний результат.

1.2. Складні системи та їх специфіка

1.2.1 Поняття системи

Термін «система» увійшов у побут сучасної людини, яка стикається з ним дуже часто – супутникова система, геоінформаційна система, комп'ютерна система, системи освіти, система торгівлі, система супермаркетів, енергетична

система, тощо. Усі природні й штучні об'єкти та процеси, які нас оточують, є системами.

Системою називають сукупність взаємопов'язаних елементів, що об'єднані спільною метою функціонування. Елементами системи зазвичай є будь-які її складники. Наприклад, у технічній системі складниками є певні частини її конструкції (якщо ця система автомобіль, або комп'ютер, або геодезичний прилад), які пов'язані один з одним, функціонують спільно та злагоджено, завдяки чому забезпечують потрібний результат. Саме цей результат виражає мету функціонування певної системи [9].

Розташування елементів та взаємозв'язки між ними визначають поняття структура системи. Структура системи забезпечує виконання системою своїх функцій, а отже досягнення мети. У процесі функціонування система змінює один свій стан на інший (автомобіль рухається або стоїть, програмний засіб обчислює завдання або очікує команду користувача тощо). Зміна станів системи супроводжується зміною значень сукупності величин, що характеризують стани певної системи. Їх називають параметрами стану.

Системи, що протягом певного часу переходять з одного стану в інший, тобто функціонують, називають динамічними. Динамічна система може мати множину можливих станів, послідовно переходячи з одного стану в інший, проте вона у певний момент часу завжди перебуває тільки в одному із станів.

Стан системи визначають її складники та зв'язки між ними. Розрізняють три види зв'язків між елементами системи:

- механічні (обмін фізичними зусиллями між елементами);
- трофічні (обмін енергією);
- сигнальні (обмін сигналами, тобто інформацією).

Метою системи називають певний кінцевий стан, за якого параметри її вихідних характеристик отримують найкращі (потрібні або задані) значення.

Розглядання певного об'єкта або процесу називають системним підходом. У сучасній науці системний підхід займає одно з найважливіших місць. Потрібність у системному підході зазвичай виникає під час вирішення таких

задач як задачі аналізу та синтезу.

У наукових дослідженнях систем задачі синтезу та аналізу пов'язані одна з одною. Процес аналізу призначений для встановлення складників системи, їх зв'язків та функцій. Синтез дає змогу поєднати частини розділеної системи і пізнати предмет як єдине ціле. Корисним уявляється такий приклад. Людина збирається вперше їхати на автомобілі. Спочатку вона аналізує систему - визначає, де місце водія, які засоби управління водію доступні, яке призначення педаль та керма, коробки передач, які параметри режимів руху індиктує панель приладів. Далі, синтезуючи функції складників, людина керує рухом системи.

Треба зауважити, що доцільність застосування системного підходу до вивчення об'єктів та процесів полягає у тім, що абстрактні системи, тобто системи, різні за фізичною природою та структурою (навіть такі як технічні та біологічні), часто мають ідентичні властивості та поведінку, а отже для їх моделювання можна використовувати той самий математичний апарат.

Системний підхід передбачає застосування методів системного аналізу, який є науковим методом пізнання. Він являє собою послідовність дій щодо встановлення структурних зв'язків між змінними або елементами досліджуваної системи, які спираються на комплекс загальнонаукових, експериментальних, природничих, статистичних та математичних методів. У зарубіжній науковій літературі поняття системного підходу та системного аналізу – синоніми, тобто однакові за змістом.

1.2.2 Ознаки та системні властивості систем

Якщо апіорна інформація про об'єкт дослідження мінімальна (майже відсутні відомості про внутрішню будову), систему подають як непрозорий «чорний ящик», виділений з навколишнього середовища (рис. 1.1). Ця максимально проста модель системи відображає дві важливі її властивості: цілісність і відокремленість від навколишнього середовища. Одночасно «чорний ящик» пов'язаний з навколишнім середовищем певними зв'язками.

Зв'язки, що впливають на систему, називають входами системи. Оскільки система реалізує певну мету, вона чинить вплив на стан навколишнього середовища. Ці зв'язки називають виходами системи.

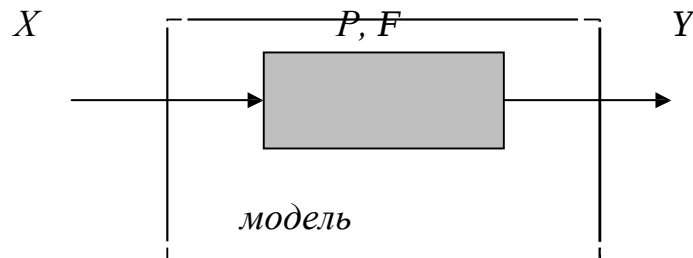


Рисунок 1.1- Система «чорний ящик»

Система «чорний ящик» в загальному випадку модель має структуру, зображену на рисунку 1.1. Тут X – множина вхідних змінних системи, Y – множина вихідних змінних системи, P – множина параметрів, F – функція, функціонал, алго-ритм або формальне представлення залежності змінних Y від змінних X .

Будь яка система характеризується такими властивостями (ознаками): – границі системи, адже зрозуміло, що нескінченну систему неможливо уявити та аналізувати, тобто у такій системі немає змісту;

- відкритість, яка проявляється у зв'язках системи з навколишнім середовищем;

- структурованість системи, що проявляється у наявності елементів та зв'язків між ними. Зауважимо, що структура системи залежить від завдання дослідження;

- динамічність системи визначається здатністю змінювати з часом свій стан

- ієрархічність системи – це така будова системи, за якою кожна її частина також є системою, що складається з своїх елементів, глибина поділу яких на складники визначається завданням дослідження.

Зауважимо, що будь-який елемент системи так само можна розглядати як систему; – емерджентність (з англійської «emergence» – виникнення, поява нового) системи полягає у наявності в неї таких властивостей, яких нема в її елементів, тобто поява нових властивостей; – підпорядкованість функціонування всіх елементів системи спільній меті.

Ця ознака формується в силу об'єктивної необхідності. Кожна з розглянутих ознак для будь-якої системи є обов'язковою. Якщо в досліджуваної системи не дістає хоча б однієї з них, то це означає, що ми помилково визначили систему, зокрема, її границі.

Тобто якщо в системи немає границь, або система ізольована від навколишнього середовища, або деякі з її складників не мають зв'язків з іншими, або якщо система немає загальної мети, то і змісту немає у вивченні такої сукупності елементів.

1.2.3. Принципи управління системою

Усі процеси та явища поза межами системи є зовнішнім середовищем. Отже система проявляє свої властивості у процесі взаємодії із зовнішнім середовищем. У сукупності зовнішніх впливів на систему, що наведена на рисунку 1.1, виділяють корисний сигнал – сигнал управління $x(t)$ та випадкові впливи - сигнали $f(t)$, які є завадами.

Сигнал на виході системи $y(t)$ реалізує мету системи. Ефективність функціонування системи визначається ступенем досягнення її мети. Очевидно, що випадкові впливи, тобто завади, спотворюють сигнал на виході системи $y(t)$, що зменшує її ефективність. Для підвищення ефективності на вхід системи подають сигнал управління $x(t)$. Управлінням називають цілеспрямований вплив на систему з метою отримання найкращого ефекту.

Розрізняють два основних принципи управління – управління за збуренням та управління за відхиленням [6,9].

Перший принцип – управління за збуренням – передбачає управління розімкненою системою, тобто системою, у структурі якої немає зворотного зв'язку. Цей принцип полягає у тім, що сигнал управління змінюється так, що

він компенсує вплив завади. Перевага цього принципу управління полягає у тому, що вихідний сигнал може зовсім не відхилятися від потрібного значення, тобто ефективність функціонування системи може становити 100 %, але це тільки теоретично.

На практиці у більшості випадків вичерпні дані про дію випадкових впливів $f(t)$, тобто момент впливу, тривалість, величина та ін. не відомі. Перевага принципу управління за збуренням полягає у можливості заздалегідь врахувати усі випадкові впливи та їх компенсувати шляхом зміни, наприклад, сигналу управління. Але у цьому принципі є недолік, адже не про всі випадкові впливи є інформація, отже більшість з них не вдається врахувати [22].

Другий принцип – управління за відхиленням – передбачає, що система замкнена зворотним зв'язком. Зворотний зв'язок передає дані про функціонування системи з її виходу на вхід, де сигнал зворотного зв'язку порівнюється з вхідним, який відповідно корегується, та на вхід надходить алгебраїчна сума сигналів.

У більшості автоматичних систем застосовують від'ємний зворотний зв'язок, тобто сигнал зворотного зв'язку $y_1(t)$ віднімається від вхідного сигналу $x(t)$.

Тоді можливі три стани: $|y_1(t)| = |x(t)|$,

$$|y_1(t)| > |x(t)|$$

$$|y_1(t)| < |x(t)|.$$

У першому випадку сигнал управління не змінюється ($|x(t)| - |y_1(t)| = 0$). У другому випадку сигнал управління зменшується, оскільки різниця стає від'ємною, а у третьому випадку – збільшується, оскільки різниця стає додатною. Перевага принципу управління за відхиленням – у можливості врахувати усі впливи та завади, а недолік – принципова наявність похибки на виході системи, оскільки зворотний зв'язок надає інформацію про похибку тільки після її виникнення. Системи класифікують за низкою ознак: за природою утворення, за ступенем зв'язку із зовнішнім середовищем, за

залежністю від часу, за керованістю, за місцем в ієрархії систем, за ступенем складності та ін.

1.2.4 Характерні особливості складних систем

Починаючи з середини XX століття, як результат технічного прогресу з'явилися людино-машинні обчислювальні системи, розроблюються космічні проекти та інші складні технічні задачі. Поряд з цим у соціальному управлінні створюються сучасні виробничі комплекси, що тягне за собою розвиток міст та комплексних заходів з охорони природи. Отже у науці відбувся перехід до таких наукових задач, як проблеми організації і функціонування складних об'єктів і людині доводиться звертатись до систем, границі і склад яких зовсім не очевидні і у кожному разі вимагають спеціальних досліджень.

Для вирішення подібних питань і почали розроблятися складні системи. У сучасній науці поняття складна система (з англ. «complexity» – «складність») дуже широко використовується. До категорії складних систем, перш за все, належать природні системи, зокрема живі організми – від людини до самих маленьких бактерій, які проявляють такі важливі властивості, як адаптація до навколишнього середовища, величезна різноманітність та надзвичайна складність внутрішньої структури. Складність цих систем зумовлена складністю механізмів взаємодії елементів. До складних систем належать, зокрема, так звані організаційні системи, під якими розуміють деякі організації, наприклад, виробничі підприємства.

Розглянемо коротко основні характерні риси складних систем.

Головною особливістю складних систем є те, що окрім відомих і чітко визначених факторів, що впливають на стан і динаміку системи, визнається вплив випадкових (стохастичних) факторів. Наукові та повсякденні факти свідчать, що не все, що відбувається навколо нас, можна точно пояснити. А отже доводиться користуватись категорією випадковості та через неї сприймати досліджувані процеси, визнаючи таким чином можливість зміни і переходу складної системи в абсолютно непередбачувані стани, тобто поведінка складної системи може опинитись непередбачуваною.

Інша особливість складної системи полягає у її ієрархічності. Складні системи можуть мати кілька рівнів ієрархії, а такі системи як біосфера або світова економіка налічують десятки рівнів. Це спричинено тим, що складники такої системи, які ми розглядаємо як її елементи, можуть у свою чергу бути також складними системами, тобто мати власну поведінку, стан і структуру, мати зворотні зв'язки, причому як від'ємні, так і додатні, а також спостережувані й контрольовані входи та виходи

. Великим і складним системам притаманні також властивості цілісності та емерджентності. Цілісність системи означає, що всієї її частини слугують спільній цілі та сприяють формуванню найкращих (оптимальних) результатів у смислі прийнятого критерію ефективності. Емерджентність (поява нового) означає, що великі й складні системи мають такі властивості, які не властиві жодному з її елементів.

Емерджентність або системні властивості кардинально відрізняють системи від не систем. Інакше кажучи, об'єднання підсистем з різною природою (наприклад, економічної, технічної, соціальної) і структурою в складні системи відбувається при їхньому взаємному впливі одна на одну, що й створює нову системну властивість, якої не має жодна з підсистем складної системи.

Емерджентність відображає неможливість зведення властивостей системи до суми властивостей її компонентів. Для розуміння поняття складної системи доцільно порівняти особливості складної та простої системи. Проста система зазвичай характеризується тим, що має однозначний зв'язок між зовнішніми впливами та вихідним параметром, тобто реакцією системи. Малі зміни в характері зовнішнього впливу викликають малі зміни в реакції системи, та навпаки – істотні за величиною зміни зовнішнього впливу викликають великі зміни в реакції системи. Як приклади поведінки таких систем можна навести рух на автомобілі, роботу з комп'ютером, або геодезичним приладом. Отже для будь-якого зовнішнього впливу поведінку простої системи можна передбачити.

Складні системи мають велику кількість елементів і характеризуються

розгалуженою структурою, виконують складніші функції. Для складної системи характерні такі властивості як наявність певної кількості зворотних зв'язків, у складній системі малі зміни зовнішніх впливів викликають суттєві, часто якісні, зміни в реакції системи, непередбачуваність та ін.

1.3. Різновиди моделювання

Моделювання, це процес опису чи формалізації об'єкта реального світу з метою створення аналога-моделі, адекватного об'єкту. Моделювання можна розділити аналітичне та імітаційне.

Під аналітичним моделюванням розуміється процес формалізації реального об'єкту і знаходження його рішення в аналітичних функціях. [

Модель, сформульована на мові математики, фізики, хімії або іншої науки за допомогою системи спеціалізованих символів з точними правилами сполучуваності, називається аналітичною моделлю, найчастіше вони представляються у вигляді формул, нерівностей, лінійних і нелінійних рівнянь, у тому числі диференціальних рівнянь і їх комбінацій

Під імітаційним моделюванням розуміється "розробка моделі системи у вигляді програми для комп'ютера і проведення експериментів з програмою, замість проведення експериментів з реальною системою або об'єктом".

Імітаційне моделювання (рис.1.2) застосовується, коли неможливе побудувати аналітичну модель системи, що враховує причинні зв'язки, наслідок, не лінійності, стохастичні змінні, коли необхідно імітувати поведінку системи в часі, розглядаючи різні можливі сценарії її розвитку при зміні зовнішніх

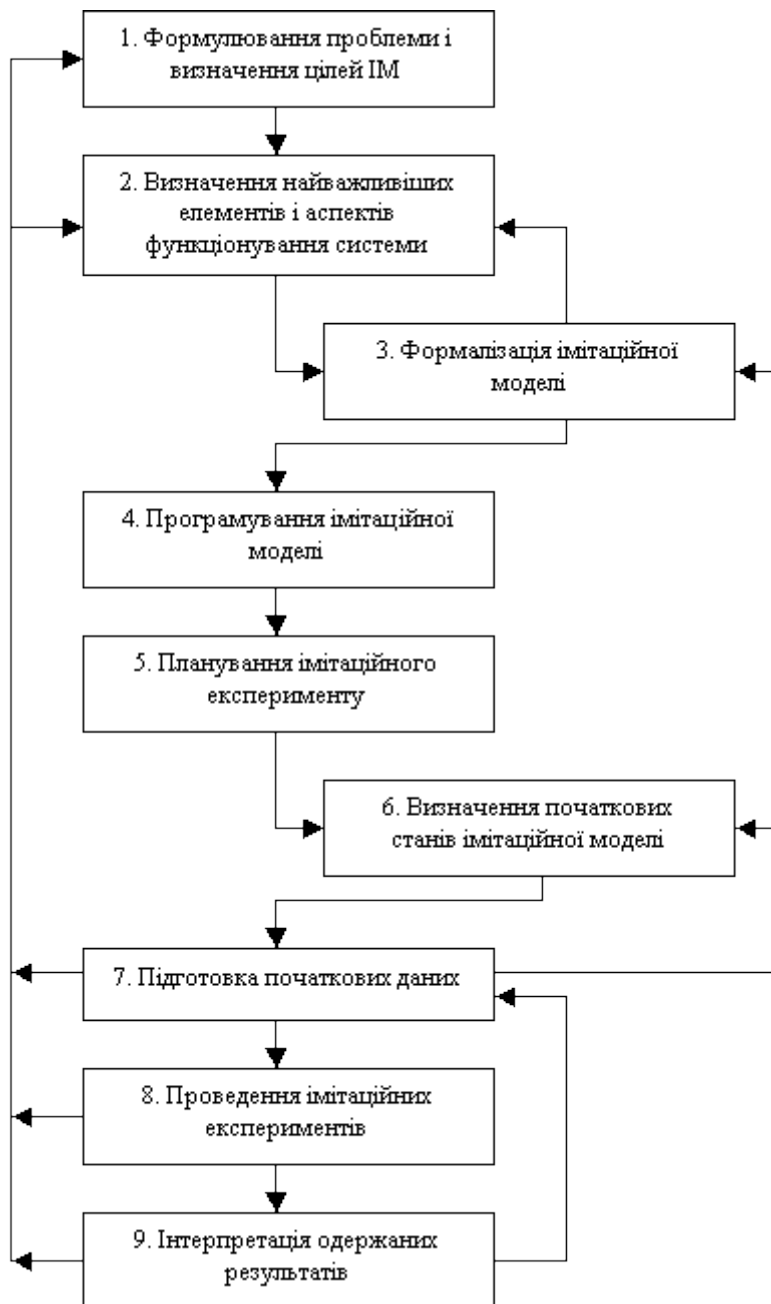


Рисунок 1.2- Схема імітаційного методу моделювання

Модель є спрощеним відображенням реальності – менш детальним, менш складним відтворенням реально існуючого об'єкту, системи або феномену, процесу.

Імітаційне моделювання умовно може бути представлено особистими різновидами або напрямками, що відповідно мають свої методології.

1. *Статистичне (чисельне) моделювання* є різновидом імітаційного моделювання. Цей підхід до дослідження реального процесу був названий

методом статистичних випробувань (методом Монте-Карло).

Реалізація рішення імовірнісної моделі реального об'єкту здійснюється на комп'ютері..

2. *Дискретно-подієве моделювання.* Цей підхід використовується для опису функціонування системи (процесу) з одного стану в інший дискретним чином у вигляді події.

Будуються графічні діаграми причинних зв'язків і глобальних впливів одних параметрів на інші в часі, а потім створена на основі цих діаграм модель імітується на комп'ютері.

3. *Агентне моделювання (agent - based model (ABM))* - різновид імітаційного моделювання, сучасний метод, що дозволяє досліджувати роботу децентралізованих агентів і те, як така поведінка визначає поведінку усієї системи в цілому

Глобальна поведінка виникає як результат діяльності безлічі агентів (моделювання "від низу до верху").

4. *Методологія когнітивного моделювання* призначена для аналізу і ухвалення рішень в погано певних ситуаціях. Під когнітивними технологіями розуміється широкий спектр технологій раціоналізації і формалізації інтелектуальних систем створення і функціонування знань, експертизи, комунікації і ухвалення рішення.

5. *Ситуаційне моделювання.* Ситуаційне моделювання (situational simulation), ситуаційне управління (management situations) - напрям досліджень і ухвалення рішень.

Ситуаційне управління - це метод управління складними технічними і організаційними системами, ґрунтованих на ідеях теорії штучного інтелекту.

1.4. Системний аналіз і етапи імітаційного моделювання складних систем

У кожному циклу розробки комп'ютерних моделей складних об'єктів можна виділити наступні етапи [22,25].

1. *Формулювання проблеми.*

Результатом цього етапу має бути документований змістовний опис об'єкту моделювання, розроблені алгоритми системного підходу до вирішення проблеми. Вони представлені у вигляді ряду етапів.

Експерименти по моделюванню проводяться з різноманітними цілями:

- прогноз - критика поведінки системи при деякому передбачуваному поєднанні робочих умов; порівняння альтернатив;
- порівняння систем, що змагаються, розрахованих на виконання конкретної функції, або ж на порівняння 1 -го і більше пропонованих робочих принципів або методик;
- виявлення багатофункціональних співвідношень - визначення залежності між двома або більше діючими чинниками, з одного боку, і відгуком цієї системи, з іншого боку;
- аналіз чутливості - виявлення з трохи більшого числа працюючих чинників тих, які більшою мірою впливають на усю поведінку системи;
- оцінка - визначення, як буквально система пропонованої структури стане підходити деяким конкретним аспектам;
- оптимізація - конкретне визначення поєднання працюючих величин і їх причин, що забезпечують найкращий відгук усієї системи в цілому;
- демонстрація - показ можливостей моделі і імітаційних досліджень системи.

2. Визначення меж.

Логіко-математичний опис модельованої системи відповідно до формулювання проблеми. Визначення меж системи і зовнішнього середовища, тобто виділення системи з довкілля. Визначення вхідних параметрів і вихідних характеристик системи.

Компонентами системи це складові частини, що утворюють систему.

Ці співвідношення за природою є або детермінованими або стохастичними.

Обмеження є встановлюваними межами зміни значень змінних або обмежуючі умови їх змін

Цільова функція (функція критерію) є точним відображенням цілей або завдань системи і необхідних правил оцінки їх виконання

3. Формулювання і розробка моделі.

Включає розробку концептуальної моделі і формалізацію побудованої концептуальної моделі.

На цій стадії роботи результатом діяльності розробника комп'ютерної моделі є створення повної концептуальної (абстрактної) моделі.

Головним змістом цього кроку є перехід від справжньої системи до логічної схеми її функціонування, формулювання загального плану моделі. У цьому кроці наводиться алгоритмізація функціонування її складової і відображення об'єкту в термінах математичних понять.

Спрощення, наближення – є основним прийомом будь-якого моделювання. Під спрощенням (наближенням) розуміється зневага несуттєвими деталями, або прийняття припущень о найбільш простих співвідношеннях (приміром, припущення о лінійної залежності між змінними).

Залежно від складності об'єкту і мети моделювання вибирається один з підходів аналітичного або імітаційного моделювання. У рамках вибраного підходу проводиться розробка математичного опису об'єкту моделювання.

Результатом цього етапу є розробка технічного проекту комп'ютерної установки для моделювання.

Процес формалізації складної системи включає наступні види робіт :

- вибір способу формалізації;
- складання формального опису системи.

В процесі побудови моделі можна виділити 3 рівні її представлення :

- неформалізований (етап 2)
- концептуальна модель;
- формалізований (етап 3)
- формальна модель;
- програмний (етап 4)
- імітаційна модель.

Кожен рівень відрізняється від попереднього мірою деталізації модельованої системи і способами опису її структури і процесу функціонування. При цьому рівень абстрагування зростає.

Концептуальна модель – це систематизований змістовне опис модельованої системи (чи проблемній ситуації) на неформальній мові.

Імітаційна модель – перетворення формалізованого опису в програму - імітатор, побудовану відповідно до деякої методики в середовищах програмування або моделювання.

Вибірання інструментального засобу для побудови комп'ютерної моделі є основним моментом в імітаційному дослідженні складної системи.

4. Підготовка даних.

Включає ідентифікацію, специфікацію і збір даних.

- Ідентифікація – статистичний аналіз моделі, статистичне оцінювання невідомих параметрів.
- Специфікація – знаходження кінцевої мети моделювання; визначення набору екзогенних і ендогенних змінних; визначення складу системи рівнянь, їх структур; формулювання початкових передумов, обмежень..
- Підготовка даних - це збір і аналіз початкових даних для моделювання.

Якщо трасування і програмування імітаційної моделі можна виконувати на гіпотетичних даних, то майбутнє експериментальне дослідження треба виконувати на реальному потоці даних.

Основні способи отримання початкових даних

- з експериментальних даних (фізичний експеримент);
- з документації на систему (фінансова і технічна документація для промислових систем, дані звітів, статистичні збірки та ін.);
- з літературних джерел по даній системі.

Друге питання пов'язане з проблемами ідентифікації вхідних даних для стохастичних систем. Імітаційне моделювання є дуже ефективним апаратом дослідження стохастичних систем, тобто таких систем, динаміка яких залежить від випадкових чинників : вхідні (або вихідні) змінні стохастичної моделі,

випадкові величини, вектори, функції, випадкові процеси.

Для підбору теоретичних розподілів випадкових величин використовують відомі прийоми математичної статистики, які спираються на визначенні параметрів емпіричних розподілів і перевірці статистичних гіпотез.

5. Трансляція моделі.

Трансляція моделі – це переклад моделі із спеціальних імітаційних мов або мови математики на мову програмування, на якому буде реалізована застосовна програма, що відповідає комп'ютерній моделі. Алгоритмізація і програмна реалізація, тобто будується програмний комплекс моделювання об'єкту дослідження. Проводиться відладка комп'ютерної моделі.

6. Оцінка адекватності (верифікація і валідація)

Верифікація – це установка правильності розробленої програми, формальний, або практичний доказ її правильної працездатності на ЕОМ, тестування комп'ютерної моделі на адекватність об'єкту моделювання. Валідація – це оцінка необхідної точності і адекватності імітаційної моделі.

Після реалізації імітаційної моделі на ЕОМ, необхідно проводити випробування для оцінки достовірності моделі.

7-8. Стратегічне і тактичне планування

На цьому етапі проводиться стратегічне і тактичне планування машинного експерименту. Результатом є складений план експерименту і проведений обчислювальний експеримент ("прогони" комп'ютерної моделі з різними початковими даними).

9. Постановка експериментів. На цьому етапі передбачається прогін програми імітаційної моделі на ЕОМ для отримання вихідних даних або результатів, що дозволяють оцінити адекватність побудованої моделі. Тут також необхідно визначити умови, в яких здійснюватиметься тестування, перевірка працездатності і можливості функціонування; параметри, на які потрібно звернути увагу при тестуванні моделі.

На останніх стадіях імітаційного моделювання необхідно вести стратегічне і тактичне планування імітаційного експерименту. На етапі застосовуються

способи планування обчислювального експерименту, статистичний, регресійний і дисперсійний аналіз, методи оптимізації. Організація і проведення досвіду вимагає коректного використання можливих аналітичних прийомів. Згідно з отриманими результатами проведене обстеження повинне дозволити зробити висновки, достатні для ухвалення рішень по позначених на ранніх стадіях проблемах і завданнях..

10. *Реалізація і документування* На основі побудованої імітаційної моделі можна дати рекомендації про ухвалення того або іншого управлінського рішення і документально відобразити процес функціонування моделі і отримані результати. Аналіз основних проблем, що виникають при створенні моделі складної системи.

1.5. Проектування та розробка імітаційних моделей складних об'єктів

Основні стадії і етапи в стадіях для розробки комп'ютерних моделей складних систем як різновиду програмного забезпечення або інформаційної системи може бути коротко представлено таким чином:

1. Передпроектна стадія – стадія формування вимог до автоматизованої системи:

- етап розробки концепції автоматизованої системи;
- етап розробки і затвердження технічного завдання.

2. Стадія проектування та розробки програмного забезпечення:

- етап розробки ескізного та технічного проекту автоматизованої системи;
- етап проектування програмного забезпечення;
- етап проектування інтерфейсу;
- етап реалізації програмного забезпечення (створення програмного коду);
- етап створення та оформлення документації.

3. Стадія впровадження.

4. Період супроводу або призначений для користувача період.

Метою систем аналізу і проектування є визначення системних вимог і властивостей, створення проекту і архітектури інформаційної системи, а також детальна «калька» проекту, що включає алгоритми і визначення структур даних

Сучасний підхід до проектування інформаційних систем передбачає створення моделі вихідної інформаційної системи, що описує всі необхідні аспекти її функціонування. Вживання моделей дозволяє скоротити терміни проектування, поліпшити якість проекту за рахунок усунення великого числа помилок у вирішенні стратегічних питань вже на ранніх стадіях роботи. При створенні такої моделі зазвичай застосовується функціональна методологія. Вона передбачає розгляд системи у вигляді набору функцій, які перетворюють вхідний потік інформації в вихідний.

Висновки до першого розділу.

У розділі проведено аналіз наукових та прикладних досліджень у галузі моделювання складних систем, застосовано методи аналітичного аналізу літературних джерел та порівняльних характеристик методів і засобів моделювання складних систем. Було розглянуто загальні принципи систем та системного аналізу, основні поняття моделі, основні принципи управління системою та характерні особливості складних систем.

Аналітичне моделювання охоплює моделювання систем засобами мереж масового обслуговування, моделювання методами самоорганізації моделей, а також дослідження властивостей систем, функціонування яких описано засобами мереж Петрі.

РОЗДІЛ 2. АЛГОРИТМИ, МЕТОДИ ТА ПРОГРАМНІ ЗАСОБИ МОДЕЛЮВАННЯ СКЛАДНИХ СИСТЕМ

2.1 Огляд програмних засобів імітаційного моделювання

Імітаційне моделювання (ІМ) є одним із перспективніших напрямів планування та управління виробництвом яке дає змогу отримати якісні та кількісні оцінки можливих наслідків прийнятих рішень. Воно відтворює поведінку досліджуваної системи за результатами аналізу найбільш істотних зв'язків між її елементами.

Імітаційна модель – це програма або комп'ютерна установка, яка описує структуру і поведінку реальної системи в часі за допомогою інформаційних технологій. Ці моделі можуть бути створені за допомогою програмного забезпечення, систем комп'ютерного моделювання.

Програмування комп'ютерної моделі імітаційного моделювання можна розділити на чотири групи [9, 22]:

1. За допомогою універсальних мов (наприклад C++, Delphi, Pascal). Динаміку системи описують рівняннями, які кодують у програму, потім проводять розрахунок рівнянь та встановлюють зв'язок вихідних величин із вхідними.
2. Із застосуванням спеціалізованих мов моделювання (наприклад, GPSS, AnyLogic), написаних універсальними мовами. Динаміка системи відображається взаємодією елементів моделі у часі та просторі. Спеціалізовані мови імітаційного моделювання потребують спеціальної підготовки користувача, який має написати програму в термінах мови для конкретного об'єкта моделювання.
3. За допомогою спеціалізованих комп'ютерних середовищ (наприклад, Arena, AnyLogic, GPSS World, VisSim). Замість написання програми користувачі складають модель із бібліотечних графічних модулів та/або заповнюють спеціальні форми. Імітаційне середовище забезпечує можливість візуалізації процесу імітації, дозволяє проводити сценарний аналіз та пошук оптимальних

рішень.

4. Включення засобів імітаційного моделювання у стандартні математичні комп'ютерні системи (наприклад, пакет Simulink системи Matlab, Mathcad, Maple). Це програмні середовища, призначені для виконання різноманітних математичних і технічних розрахунків, що надають користувачеві інструменти для роботи з формулами, числами, графіками, текстом, включають засоби для управління змінними, введенням і виведенням даних, а також забезпечені графічним інтерфейсом

Існує цілий ряд популярних систем імітаційного моделювання різного класу – від простих програм, призначених для установки на персональному комп'ютері, до потужних систем, що включають бібліотеки більшості наявних на ринку комунікаційних пристроїв і дозволяють в значній мірі автоматизувати дослідження мережі, яка вивчається.

Для імітаційного моделювання використовується величезна кількість програмних продуктів, таких як: Aimsun (Розробник: компанія TSS – Transport Simulation Systems, S.L, Іспанія), AnyLogic (Розробник: The AnyLogic Company (XJ Technologies), Arena (Розробник: Rockwell Automation Inc., Wexford, PA, США), AutoMod (Розробник: Brooks Automation, США), AweSim (Розробник: Symix Systems Inc., США).

Але нами більш детально буде розглянута одна з найбільш поширених і доступніших програм, що використовуються при дискретно-подієвому моделюванні, – GPSS. GPSS (General Purpose Simulation System) – система моделювання складних об'єктів загального призначення.

Спочатку розроблювалась і підтримувалась компанією IBM. Наразі існують версії різних розробників, найсучасніша з яких – GPSS World – версія GPSS для персональних ЕОМ та ОС Windows, що розроблена компанією Minuteman Software.

GPSS World є об'єктно-орієнтованою мовою[16]. Можливості візуального представлення інформації мови GPSS World дозволяють спостерігати та фіксувати внутрішні механізми функціонування моделей. Інтерактивність мови

GPSS World дозволяє одночасно досліджувати та управляти процесами моделювання.

Система GPSS добре підходить для моделювання процесів масового обслуговування. Однак ця система має кілька недоліків: перший із них полягає в тому, що GPSS дозволяє моделювати лише один пристрій. Якщо в моделі є два або більше паралельно працюючих об'єктів, то таку модель необхідно моделювати за допомогою кількох паралельно працюючих пристроїв. Іншим недоліком є відсутність графічних даних, що позбавляє можливості візуального представлення моделі в реальності, а також ускладнює процес обробки інформації. Крім того, в цій програмі дуже складно представити процеси обробки інформації у вигляді алгоритму.

2.2. Поняття про мережі Петрі

Мережа Петрі – це графічний і математичний засіб моделювання систем і процесів. Як правило, мережами Петрі моделюють паралельні(синхронні та асинхронні) системи і процеси. Спочатку запропоновані в докторській дисертації Карла Петрі в 1962 році вони одержали подальший розвиток у роботах таких вчених як Тадао Мурата, Курт Йенсен, Віталій Котов, Анатолій Слепцов. В останні часи проводиться щорічна конференція «Застосування і теорія мереж Петрі», видається в Боні інформаційний бюлетень «Новини мереж Петрі» (Petri Net Newsletter), відомо кількасот моделюючих систем для різних програмно-апаратних платформ, існують реалізації процесорів мереж Петрі. Галузі застосування мереж Петрі включають дослідження телекомунікаційних мереж, мережних протоколів, обчислювальних систем і обчислювальних процесів, виробничих і організаційних систем.

Мережа Петрі – це засіб моделювання системі процесів, яка графічно представляється у вигляді маркованого дводольного орієнтованого мультиграфу, який має дві групи вершин: позиції та переходи. Позиції визначають стан мережі, переходи визначають дії. Кожній позиції можна поставити у відповідність кількість маркерів (фішок), які відображають виконання / невиконання визначених умов. Орієнтовані ребра графу задають

зв'язки між позиціями та переходами. Процес функціонування мережі Петрі полягає в послідовному «виконанні» переходів, та відповідному перерахунку кількості «фішок» у позиціях. Дуги можуть бути кратними, коли два вузли з'єднані більше ніж однією дугою однакового напрямку. Альтернативно, для відображення кратності дуг може використовуватися функція «ваги» дуг.

Як правило, мережами Петрі моделюють паралельні (синхронні та асинхронні) системи і процеси. В останні часи проводиться щорічна конференція «Застосування і теорія мереж Петрі», видається в Боні інформаційний бюлетень «Новини мереж Петрі» (Petri Net Newsletter), відомо декілька сотень моделюючих систем для різних програмно-апаратних платформ, існують реалізації процесорів мереж Петрі. Галузі застосування мереж Петрі включають дослідження телекомунікаційних мереж, мережних протоколів, обчислювальних систем і обчислювальних процесів, виробничих і організаційних систем. [2].

Класифікація мереж Петрі представлена на рис. 1

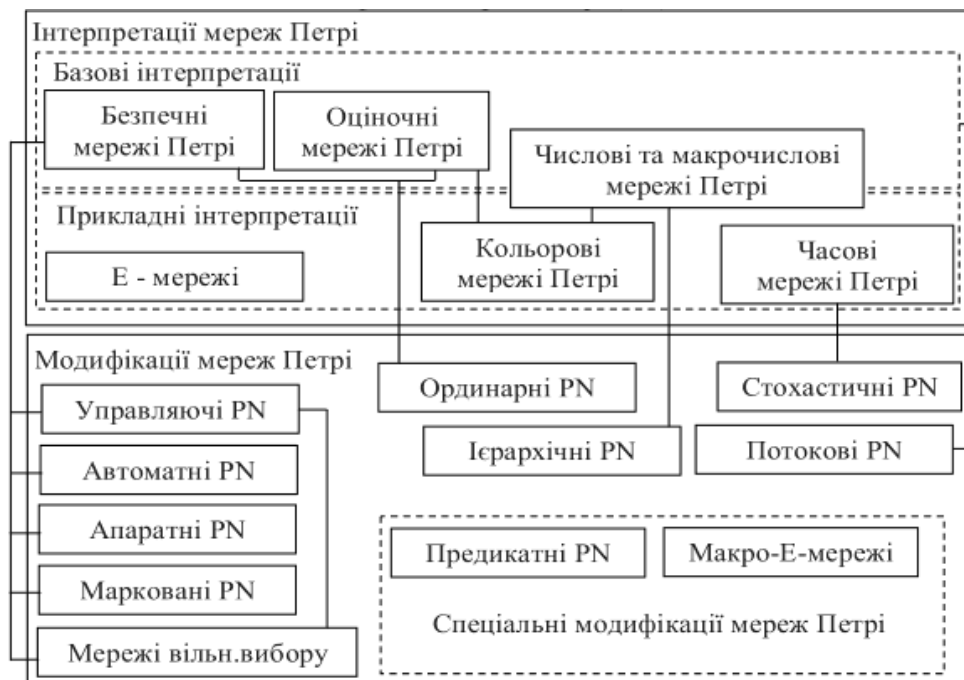


Рис.2. 1. Класифікація мереж Петрі

Створити модель системи, змоделювати модель за допомогою математичного апарату Мереж Петрі та дослідити її роботу. Маніпулюючи моделлю системи,

можна отримати нові знання про неї. Після моделювання, виконаємо верифікацію моделі, за допомогою CPN Tools (спеціальна моделююча система, яка використовує мову мереж Петрі для опису моделей).

Та розробити графічну програму, яка візуально продемонструє роботу розробленої моделі.

Мережа Петрі є орієнтованим дводольним графом, який має чотири базових елементи: *вузли*, або *місця* (places), *переходи* (transitions), *дуги* (arcs) і *маркери* (tokens). Вузли позначаються кружками і визначають стан, в якому може знаходитись мережа або її частина. Переходи - це активні елементи мережі, які позначають дії, виконувани під час спрацювання переходів. Для того щоб перехід міг спрацювати, необхідне виконання певних умов, які визначаються наявністю маркерів у вузлах мережі, з'єднаних з переходом. Якщо умови настання подій виконано, то вважають, що перехід збуджений. Переходи позначаються короткими вертикальними або горизонтальними лініями.

Вузли та переходи з'єднуються орієнтованими ребрами (дугами). Вузли, з яких виходять дуги до певного переходу, називаються вхідними вузлами переходу, а вузли, до яких ведуть дуги з певного переходу, - вихідними. Два вузли або два переходи з'єднуватись дугами не можуть. Кожний перехід може бути з'єднаним з вузлом тільки однією дугою (вхідною або вихідною).

Застосування мереж Петрі

В даний час мережі Петрі застосовуються в основному, у моделюванні. У багатьох областях досліджень явище вивчається не безпосередньо, а опосередковано, через модель. Маніпулюючи моделлю системи, можна отримати нові знання про неї, уникаючи небезпеки, дорожнечу або незручності аналізу самої реальної системи. Зазвичай моделі мають математичну основу.

Моделювання залізничних мереж

Оскільки МП це компактна, формальна і графічна мова з вбудованою паралельністю, вона дуже зручна для моделювання ж/д систем. Однак одержувані моделі настільки об'ємні, що необхідно абстрагуватися від деталей.

Рішенням є введення ієрархії МП моделей.

Моделювання комп'ютерних атак

Створення моделі комп'ютерної атаки, яку можна застосовувати для синтезу мережевого трафіку атакуючого впливу, повинна відображати як статичні, так і динамічні складові мережевого трафіку.

Але застосування мереж Петрі також зустрічаються в різноманітних галузях, та тисячі задач для моделювання та діагностики, наприклад:

- Використання мереж для оцінки часу евакуації людей в будівлях і спорудах.
- Використання при проведенні аналізу живучості бортових інформаційно-керуючих систем.
- Використання мереж Петрі для моделювання процесів транспортно-експедиційного обслуговування.
- Застосування мереж для опису паралельних процесів.
- Застосування мереж Петрі для діагностування проблем синхронізації в обчислювальних системах із загальною пам'яттю.

Створення моделі.

Змоделюємо простий процес, який ми часто виконуємо у повсякденному житті, це процес користування банкоматом. Коли спрацює перехід «клієнт підійшов до банкомату», у позиції «у черзі клієнтів є клієнт» з'явиться маркер. Наявність маркерів у позиціях «у черзі клієнтів є клієнт» та «банкомат вільний» означає, що виконані умови для події «клієнт розпочав виконувати операції». При запуску переходу «клієнт розпочав виконувати операції» маркери з позицій «у черзі клієнтів є клієнт» та «банкомат вільний» видаляються, а в позиції «клієнт обслуговується» маркер з'являється. Наявність маркеру в позиції «клієнт обслуговується» означає, що виконані умови для події «клієнт закінчив виконувати операції». При запуску цього переходу маркер з позиції «клієнт обслуговується» видаляється, а в позиції «банкомат вільний» та «є обслугований клієнт» маркери додаються. [8]

Змоделюємо роботу торгового магазину, при покупці товару.

Змодельюємо систему яка буде через деякий час t здійснювати замовлення. Представимо графічно цей процес купування товару мережею Петрі. Вона має вид , як на рисунку 2.1.

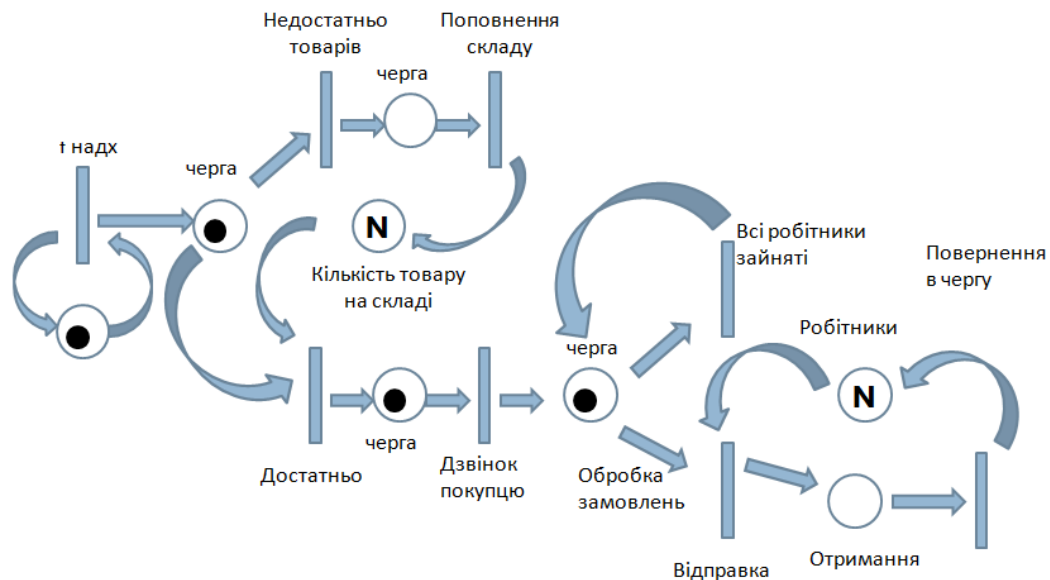


Рис. 2.2. Мережа Петрі моделі покупки товару

Наша схема має часткові тупики. Перший тупик який ми можемо зустріти під час обробки замовлень – це недостатня кількість товару, наше замовлення потрапляє в чергу на закупку певного товару. Далі замовивши товар, кількість товару на складі збільшується для обробки товару, і переходить в достатню кількість товару. Наступний частковий тупик який може статися під час роботи системи, недостатня кількість працівників для обробки та відправки товару до покупця. У такому разі система повертає замовлення до черги замовлень служби доставки. Де буде знову перевіряти наявність вільних. Також при моделюванні нашої системи будемо враховувати часові затримки, при генерації замовлення, тобто кожне наступне замовлення буде генеруватись через деякий час, інші часові затримки, це при відправленні товару, та при поповненні складу, у разі недостатності товару для виконання замовлення.

2. 3.Алгоритм представлення системи засобами мереж Петрі

Для того, щоб представити систему засобами мереж Петрі, потрібно:

- виділити події, що виникають в системі, і поставити у відповідність

кожній події перехід мережі Петрі;

- з'ясувати умови, при яких виникає кожна з подій, і поставити у відповідність кожній умові позицію мережі Петрі;
- визначити кількість фішок у позиції мережі Петрі, що символізує виконання умови;
- з'єднати позиції та переходи відповідно до логіки виникнення подій у системі: якщо умова передує події, з'єднати в мережі Петрі відповідну позицію з відповідним переходом; якщо умова є наслідком виконання події, з'єднати в мережі Петрі відповідний перехід з відповідною позицією;
- з'ясувати зміни, які відбуваються в системі при здійсненні кожної події, і поставити у відповідність змінам переміщення визначеної кількості фішок із позицій в переходи та з переходів у позиції;
- визначити числові значення часових затримок в переходах мережі;
- визначити стан мережі Петрі на початку моделювання.

Математичне представлення мережі Петрі забезпечуються описом матриць входів і виходів, а також початкової розмітки, яка визначає стан та умови функціонування мережі на початку моделювання. Матриця входів D^- складається з елементів – кількості зв'язків між позицією P_i та переходом T_j . А матриця виходів D^+ складається з елементів – кількості зв'язків між переходом T_j та позицією P_i .

Система масового обслуговування (СМО) - це система, що займається обслуговуванням заявок (вимог). Найчастіше системи масового обслуговування можна зустріти у виробництві, бізнесі, сфері послуг. Прикладами систем масового обслуговування можуть служити квиткові каси, магазини, перукарні, кафе [13].

Як правило, в завданнях теорії систем масового обслуговування потік заявок (вимог), що входить, рахують пуассоновским (простим).

Для такого потоку частота вступу заявок в систему підкоряється закону розподілу Пуассона. Вірогідність вступу k заявок за час t обчислюється

по формулі 1:

$$P_k(t) = (\mu t)^k / k! e^{-\mu} \quad (1)$$

Важливою характеристикою будь-якої системи є час обслуговування заявок в системі. Час обслуговування однієї заявки - випадкова величина, вона може бути описана законом розподілу. Частіше всього час обслуговування заявок вважають розподіленим за експоненціальним законом, функція розподілу якого має вигляд:

$$F(t) = 1 - e^{-\mu} \quad (2)$$

СМО з очікуванням діляться на замкнуті і незамкнуті. До замкнутих відносяться системи, в яких потік заявок виникає в самій системі, в незамкнутих системах потік заявок приходить із зовні.

В якості основних показників ефективності роботи СМО виступають вірогідність того, що усі канали вільні або зайняті, середня довжина черги, коефіцієнти зайнятості і простою каналів обслуговування.

Розглянемо параметр $\alpha = 1/m$. Якщо $\alpha/n < 1$, то черга не може рости безмежно.

Найважливіші характеристики роботи СМО :

$$1. \quad P_0 = [\sum a^k/k! * a^n/n! (1-a/n)]^{-1} \quad (3)$$

2. Вірогідність того, що зайнято рівно k обслуговуючих каналів за умови, що загальне число вимог, що знаходяться на обслуговуванні, не перевершує числа обслуговуючих апаратів :

$$P_k = a^k/k! P_0, \quad \text{при } 1 \leq k \leq n \quad (4)$$

3. Вірогідність того, що в системі знаходиться e вимог у разі, коли їх число більше числа обслуговуючих каналів :

$$P_k = a^k/n! n^{k-n} P_0, \quad \text{при } k \geq n \quad (5)$$

4. Вірогідність того, що усі обслуговуючі канали зайняті :

$$P_n = a^n/n! (1-a/n) P_0 :: (a/n < 1) \quad (6)$$

5. Середній час очікування вимогою початку обслуговування в системі:

$$t_{об} = Pn/\mu(n-\alpha) \quad (7)$$

6. Середнє число вільних від обслуговування каналів :

$$N_0 = \sum n - k/k! \quad (8)$$

7. Коефіцієнт простою каналів :

$$K_{\text{пр}} = N_0/n \quad (9)$$

8. Середнє число зайнятих обслуговуванням каналів :

$$N_0 = n. \quad (10)$$

9. Коефіцієнт завантаження каналів:

$$K_a = Na/n \quad (11)$$

Кожна система масового обслуговування – це одне або декілька обслуговуючих пристроїв з чергою. У розімкнутій мережі МО вимоги надходять ззовні мережі і після обробки залишають її. У замкнутій мережі МО деяка кількість вимог весь час знаходиться в ній, переходячи з однієї СМО до іншої, але ніколи не залишаючи мережу МО.

Мережі масового обслуговування функціонують за наступними простими правилами. Вимоги, що надходять на обслуговування, проходять одну за одною СМО згідно вказаного маршруту. На вході кожної СМО вимога намагається знайти один вільний пристрій з декількох паралельно функціонуючих. Правило, за яким вимога обирає той чи інший пристрій за умови, що вони вільні, може бути вказано окремо у відповідності до процесу функціонування реальної системи. Якщо всі пристрої однакові, то вимога переглядає пристрої по порядку і займає перший, що виявився вільним. Якщо в момент надходження вимоги до СМО всі пристрої виявились зайнятими, то вимога при наявності черги у СМО намагається в неї потрапити, або залишає мережу масового обслуговування і вважається не обслугованою.

2.4. Методи моделювання систем масового обслуговування

При графічному представленні мережі МО користуються позначеннями, представленими на рисунку 2.3, 2.4.

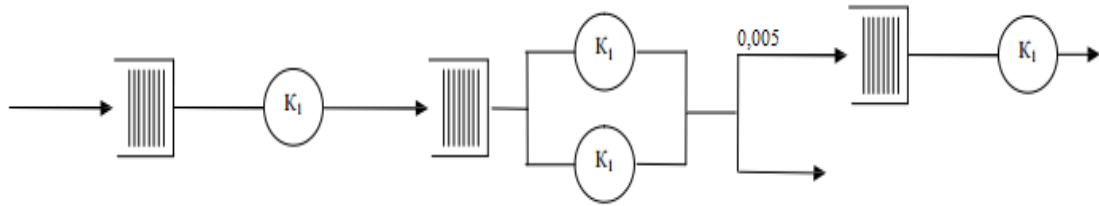


Рис.2.3 - Мережа масового обслуговування

Черга у СМО обов'язково одна. Кількість місць у черзі може бути обмежена або необмежена. Якщо в реальному процесі не виникає проблеми з наявністю вільного місця у черзі, то при моделюванні така черга може вважатись необмеженою, навіть якщо місць у ній декілька. Отже, обмеження на кількість місць у черзі вводять, коли потрібно моделювати ситуацію в даний момент вимога не може бути прийнята у СМО.

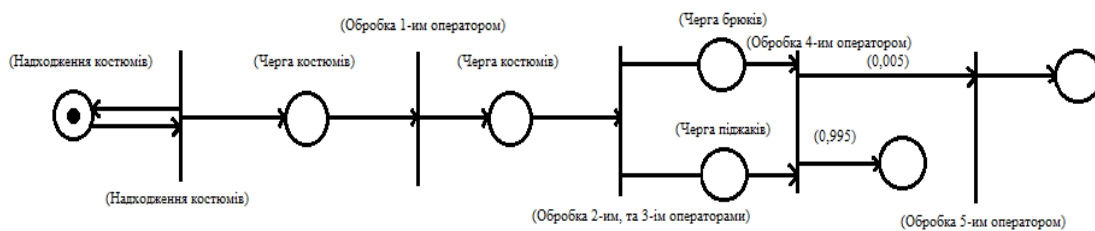


Рис. 2.4 - Мережа Петрі

Мережа масового обслуговування може мати такі ускладнюючі елементи, як декілька типів вимог, що обслуговуються, розгалуження та блокування маршрутів, черги із складними правилами упорядкування вимог у них і т.і.

Декілька типів вимог вводять у разі, якщо потрібно розрізняти час обслуговування у пристроях для різних типів вимог, або маршрути слідування (у тому числі ймовірності слідування від однієї СМО до іншої).

Розгалуження маршруту означає, що маршрут вимоги може бути визначений тільки з визначеною ймовірністю. Сума ймовірностей у місці розгалуження маршруту складає одиницю, тому на одній із гілок розгалу-

ження допускається, що ймовірність не указується.

Блокування маршрутів допомагають ввести в мережу МО елементи, що управляють процесом обслуговування. Наприклад, вилучити поламаний пристрій з процесу обслуговування, заборонити подальше просування вимоги до виконання деякої умови і т.і.

Черги із складними правилами упорядкування розглядаються у випадках, коли потрібно установити пріоритети обслуговування вимог в залежності від, наприклад, типу вимоги, накопленого часу обслуговування вимоги і т.і.

Основна задача ТСМО полягає у встановленні залежності між характером потоку заявок на вході СМО, продуктивністю одного каналу, числом каналів і ефективністю обслуговування. При цьому критеріями ефективності СМО можуть бути різні функції і величини:

- середній час простою системи;
- середній час очікування у черзі;
- закон розподілу тривалості очікування заявки у черзі;
- середній відсоток заявок, яким відмовили і т. д.

Вибір критеріїв залежить від виду системи.

Система обслуговування вважається заданою, якщо відомі: потік вимог Π , множина обслуговуючих приборів S , дисципліна обслуговування D .

Таблиця 1

Закон розподілу інтервалів часу між надходженням заявок

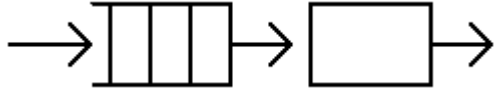
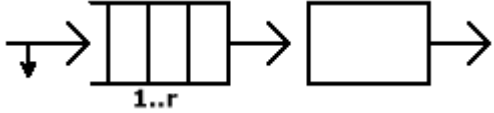
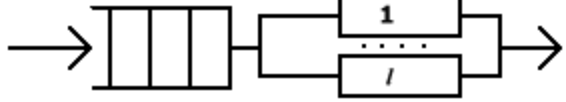
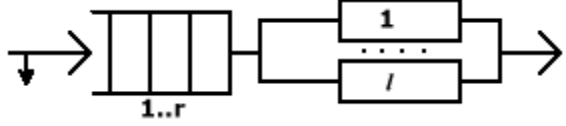
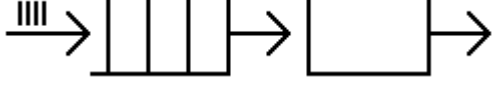
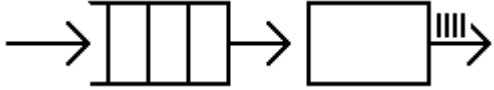
	Закон розподілу інтервалів часу між надходженням заявок.	<i>M</i> - експонентний <i>E</i> - Ерланга
<i>B</i>	Закон розподілу часу обслуговування в приладах СМО.	<i>H</i> - Гіперекспоненціальний <i>\Gamma</i> - гамма-розподіл <i>D</i> - Детерміноване розподіл <i>G</i> - довільний розподіл
<i>C</i>	Число обслуговуючих приладів.	<i>1</i> - для одноканальних систем

		<i>l</i> - Для багатоканальних систем
D	Число місць у черзі. Якщо число місць не обмежена, то поле можна опустити.	<i>r</i> або <i>n</i> - для кінцевого числа місць
E	Дисципліна обслуговування. За замовчуванням LIFO - у цьому випадку поле може опускатися.	<i>FIFO, LIFO, RANDOM</i>

Таблиця 1- Таблиця відповідностей законів характеристикам СМО

Для скороченої запису при моделюванні часто використовуються наступні типи схем СМО:

Таблиця 2 – Типи схем СМО

Найменування	Позначення	Схема
Одноканальна з очікуванням	G G 1	
Одноканальна з втратами	G G 1 r	
Багатоканальна з очікуванням	G G l	
Багатоканальна з втратами	G G l r	
Одноканальна з груповим надходженням заявок	Gr G 1 <i>r</i>	
Одноканальна з груповим обслуговуванням заявок	G Gr 1 <i>r</i>	

Висновки до другого розділу.

Імітаційне моделювання систем, як метод, є найбільш практично застосовуваним для дослідження складних систем. В розділі розглядаються алгоритми імітаційного моделювання систем, що базуються на представленні процесу функціонування системи засобами мереж масового обслуговування та засобами мереж Петрі. Було проведено порівняльний аналіз моделювання мережами Петрі та системами масового обслуговування

У другому розділі було проведено огляд програмних засобів імітаційного моделювання, описані найважливіші характеристики роботи СМО, поняття мереж Петрі та систем масового обслуговування, призначення, вимоги та класифікація систем масового обслуговування та представлені основні задачі теорії систем масового обслуговування. та типи схем СМО.

РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ МОДЕЛЮВАННЯ

3. 1. Програмне забезпечення комп'ютерного експерименту.

Програмне забезпечення комп'ютерного експерименту включає різні середовища програмування (Pascal, Delphy, Fortran, Visual Basic), системи комп'ютерної математики (MathCad, MATLAB), спеціалізовані програмні продукти, орієнтовані рішення конкретних прикладних завдань у конкретній області, та інших. В останні роки все більшої популярності набувають системи комп'ютерної математики – програми, що виникли на стику математики та інформатики та призначені для вирішення та візуалізації математичних завдань. Система Matlab (MATrix LABoratory – матрична лабораторія) займає серед них особливе місце як за ступенем універсальності, так і складністю. За великою кількістю функцій та швидкості обчислень MATLAB перевершує більшість подібних систем і є безперечним лідером у галузі чисельних розрахунків та математичного моделювання різних систем та пристроїв. Система MATLAB може бути обрана та ефективно використана як середовище вирішення реальних інженерних завдань. У MATLAB реалізовано чисельні алгоритми розв'язання задач лінійної алгебри, рівнянь, знаходження значень певних інтегралів, апроксимації, розв'язання систем або окремих диференціальних рівнянь [24]. MATLAB має добре розвинені можливості візуалізації двовимірних і 18 тривимірних даних, що дозволяє створювати власний графічний інтерфейс користувача. Робота в середовищі MATLAB може здійснюватися у двох режимах: – у командному режимі (калькулятора чи режимі безпосередніх обчислень), тобто. обчислення проводяться безпосередньо після набору чергового оператора; – шляхом виклику програми, записаної мовою MATLAB. Проста вбудована мова програмування дозволяє легко створювати власні алгоритми. Матрична форма подання операцій робить програмування у середовищі MATLAB дуже лаконічним, що знижує трудомісткість робіт та підвищує швидкість обчислень. Існує велика кількість пакетів прикладних програм, що розширюють та доповнюють можливості MATLAB, призначених для вирішення завдань з тієї чи іншої галузі науки або

техніки.

3.2. Програмні засоби моделювання мереж Петрі

Проблема моделювання мережами Петрі в даний час дуже актуальна. У зв'язку з цим створюється велика кількість програмних продуктів (ПП), які зачіпають різні аспекти використання мереж Петрі. Співвідношення комерційних і вільно розповсюджуваних програм на комп'ютерному ринку в цій області приблизно рівне, їх кількість досить велика і з часом лише зростає.

Якісних безкоштовних програм по мережах Петрі в українській частині Інтернету знайти практично неможливо. А от у Європі та Америці можна знайти чимало цілих каталогів програм, доступних для скачування.

Проаналізуємо такі програмні продукти:

- **Petri.NetSimulator 2.0** – Програма написана на Delphi, дозволяє будувати мережу Петрі на екрані за допомогою миші, проводити синтез мережі.

Для позицій є можливість завдання кількості маркерів, для переходів – часового розподілу (задається миттєво, детерміновано або стохастично).

Переваги системи: можливість аналізу широкого спектра властивостей побудованої мережі Петрі, анімаційне зображення активних переходів.

До недоліків даного програмного забезпечення можна віднести: незручний редактор, потребу у використанні застарілих бібліотек платформи .NET версії 1.1 (найновіша на сьогоднішній день – 4.5.5), відсутність можливості ручного налаштування зовнішнього вигляду схеми через автоматичне розміщення стрілок.

На рис.3.1 представлений приклад мережі Петрі у середовищі Petri.NetSimulator 2.0.

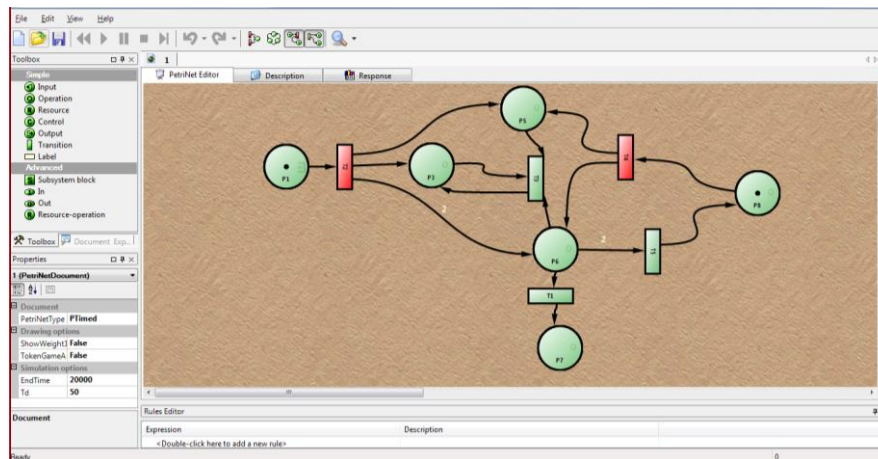


Рис.3.1. Приклад мережі в системі Petri.NetSimulator 2.0

VisualPetri – багатофункціональна система, оскільки пропонує користувачу механізми аналізу мережі, котрі включають в себе побудови графа досяжності, для заданої мережі та її довільного початкового стану (розмітки) і знаходження S і T інваріантів, які служать для аналізу мережі на обмеженість, досяжність, наявність/ відсутність циклів.

Переваги даного проекту: підтримка роботи з дворівневими мережами Петрі та підмережами, простота і зрозумілість інтерфейсу.

Однак даний програмний продукт відсутній для скачування у мережі Інтернет, тому у порівняльній характеристиці програмних продуктів брати участь не буде.

- **VisualPetri** – однойменна розробка, яка представляє собою графічний редактор мереж Петрі для Windows, платформу на основі бібліотеки GDI Plus з інтегрованим симулятором. Даний програмний продукт належить до програмного забезпечення Open Sourc. [28].

Переваги програми: простота, можливість масштабування мережі,

До недоліків програми належать: слабкі функціональні можливості, відсутність математичного моделювання, незручність з'єднання елементів, відсутність можливості підписувати елементи та додавати коментарі до них.

На рис.3.2 представлений приклад мережі Петрі у середовищі VisualPetri.

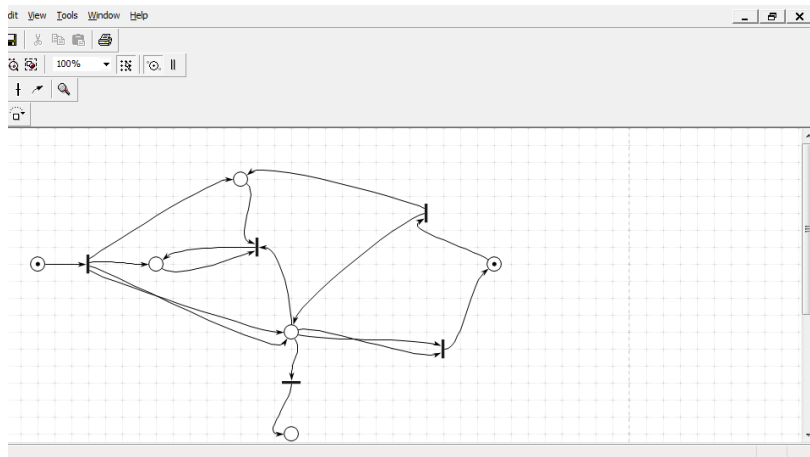


Рис. 3.2 Приклад мережі в системі VisualPetri

▪ **Hpsim1_1** – графічний редактор, розроблений мовою Visual C++, який забезпечує моделювання та базове редагування мережі Петрі. Інструмент є корисним для початківців, таких як студенти, з метою ознайомлення з мережами Петрі. Розробник – Henryk Anschuetz, Німеччина. Розробка підтримує класичні мережі Петрі та моделювання з тимчасовим параметром.

Для елемента «операція» можливо задати такі опції: розмір; кількість маркерів; можливість приховати/відобразити ім'я, час моделювання.

Переваги програми: програма вільна для скачування, є можливість коментувати елементи мережі, можливість приховувати/відображати ємність та назви позицій і переходів[36,37].

Недоліки даної програми: відсутність можливості повороту елементів та математичного моделювання мережі.

На рис. 3.3 представлений приклад мережі Петрі у середовищі Hpsim1_1.

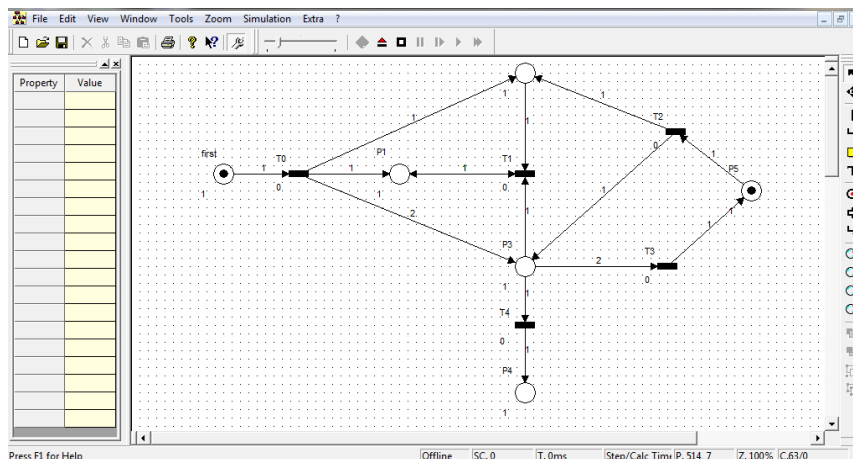


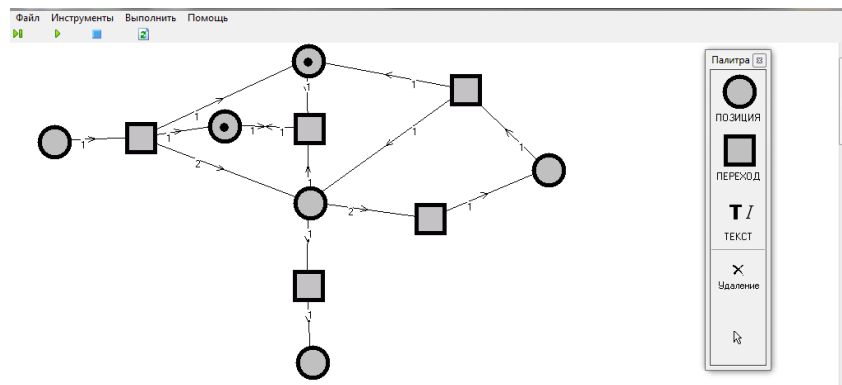
Рис.3 3. Приклад мережі в системі Hpsim1_1

▪ **Petriemulopensource** – проект з відкритим вихідним кодом під ліцензією GNU Public License. Програма розроблена НТУУ «КПІ» і написана на Java. Цей настільний додаток використовується для редагування і емуляції мереж Петрі. Він має графічний інтерфейс і може бути використаний для аналізу даних, діагностики і т.д.

До переваг даної програми можна віднести: можливість збереження мережі Петрі в математичному представленні в текстовому файлі, зрозумілу візуалізацію покрокової роботи мережі, можливість задати для переходів пріоритет виконання і точку зупинки.

Недоліки програми: не зручне додавання переходів та завдання кількості стрілок, відсутність можливості видалення стрілок та повернення на крок назад (щоб видалити стрілку, потрібно видалити елемент, пов'язаний з нею).

На рис.3.4. представлений приклад мережі Петрі у середовищі Petriemulopensource.

Рис.3.4. Приклад мережі в системі *Petriemulopensource*

Порівняльна характеристика розглянутих вище безкоштовних програмних засобів для використання в початковому процесі при моделюванні мереж Петрі наведена в табл. 1.

Таблиця 1. Порівняльна характеристика безкоштовних програмних засобів для моделювання мереж Петрі [39].

Показники	Програмні продукти			
	<i>VisualPetr i</i>	<i>hpsim1_1</i>	<i>Petri.Net Simulator 2.0</i>	<i>Petriemulop ensource</i>
1. Вимоги до інтерфейсу				
Можливість підписувати назви переходів чи подій	—	Автоматично / можливість змінювати	Автоматично	—
Можливість повороту об'єкта	+	—	+	Немає необхідності повертати (переходи квадратні)
Можливість вигинати стрілки	+	Прямі лінії переходу	Автоматично	Прямі лінії переходу
Кількість стрілок задається	-	Цифрами	Цифрами	Цифрами
Мова інтерфейсу/надписів	Англійська / —	Англійська / латиниця	Англійська/ кирилиця, латиниця	Російська/ кирилиця, латиниця
Загальний бал за інтерфейс (за 5 бальною шкалою)	3	4	4	3

Продовження таблиці 1.

2. Функціональні можливості				
Задання властивостей в позиціях і переходах	—	+	+	+
Можливість задати початкову кількість маркерів	Цифрами	Цифрами	Цифрами	Цифрами
Математичне моделювання	—	—	+	+
Можливість покроково відстежувати роботу моделі	+	+	+	+
Загальний бал за функціональні можливості (за 5 бальною шкалою)	3	3	3	4
3. Вимоги до ліцензії				

Безкоштовна	+	+	+	+
4. Системні вимоги				
Операційна система	Windows	Windows	Windows	Windows

3.3. Моделювання системи в CPN Tools

CPN Tools - це спеціальна моделююча система, яка використовує мову мереж Петрі для опису моделей. Система була розроблена в Університеті Орхуса в Данії і вільно поширюється для некомерційних організацій. Рівень сервісу дозволяє класифікувати CPN Tools як промислову систему. Вона була використана в великій кількості реальних проектів, особливо в галузі телекомунікації.

Такий вигляд має модель магазину (рис.3.5) при замовленні деякого товару, вона нічим не відрізняється від раніше побудованої моделі. У цій моделі ми маємо декілька типів змінних, 1-й тип це товар, 2-й тип це робітники. За значення змінної при генеруванні замовлення ми беремо деяке число у діапазоні від 5 до 20. Тобто генерується деяке замовлення товару кількістю. Якщо ж на складі не можливо виконати замовлення генерується інше замовлення на поповнення складу товаром, якого не вистачає. Також ми додали у систему так звані часові затримки, бачимо, що при генерації замовлення використовуємо її, а також при відправці замовлення, та поповнення запасу складу.

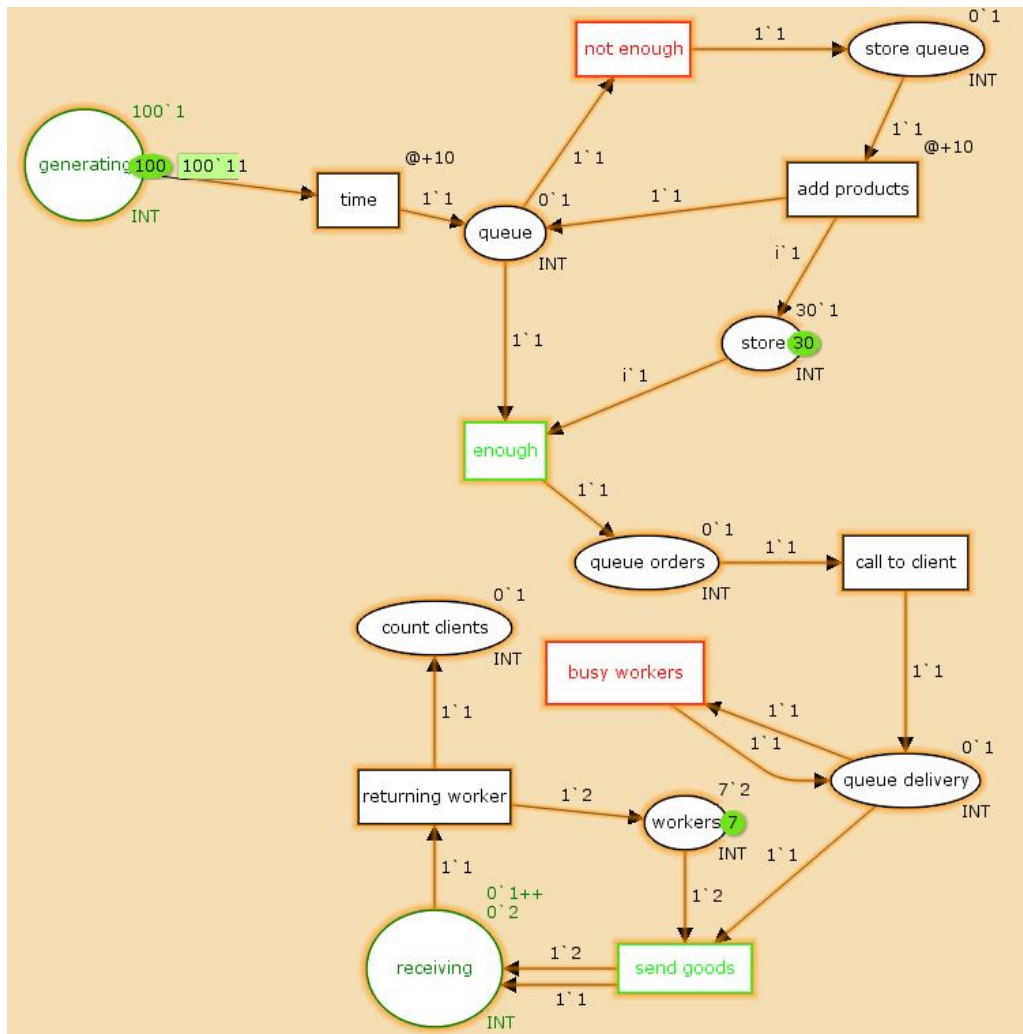


Рис 3.5 Модель роботи магазину

Програмна реалізація. Програмне забезпечення комп'ютерного експерименту включає різні середовища програмування (Pascal, Delphy, Fortran, Visual Basic), системи комп'ютерної математики (MathCad, MATLAB), спеціалізовані програмні продукти, орієнтовані рішення конкретних прикладних завдань у конкретній області, та інших. Порівняльна характеристика розглянутих вище безкоштовних програмних засобів для використання при моделюванні мереж Петрі наведена в табл. 1.

Крім побудови системи купування товару у магазині за допомогою графів Петрі, реалізуємо за допомогою побудованої схеми та мови програмування Python візуалізацію роботи системи. Для слідкування за замовлення побудували зручний інтерфейс з базою даних складу цукерок, записами зі змінами на складі, також є інформація про всі замовлення, та про

службу доставки.

Store

	Name	Much	Cost	Data
1	Romashka	109 kg	23 \$	22 days
2	Bitter Sweet	177 kg	32 \$	27 days
3	Waffles	435 kg	15 \$	0 days
4	Snickers	127 kg	39 \$	17 days
5	Bounty	228 kg	40 \$	29 days
6	Lion	294 kg	32 \$	1 days
7	Nuts	330 kg	37 \$	21 days

Store Update

- > Store get 179 kg Snickers
- > Store delete 140 kg Nuts
- > Store get 443 kg Nuts
- > Store delete 146 kg Romashka
- > Store get 198 kg Romashka
- > Store delete 41 kg Bitter Sweet
- > Store get 177 kg Bitter Sweet
- > Store delete 113 kg Bounty
- > Store get 228 kg Bounty
- > Store delete 435 kg Waffles

Journal

- der 15 Adam Jonson order Waffles 53 kg sum=795 \$ [card: 97346758]
- der 16 Steven Gerrard order Bitter Sweet 40 kg sum=1088 \$ [card: 95736582]
- der 17 Adam Jonson order Snickers 52 kg sum=2028 \$ [card: 97346758]
- der 18 Daniel Sturidge order Bounty 68 kg sum=2312 \$ [card: 94500361]
- der 19 Jordan Henderson order Nuts 47 kg sum=1739 \$ [card: 94644820]
- der 20 Adam Jonson order Waffles 94 kg sum=1198 \$ [card: 97346758]
- der 21 Martin Skrtel order Romashka 89 kg sum=2047 \$ [card: 93681945]

Delivery

Evgenii Petrenko take order 14, street Cat 34 door 182
 Vladislav Regi take order 15, street Green Avenue 26 door 2
 Serhii Soin take order 16, street Lombard.Str 110 door 23
 Vladislav Regi take order 17, street Green Avenue 26 door 2
 Alexandr Kuhn take order 18, street Cat 34 door 182
 Serhii Nosov take order 19, street Lime.Str 60 door 93
 Vladislav Regi take order 20, street Green Avenue 26 door 2
 Evgenii Petrenko take order 21, street Strand 65 door 26

Next Order
 Cancel Order

STORE COUNT 43862 \$
 PROFIT 5496 \$

Далі запрограмуємо, щоб за деякий час генерувалося 1 замовлення, згідно за моделлю побудованою в CPN Tools, як на малюнку 3.6.

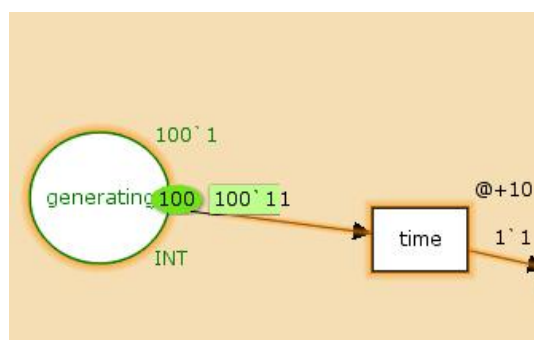


Рис 3.6. Генерування замовлення

Далі створюємо базу даних заповнюючи її даними, ми будемо продавати цукерки 7 типів, у нас будуть 7 тестових покупців, та 5 робітників служби доставки. Наступним кроком будемо інтерфейс програми, де ми можемо слідкувати за роботою нашої системи. Отже, спочатку виведемо таблицю товарів, де ми можемо слідкувати за їх кількістю та терміном придатності.

	Name	Much	Cost	Data
1	Romashka	330 kg	23 \$	11 days
2	Bitter Sweet	192 kg	32 \$	16 days
3	Waffles	630 kg	15 \$	19 days
4	Snickers	280 kg	39 \$	6 days
5	Bounty	400 kg	40 \$	18 days
6	Lion	350 kg	32 \$	21 days
7	Nuts	214 kg	37 \$	10 days

Рис 3.7 Представлення складу

У цій таблиці ми можемо слідкувати за запасами, якщо кількість запасів не вистачає для виконання замовлення, ми виконуємо процес поповнення складу як на малюнку 3.8. Так як ми продаємо цукерки, вони мають термін придатності, у разі закінчення його, знову вмикається процес поповнення складу, разом з видаленням непридатних до вживання цукерок.

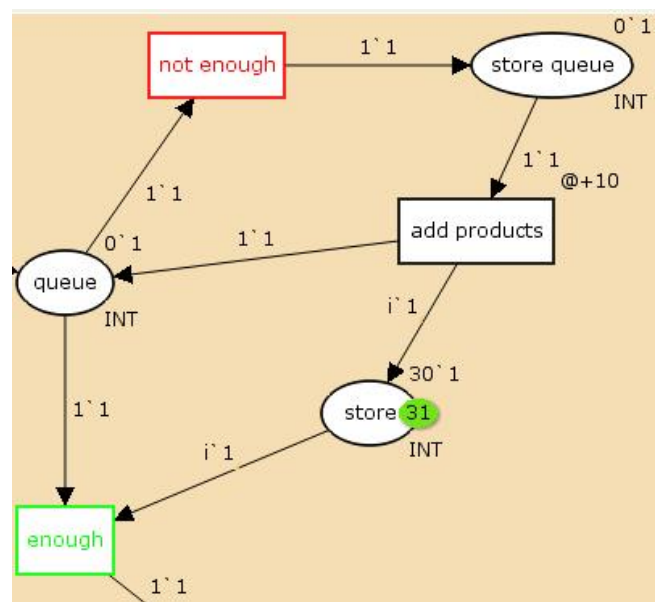


Рис 3.8. Представлення процесу поповнення складу

Далі створюємо вікно у якому будуть відзначатися зміни у нашому складі, тобто це змодельований процес поповнення запасів складу, у разі нестачі та якщо термін придатності закінчився, програма видаляє залишок цукерок зі складу, та закуповує нову партію.

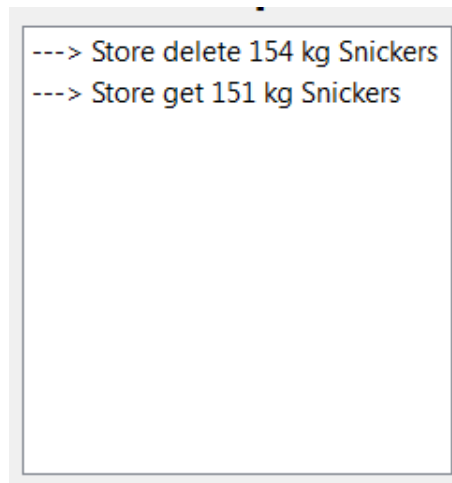


Рис 3.9. Вікно поповнення запасів складу

Інші вікна які ми створюємо, будуть контролювати наші замовлення, перше з них , це вікно-журнал у якому виводимо список замовлень, з даними покупця. Та друге вікно служби доставки, у якому відображаються відправка замовлень, працівниками, у якому ми можемо бачити які працівники приймають замовлення, та адресу доставки.

Order 1	Steven	Gerrard	order Nuts 54 kg	sum=1998 \$	[card: 95736582]
Order 2	Steven	Gerrard	order Romashka 20 kg	sum=460 \$	[card: 95736582]
Order 3	Adam	Lallana	order Snickers 68 kg	sum=2254 \$	[card: 94627483]
Order 4	Martin	Skrstel	order Bitter Sweet 84 kg	sum=2688 \$	[card: 93681945]
Order 5	Jordan	Henderson	order Bounty 46 kg	sum=1840 \$	[card: 94644820]
Order 6	Adam	Lallana	order Bounty 87 kg	sum=3480 \$	[card: 94627483]
Order 7	Martin	Skrstel	order Nuts 61 kg	sum=1918 \$	[card: 93681945]
Order 8	Adam	Jonson	order Snickers 58 kg	sum=2262 \$	[card: 97346758]

Рис 3.10.Список записів про замовлення

Якщо у вікні з'явилося повідомлення про замовлення , отже наше замовлення знаходиться у черзі замовлень, де робітники повинні подзвонити покупцю та домовитись про доставку замовлення рис. . 3.11.

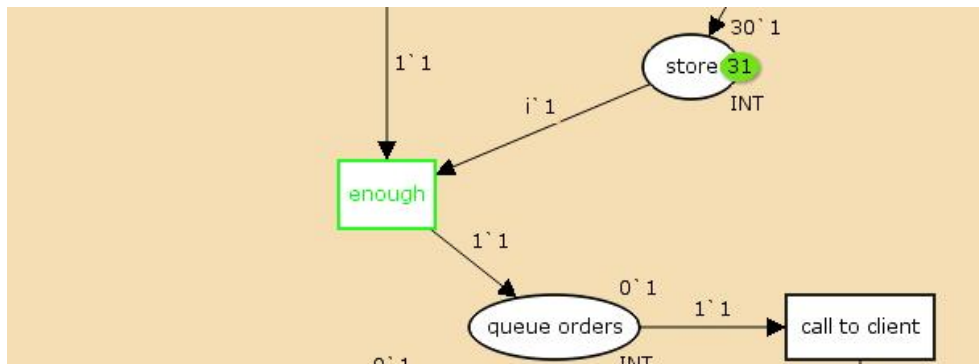


Рис. 3.11 Список записів про замовлення

Далі розглянемо модель яка розроблена у CPN Tools, бачимо що замовлення надходить до служби доставки, де робітники доставляють замовлення за адресом вказаним покупцем, тому розробимо вікно, у якому ми можемо слідкувати за доставкою замовлень, та працівниками служби.

У вікні виводиться інформація про працівника який займається доставкою замовлення, номер замовлення, та адреса покупця. Якщо усі працівники працюють над замовленнями, і одразу доставити замовлення ми не можемо, програма чекає на повернення працівника на своє робоче місце. Та виводить інформацію у вікно.

3.4. Розробка програмних модулів на мові GPSS та Python з використанням бібліотеки `rm4py`.

1. Мова моделювання GPSS.

Симуляційне моделювання складних систем є важливим інструментом для дослідження, аналізу та оптимізації процесів у різних галузях. GPSS (General Purpose Simulation System) — це одна з мов програмування, спеціально розроблена для симуляційного моделювання дискретних подій у системах, широко застосовується для аналізу складних систем [17,36].

Основні особливості мови GPSS:

1. Мова моделювання: GPSS дозволяє створювати моделі систем з дискретними подіями, що включає події, які відбуваються у визначені моменти часу.
2. Об'єктно-орієнтований підхід: Моделі складаються з різних об'єктів, таких як блоки, які представляють різні дії або процеси у системі (надходження, черга, обробка).
3. Симуляційне середовище: GPSS забезпечує середовище для виконання симуляцій, де користувач може задавати вхідні параметри, керувати виконанням симуляції та аналізувати результати.
4. Моделювання потоків: GPSS моделює потоки об'єктів, що проходять через різні блоки (наприклад, клієнти, що проходять через обслуговування в банку).

Основні етапи створення моделі в GPSS:

1. Визначення проблеми: Спочатку необхідно чітко визначити систему, яка моделюється та які питання або проблеми потрібно вирішити.
2. Розробка моделі: Створення блок-схеми, що описує всі компоненти системи та їх взаємодію. Потім ця схема кодується за допомогою GPSS.
3. Кодування: Введення моделі у середовище GPSS з використанням його специфічної мови. Тут використовується набір стандартних блоків, таких як GENERATE (генерація об'єктів), QUEUE (черга), SEIZE (захоплення ресурсу), RELEASE (звільнення ресурсу), і TERMINATE (завершення обробки).
4. Виконання симуляції: Модель запускається і GPSS обробляє симуляцію на основі заданих параметрів.
5. Аналіз результатів: Результати симуляції використовуються для аналізу ефективності системи, виявлення вузьких місць, прогнозування поведінки під різними умовами.

Застосування GPSS:

- Виробництво: Моделювання виробничих ліній, оптимізація використання обладнання, аналіз черг і потоків робочих процесів.
- Логістика і транспорт: Моделювання складських систем, транспортних мереж, оптимізація розподілу ресурсів.
- Сервісні системи: Аналіз обслуговування в банках, лікарнях, кол-центрах, супермаркетах.
- Комп'ютерні системи: Моделювання продуктивності комп'ютерних мереж, систем управління даними.

GPSS є потужним інструментом для аналізу і оптимізації складних систем, особливо коли важливо враховувати дискретні події та їх взаємодію у часі[16].

Приклад 1.

Приклад симуляції черги в банку, де клієнти прибувають, очікують на обслуговування та обслуговуються касирами.

Опис системи:

- Клієнти прибувають в банк через випадкові проміжки часу, розподілені за експоненціальним законом з середнім інтервалом в 5 хвилин.
- В банку працює один касир, час обслуговування кожного клієнта також розподілений експоненційно із середнім часом обслуговування 3 хвилини.
- Якщо касир зайнятий, клієнти чекають в черзі.

Програма на GPSS:

* GPSS програма для симуляції роботи касира в банку

* Визначення вхідного потоку клієнтів

GENERATE 5,1 * Клієнти прибувають в середньому кожні 5 хвилин з варіацією 1 хвилини

QUEUE LINE * Клієнти стають в чергу (черга з назвою LINE)

SEIZE SERVER * Клієнти захоплюють сервер (касир з назвою SERVER)
 DEPART LINE * Клієнти покидають чергу
 ADVANCE 3,1 * Касир обслуговує клієнта в середньому 3 хвилини з
 варіацією 1 хвилина
 RELEASE SERVER * Клієнт звільняє касира
 TERMINATE 1 * Завершення обробки одного клієнта

* Вихідні результати симуляції

START 100 * Запустити симуляцію для 100 клієнтів

Пояснення програми:

1. GENERATE 5,1 - генерує клієнтів, які прибувають в середньому кожні 5 хвилин з деякою випадковою варіацією в 1 хвилину.
2. QUEUE LINE - клієнти додаються до черги LINE, якщо касир зайнятий.
3. SEIZE SERVER - клієнт захоплює сервер (касира), якщо той вільний. Якщо сервер зайнятий, клієнт залишається в черзі.
4. DEPART LINE - клієнт покидає чергу, коли переходить до обслуговування.
5. ADVANCE 3,1 - симулює процес обслуговування клієнта касиром, який триває в середньому 3 хвилини з випадковою варіацією в 1 хвилину.
6. RELEASE SERVER - клієнт звільняє сервер після обслуговування.
7. TERMINATE 1 - завершує процес обробки одного клієнта.
8. START 100 - запускає симуляцію для 100 клієнтів.

Результати:

Після виконання цієї програми GPSS покаже статистику черги, використання касира, середній час очікування клієнтів та інші корисні дані. Це допомагає зрозуміти продуктивність системи і, за потреби, внести корективи для оптимізації.

Такі моделі можна використовувати для аналізу і оптимізації роботи будь-яких сервісних систем, де є потоки клієнтів і обмежені ресурси обслуговування.

Для моделювання складних систем з використанням мереж Петрі часто використовуються спеціалізовані інструменти, такі як CPN Tools, PIPE, або Petri Net Toolbox для MATLAB, але ми можемо розглянути приклад програми на Python з використанням бібліотеки pm4py, яка підтримує моделювання та аналіз процесів за допомогою мереж Петрі.

Приклад 2.

Розглянемо виробничий процес з двома послідовними етапами:

1. Машина А виконує початкову обробку продукту.
2. Машина В завершує обробку продукту.

Продукти повинні пройти обробку на обох машинах. Кожна машина має обмежену кількість ресурсів, і якщо ресурс зайнятий, продукти чекають у черзі.

Програма на Python з використанням мереж Петрі:

Для цього прикладу використовуватимемо бібліотеку pm4py, яка підтримує роботу з мережами Петрі.

```
import pm4py
from pm4py.objects.petri import utils
from pm4py.objects.petri.petrinet import PetriNet, Marking

# Створення мережі Петрі
net = PetriNet("Manufacturing Process")

# Створення місць
p1 = PetriNet.Place("p1") # Початок процесу
p2 = PetriNet.Place("p2") # В черзі для Машини А
p3 = PetriNet.Place("p3") # Після обробки Машиною А
p4 = PetriNet.Place("p4") # В черзі для Машини В
p5 = PetriNet.Place("p5") # Завершення процесу

# Додавання місць до мережі
net.places.add(p1)
net.places.add(p2)
```

```

net.places.add(p3)
net.places.add(p4)
net.places.add(p5)

# Створення переходів
t1 = PetriNet.Transition("t1", "Start Processing at Machine A")
t2 = PetriNet.Transition("t2", "Finish Processing at Machine A and Queue for Machine B")
t3 = PetriNet.Transition("t3", "Start Processing at Machine B")
t4 = PetriNet.Transition("t4", "Finish Processing at Machine B")

# Додавання переходів до мережі
net.transitions.add(t1)
net.transitions.add(t2)
net.transitions.add(t3)
net.transitions.add(t4)

# Створення дуг (зв'язків) між місцями та переходами
utils.add_arc_from_to(p1, t1, net) # Від початку до першого переходу
utils.add_arc_from_to(t1, p2, net) # Від першого переходу до черги для Машини А
utils.add_arc_from_to(p2, t2, net) # Від черги для Машини А до завершення обробки
Машиною А
utils.add_arc_from_to(t2, p3, net) # Від завершення обробки Машиною А до черги для
Машини В
utils.add_arc_from_to(p3, t3, net) # Від черги для Машини В до початку обробки
Машиною В
utils.add_arc_from_to(t3, p4, net) # Від початку обробки Машиною В до завершення
utils.add_arc_from_to(p4, t4, net) # Завершення обробки Машиною В
utils.add_arc_from_to(t4, p5, net) # Від завершення до кінця процесу

# Визначення початкового маркування (розміщення маркерів)
initial_marking = Marking()
initial_marking[p1] = 1 # Маркер на початку процесу

# Визначення кінцевого маркування
final_marking = Marking()
final_marking[p5] = 1 # Маркер в кінцевому місці

# Візуалізація мережі Петрі
pm4py.view_petri_net(net, initial_marking, final_marking)

```

Пояснення коду:

1. Створення мережі Петрі: Ініціалізація нової мережі Петрі з назвою "Manufacturing Process".
2. Місця: Визначають стани або етапи процесу (наприклад, черга для обробки на Машині А або В).
3. Переходи: Визначають події або дії, що змінюють стан системи (наприклад, початок або завершення обробки).
4. Дуги: Зв'язки між місцями і переходами, що показують потік процесу.
5. Маркування: Визначає початковий стан системи і кінцевий стан для завершення моделювання.
6. Візуалізація: Відображає мережу Петрі для наочного аналізу.

Використання і результати:

- Ця модель допоможе відслідковувати потоки продуктів через процес обробки, визначати затримки в чергах, і оцінювати завантаження машин.
- Аналіз може включати виявлення вузьких місць (bottlenecks), оптимізацію ресурсів і покращення загальної ефективності виробництва.

Ця програма демонструє основи створення і використання мереж Петрі для моделювання складних систем у Python, що є універсальним підходом для симуляційних задач в реальному світі.

Приклад 3.

Розглянемо повний цикл створення та верифікацію СМО на прикладі хімчистці.

В ході моделювання повинні визначити наступні показники:

1. максимальну та середню довжину кожної черги;
2. коефіцієнт завантаження кожного оператора
3. максимальний та середній час перебування одягу у хімчистці.

По – перше, треба розробити концептуальну модель системи (рис.3.12).

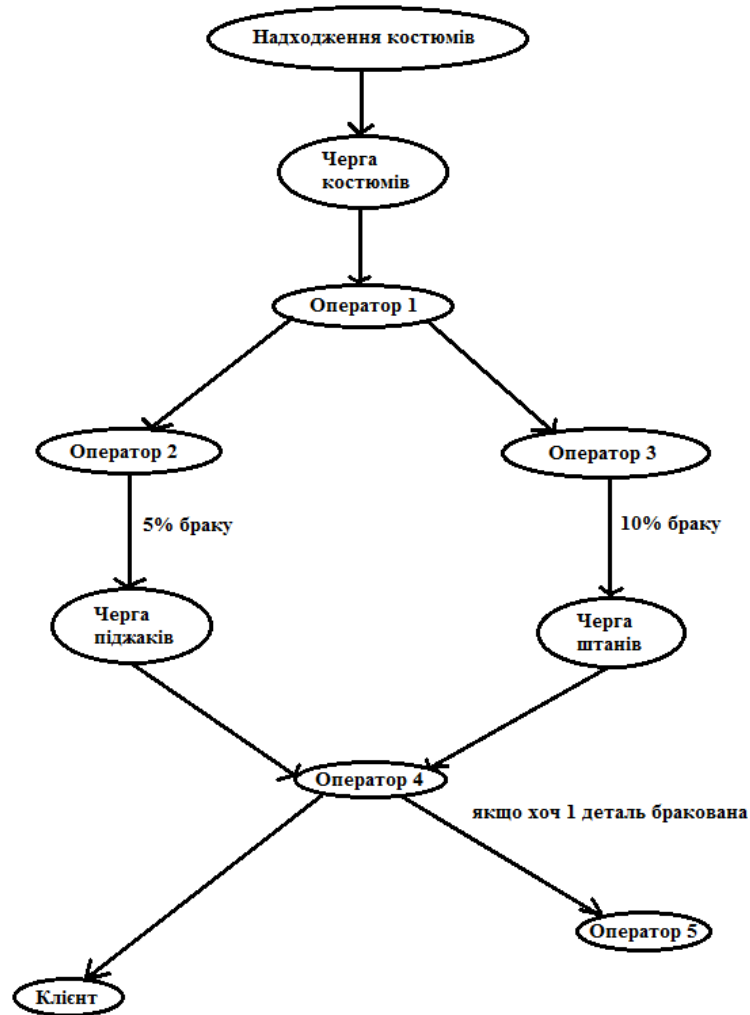


Рис. 3.12. Концептуальна модель СМО.

По- друге, провести формалізацію моделі системи.

Виходячі із умови задачі представимо СМО двома видами мереж:

1. Мережа масового обслуговування:

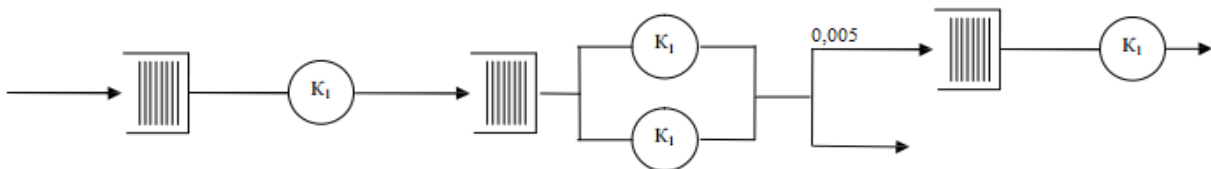


Рис.3.13 - Мережа масового обслуговування

2. Мережа Петрі

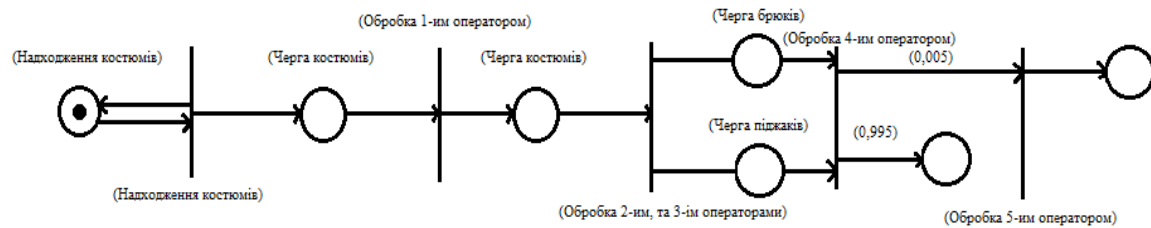


Рис. 3.14 - Мережа Петрі

Третій етап - Алгоритмізація моделі системи і її машинна реалізація мовою GPSS:

GENERATE (EXPONENTIAL(1, 0, 50)) ; генерація костюмів

ADVANCE (EXPONENTIAL(1, 0, 10)) ; надходять в середньому кожні 10 хв

QUEUE STAND ; попадають в чергу

SEIZE OPERATOR_1 ; перший оператор починає роботу

DEPART STAND ; костюм видаляється з черги

ADVANCE (EXPONENTIAL(1, 0, 6)) ; оператор виконує роботу

RELEASE OPERATOR_1 ; оператор завершує роботу

SPLIT 1,OP3 ; костюм розбивається на 2 частини

OP2 SEIZE OPERATOR_2 ; другий оператор починає роботу

ADVANCE (EXPONENTIAL(1, 0, 5)) ; виконує

RELEASE OPERATOR_2 ; завершує

QUEUE JACKET ; додається в чергу для 4 оператора

TRANSFER ,OP4 ; перехід до 4 оператора

OP3 SEIZE OPERATOR_3 ; третій оператор починає роботу

ADVANCE (EXPONENTIAL(1, 0, 4)) ; виконує

RELEASE OPERATOR_3 ; завершує

QUEUE PANTS ; штани додаються до черги 4 оп

TRANSFER ,OP4 ; перехід до 4 оператора

OP4 ASSEMBLE 2 ; очікуємо коли обидві деталі будуть готові

SEIZE OPERATOR_4 ; четвертий оператор починає роботу

DEPART PANTS ; видаляємо з черги штани

DEPART JACKET ; видаляємо з черги піджак

RELEASE OPERATOR_4 ; завершує

TRANSFER 0.15,OP4_OK,OP4_BRAK ; з ймовірністю 0.15 костюм буде пошкоджений

OP4_BRAK SEIZE OPERATOR_4_BRAK ; якщо костюм пошкоджений

ADVANCE (EXPONENTIAL(1, 0, 8)) ; то його обробка триватиме 8 хв
 RELEASE OPERATOR_4_BRAK ; завершення обробки
 TRANSFER ,OP5 ; потім цей костюм переходить до 5 оператора

OP4_OK SEIZE OPERATOR_4_OK ; якщо костюм непошкоджений
 ADVANCE (EXPONENTIAL(1, 0, 5)) ; то його обробка триватиме 5 хв
 RELEASE OPERATOR_4_OK ; завершення обробки
 TRANSFER ,EX ; потім цей костюм переходить до клієнта

OP5 SEIZE OPERATOR_5 ; п'ятий оператор починає роботу
 ADVANCE (EXPONENTIAL(1, 0, 12)) ; виконує
 RELEASE OPERATOR_5 ; завершує

EX TERMINATE 1 ; завершення роботи
 START 1000 ; запуск 1000 симуляцій

Четвертий етап - Верифікація моделі

GPSS World Simulation Report - coursework.62.1

Sunday, 22 September , 2024 19:21:11

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	1002511.752	38	7	0

NAME	VALUE
EX	38.000
JACKET	10004.000
OP2	9.000
OP3	14.000
OP4	19.000
OP4_BRAK	23.000
OP4_OK	29.000
OP5	35.000
OPERATOR_1	10001.000
OPERATOR_2	10002.000
OPERATOR_3	10003.000
OPERATOR_4	10006.000
OPERATOR_4_BRAK	10008.000
OPERATOR_4_OK	10007.000

OPERATOR_5	10009.000
PANTS	10005.000
STAND	10000.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	10000	0	0
	2	ADVANCE	10000	0	0
	3	QUEUE	10000	0	0
	4	SEIZE	10000	0	0
	5	DEPART	10000	0	0
	6	ADVANCE	10000	0	0
	7	RELEASE	10000	0	0
	8	SPLIT	10000	0	0
OP2	9	SEIZE	10000	0	0
	10	ADVANCE	10000	0	0
	11	RELEASE	10000	0	0
	12	QUEUE	10000	0	0
	13	TRANSFER	10000	0	0
OP3	14	SEIZE	10000	0	0
	15	ADVANCE	10000	0	0
	16	RELEASE	10000	0	0
	17	QUEUE	10000	0	0
	18	TRANSFER	10000	0	0
OP4	19	ASSEMBLE	20000	0	0
	20	SEIZE	10000	0	0
	21	RELEASE	10000	0	0
	22	TRANSFER	10000	0	0
OP4_BRAK	23	SEIZE	1523	0	0
	24	DEPART	1523	0	0
	25	DEPART	1523	0	0
	26	ADVANCE	1523	0	0
	27	RELEASE	1523	0	0
	28	TRANSFER	1523	0	0
OP4_OK	29	SEIZE	8477	0	0
	30	DEPART	8477	0	0
	31	DEPART	8477	0	0
	32	ADVANCE	8477	0	0
	33	RELEASE	8477	0	0
	34	TRANSFER	8477	0	0
OP5	35	SEIZE	1523	0	0
	36	ADVANCE	1523	0	0

	37	RELEASE	1523	0	0
EX	38	TERMINATE	10000	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
OPERATOR_1	10000	0.060	6.040	1	0	0	0	0	0
OPERATOR_2	10000	0.049	4.952	1	0	0	0	0	0
OPERATOR_3	10000	0.040	3.999	1	0	0	0	0	0
OPERATOR_4	10000	0.000	0.000	1	0	0	0	0	0
OPERATOR_4_OK	8477	0.041	4.901	1	0	0	0	0	0
OPERATOR_4_BRAK	1523	0.012	7.925	1	0	0	0	0	0
OPERATOR_5	1523	0.018	11.971	1	0	0	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
STAND	3	0	10000	9379	0.004	0.389	6.259	0
JACKET	3	0	10000	5523	0.020	2.020	4.511	0
PANTS	3	0	10000	4477	0.030	3.030	5.486	0

FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
20000	0	10025	16.185	20000	0	1		

5. ВИСНОВКИ

1. Середній / максимальний час становить:

А) для пошкоджених костюмів	—	35.502 / 38.638
Б) для непошкоджених	—	20.422 / 24.851

2. Середня / максимальна довжина черги:

А) черга костюмів для 1 оператора	—	3 / 0.004
Б) черга піджаків для 4 оператора	—	3 / 0.020
В) черга штанів для 4 оператора	—	3 / 0.030

3. Коефіцієнт завантаження оператора:

А) OPERATOR 1	—	0.060
Б) OPERATOR 2	—	0.049
В) OPERATOR 3	—	0.040
Г) OPERATOR 4	—	0.053
Д) OPERATOR 5	—	0.018

Висновки до третього розділу

В третьому розділі було проведено аналіз спеціалізованих програмних продуктів, орієнтованих на рішення конкретних прикладних завдань у конкретній області, показано переваги оглянутих програм, представлена таблиця порівняльних характеристик безкоштовних програмних засобів для моделювання мереж Петрі.

Була розглянута спеціальна моделююча система CPN Tools, яка використовує мову мереж Петрі для опису моделей та вільно поширюється для некомерційних організацій. Робота системи була розглянута на прикладі роботи магазину за допомогою графів Петрі та мови програмування Python для візуалізації роботи системи.

В розділі на прикладі розглянуто основні етапи створення моделі в GPSS та основні особливості використання цієї мови, предсталені приклади розробки моделей Python з використанням бібліотеки pm4py, яка підтримує моделювання та аналіз процесів за допомогою мереж Петрі.

ВИСНОВКИ

В дипломній кваліфікаційній роботі на тему: Аналіз та дослідження складних систем із застосуванням методів моделювання, розглядаються процеси моделювання складних систем на основі імітаційного моделювання.

Модель дає можливість правильно поставити експеримент, а експеримент вже уточнює модель. Одним із видів експерименту виступає імітаційних моделей в системах масового обслуговування. Імітаційні моделі дозволяють швидко та ефективно провести тестування роботи системи при різних умовах за допомогою комп'ютерного експерименту.

Робота містить докладний аналіз наукових та прикладних досліджень у галузі моделювання складних систем, методів аналітичного аналізу літературних джерел та порівняльних характеристик засобів моделювання складних систем; моделювання систем, теорії мереж Петрі, програмування модулів різних видів систем різними мовами програмування та пакетами програмування, експеримент і тестування – при апробації запропонованого методу і засобу.

На основі проведеного дослідження обґрунтована доцільність застосування імітаційного моделювання в прикладних пакетах при моделюванні різних видів складних систем, розглянути спеціальні моделюючі системи, реалізація графів Петрі за допомогою побудованої схеми та мови програмування Python, бібліотеки pm4py, моделювання СМО з використанням системи GPSS .

Крім того в роботі було проведено аналіз найбільш розповсюджених безкоштовних програмних засобів моделювання мереж Петрі, які можуть бути використані при розв'язанні прикладних задач.

Інші методи, такі як математичні, фізичні та інші, є мало застосовуваними для моделювання складних систем. Причинами цього є складність реалізації даних методів та коштовність.

Сформульовані основні вимоги до прикладних пакетів і критерії їх вибору при проведенні імітаційного моделювання.

Широкий вибір програмних продуктів дає змогу обрати для моделювання конкретної системи той засіб, який дозволить максимально наочно відтворити її властивості та надасть можливість аналізу та імітації функціонування моделі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Павленко П. М. Основи математичного моделювання систем і процесів : навч. посіб. / П. М. Павленко, С. Ф. Філоненко, О. М. Чередников, В. В. Трейтяк. – К. : НАУ, 2017. – 392 с.
2. Дубовой, В. М. Моделювання та оптимізація систем : підручник / В. М. Дубовой, Р. Н. Кветний, О. І. підручник / В. М. Дубовой, Р. Н. Кветний, О. І. Михальов, А. В. Усов. – Вінниця : ПП «ТД«Еднльвейс», 2017. С. 804
3. Методи представлення, збереження та аналізу даних інформаційних систем : колективна монографія / В. Ю. Щербань, С. М. Краснитський, Т. І. Астісова, В. М. Яхно. – Київ : ТОВ "Фастбінд Україна", 2023
4. Безвесільна О. М. Наукові дослідження в галузі автоматизації та приладобудування. Проектування комп'ютеризованих та інформаційно-вимірювальних моделювання систем : підручник, затверджений Вченою радою Державного університету "Житомирська політехніка" / О. М. Безвесільна, Ю. О. Подчашинський. – Київ : ДП «Редакція інформаційного бюлетеня «Офіційний вісник Президента України», 2021. – 896 с.
5. Пасічник В. В. Моделювання складних систем / В. В. Пасічник, Я. І. Виклюк, Р. М. Камінський. – Львів : Новий світ, 2021. – 404 с..
6. Рябенський В. М. Моделювання пристроїв обробки цифрових сигналів / В. М. Рябенський, Л. В. Солобутко. – К. : Кондор, 2021. – 352 с.
7. Павлиш В. А. Основи біотехнічних систем та їх моделювання / В. А. Павлиш, Л. К. Гліненко. – Львів : Львівська політехніка, 2020. – 380 с.
8. Ямпольський Л.С., Лавров О.А. Штучний інтелект у плануванні та управлінні виробництвом. // Л.С., Ямпольський, О.А Лавров– К.: Вища школа, 2015. – 254с.
9. Тимченко А.А. Основи системного проектування та системного аналізу складних об'єктів: Підручник для студентів вищих закладів освіти/За ред..В.І.Бикова – К.:Либідь, 2000. – 270с.

10. Стеценко І.В. Бойко О.В. Технологія імітаційного моделювання систем управління засобами сіток Петрі // Вісник Черкаського державного технологічного університету. – Черкаси, 2006.с.29-32
11. Зайцев Д.А. Математичні моделі дискретних систем: Навчальний посібник //Одеса: ОНАЗ ім. О.С. Попова, 2004.
12. Братушка С.М. Імітаційне моделювання як інструмент дослідження складних економічних систем. Вісник Української академії банківської справи, 2009. № 2(27), С. 113–118.
13. Жерновий Ю.В. Імітаційне моделювання систем масового обслуговування: практикум. Львів : ЛНУ імені Івана Франка, 2007. 307
14. Томашевський В.Н., Жданова О.Г., Жолдаков О.О. Вирішення практичних завдань методами комп'ютерного моделювання: Навч. посібник. К.: «Корнійчук», 2001. 268 с.
15. Лубко Д.В., Шаров С.В., Зінов'єва О.Г. Проєктування імітаційної моделі роботи технологічної лінії прибирання гною на тваринницькій молочній фермі. Вчені записки Таврійського національного університету ім.В.І. Вернадського. Серія: Технічні науки, 2022. Том 33(72) № 3. С. 56-60.
16. Соколовський Я. І., Шабатура Ю.В., Виклюк Я.І. та ін. Моделювання систем в середовищі GPSS World:навч. посіб. Львів : Новий Світ, 2020. 288 с.
17. Томашевський В. М. Моделювання систем: підруч. для студ. ВНЗ / за заг. ред. М. З. Згуровського. – К. : Видавнича група ВНУ, 2005. – 352 с.
18. Кветний Р. Н. Комп'ютерне моделювання систем та процесів. Методи обчислень : навч. Посібник / Р. Н. Кветний, І. В. Богач, О. Р. Бойко та ін. – Вінниця : ВНТУ, 2012. – 193 с.
19. Дубовой В. М. Ідентифікація та моделювання технологічних об'єктів і систем керування : навч. посібник / В. М. Дубовой. – Вінниця : ВНТУ, 2012 – 308 с.

20. Кузьмичов А. І. Ймовірнісне та статистичне моделювання в Excel для прийняття рішень : навч. посібник / А. І. Кузьмичов, Н. Г. Бишовець, Г. В. Куценко та ін. – К. : Ліра К, 2019. – 300 с.
21. Гліненко Л. К. Основи моделювання технічних систем: навч. посіб. для студ. ВНЗ / Л. К. Гліненко, О. Г. Сухоносів. – Львів : Бескид Біт, 2003. – 176 с.
22. Лебідь Р. Д. Математичні методи в моделюванні систем: навч. посіб. для студ. ЗВО / Р. Д. Лебідь, І. А. Жуков, М. М. Гузій. – К. : КМУЦА, 2000. – 158 с..
23. Фельдман Л. П. Чисельні методи в інформатиці / Л. П. Фельдман, А. І. Петренко, О. А. Дмитрієва. – К. : Видавнича група ВНУ, 2006. – 480 с.
24. Шматок С. О. Автоматизоване проектування систем керування на основі MATLAB : навч. посібник / С. О. Шматок, Ю. О. Подчашинський. – Житомир: ЖДТУ, 2005. – 172 с.
25. Бутко І. М. Системний підхід і моделювання в наукових дослідженнях / І. М. Бутко, М. Ю. Дітковська, М. П. Бутко. – К. : Центр учбової літератури, 2014. – 360 с.
26. Федосов Б. Т. Моделювання електромеханічних систем : навч. посібник / Б. Т. Федосов, С. Г. Чорний, В. П. Щокін. – К. : Кондор, 2018. – 204 с
27. Програмування і математичне моделювання: підручник для студ. вищих навч. закл. / І. О. Хвищун ; Львівський національний ун-т ім. Івана Франка. — Київ : Видавничий центр Львівського національного університету ім. Івана Франка, 2007. — 544 с
28. VisualPetri. [Електронний ресурс] – Режим доступу: <http://visual-petri.sourceforge.net/>.
29. Математичне моделювання в електроенергетиці: підручник / О. В. Кириленко, М. С. Сегеда, О. Ф. Буткевич, Т. А. Мазур ; за ред. М. С. Сегеди ; М-во освіти і науки, молоді та спорту України, Нац. ун-т «Львів. політехніка». — 2-ге вид. — Л. : Вид-во Львів. політехніки, 2013. — 608 с.

30. Онисик С. Б. Моделювання об'єктів керування. Поняття. Тлумачення. Моделі. Дослідження / С. Б. Онисик. – Львів : Львівська політехніка, 2019. – 300 с.
31. Томашевський В. М. Моделювання систем / В. М. Томашевський. – К. : Видавнича група BHV, 2005. – 352 с.
32. Системне моделювання в економіці [Електронний ресурс] – Режим доступу <http://cyb.univ.kiev.ua/library.school-guides.html/> Вергунова І.М. Системне моделювання в економіці. – 2018.
33. Жерновий Ю. В. Марковські моделі масового обслуговування// Ю. В. Жерновий, – Львів, 2004.
34. Заяць В. М. Методи, алгоритми та програмні засоби для моделювання і аналізу динаміки складних об'єктів і систем на основі дискретних моделей / В. М. Заяць. – Львів : Новий світ - 2000, 2020. – 400 с.
35. Методи, моделі та інформаційні технології оцінювання станів складних об'єктів: монографія / Є. І. Кучеренко, В. Є. Кучеренко, І. С. Глушенкова, І. С. Творошенко; ХНАМГ, ХНУРЕ. – Х. : ХНАМГ; ХНУРЕ, 2012. – 278 с
36. Стеценко І. В. Проектування графічного модуля програмного забезпечення Петрі-об'єктного моделювання систем / І. В. Стеценко, О. В. Василевська // Вісник Черкаського державного технологічного університету. – Черкаси : ЧДТУ, 2013. № 2. с. 13-18.
37. Dehouck R. Craft AI. The maturity of visual programming [Електронний ресурс]. Режим доступу до ресурсу – <http://www.craft.ai/blog/thematurity-of-visual-programming/> (дата звернення 20.11.2020 р)
38. Соколовський Я. І., Шабатура Ю.В., Виклюк Я.І. та ін. Моделювання систем в середовищі GPSS World: навч. посіб. Львів : Новий Світ, 2020. 288 с.
39. Зінов'єва О. Г. Огляд програмних засобів імітаційного моделювання // Таврійський державний агротехнологічний університет імені Дмитра Моторного, Інформаційні технології - Вісник ХНТУ № 3(82), 2022 р.

