

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки

Кафедра Інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка імітаційної моделі функціонування мережі зв'язку»

Виконав студент 2 курсу групи МІС- 19
спеціальності 122 Комп'ютерні науки

Талеба Ільяса Маджидовича


Керівник к.геог.н., доц.
Коваленко Людмила Борисівна

Рецензент к.т.н., доц.
Гнатовська Ганна Арнольдівна

Одеса 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
 Глава 1 ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ
 Факультет Магістерської підготовки
 Кафедра інформаційних технологій
 Рівень вищої освіти магістр
 напрямок 122 Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри 

“ 26 ” жовтня 2020 р.

З А В Д А Н Н Я НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

_____ Талебу Ільясу Маджидовичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи «Розробка імітаційної моделі функціонування мережі зв'язку»

керівник роботи к.географ.н., доцент Коваленко Людмила Борисівна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 16 ” жовтня 2020 р. №194-С

2. Строк подання студентом проекту 7 грудня 2020 р.

3. Вихідні дані до роботи вихідні дані до моделювання: час відправлення повідомлення абонентом, довжина повідомлення в байтах, час обробки повідомлення обчислювальним комплексом, час передачі повідомлення по каналу зв'язку, кількість абонентів, середній час відправлення повідомлень, ймовірності розподілу видів повідомлень різних категорій, продуктивність обчислювального комплексу, середовище моделювання AnyLogic.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Методологічні основи імітаційного моделювання

2. Моделювання мережі передачі даних

3. Огляд системи імітаційного моделювання AnyLogic

4. Постановка задачі

5. Побудова моделі функціонування мережі зв'язку

6. Аналіз результатів моделювання

5. Перелік графічного матеріалу

1. Загальна схема роботи маршрутизатора

2. Алгоритм роботи блоку «Абонент»

3. Алгоритми роботи блоків маршрутизатора

4. Схеми подієвих частин для блоків моделі: абонент, маршрутизатор канал зв'язку

6. Консультанти розділів проекту

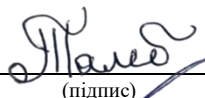
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

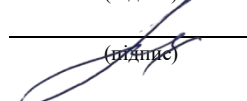
7. Дата видачі завдання “ 26 ” жовтня 2020 р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів дипломного проекту	Термін виконання етапів проекту	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Дослідження літературних джерел за темою кваліфікаційної роботи	26.10	80	добре
2	Дослідження методологічних основ імітаційного моделювання складних систем	02.11	80	добре
3	Аналіз функціональних можливостей середовища моделювання AnyLogic.	08.11	80	добре
4	Постановка завдання до моделювання	12.11	80	добре
5	Опис параметрів моделі	15.11	80	добре
6	Рубіжна атестація	19.11	80	добре
7	Побудова подієвої частини моделі	25.11	80	добре
8	Налагодження і тестування моделі	29.11	80	добре
9	Аналіз результатів моделювання. Формулювання висновків до роботи	02.12	80	добре
	Подання роботи на кафедру	07.12		
	Перевірка на плагіат	08.12		
	Рецензування	16.12		
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)		80 (В)	добре

Студент

Керівник проекту


(підпис)


(підпис)

Талєб І.М.

(прізвище та ініціали)

Коваленко Л.Б.

(прізвище та ініціали)

АНОТАЦІЯ

на магістерську роботу «Розробка імітаційної моделі функціонування мережі зв'язку»,

студента Талеба Ільяса Маджидовича

Актуальність роботи полягає в необхідності визначення завантаження каналів при різних параметрах мережі для забезпечення необхідної якості обслуговування.

Мета роботи – розробка імітаційної моделі функціонування мережі зв'язку, на підставі якої можна дослідити вплив різних вихідних параметрів на показники функціонування мережі для їх оцінки та прийняття рішень щодо поліпшення якості обслуговування мережі.

Задачі дослідження: обґрунтувати вибір пакету імітаційного моделювання; розробити моделі типових об'єктів мережі зв'язку; розробити імітаційну модель функціонування мережі зв'язку; дослідити вплив вихідних параметрів на показники функціонування мережі для оцінки та прийняття рішень щодо поліпшення якості обслуговування мережі.

Об'єкт дослідження – модель мережі зв'язку.

Предмет дослідження – методика проведення моделювання процесу функціонування мережі зв'язку.

Методи дослідження: методи імітаційного моделювання, об'єктно-орієнтоване програмування.

Результати, їх новизна, теоретичне та практичне значення: результати дослідження можуть бути корисні для постачальників послуг зв'язку, і використані для фінансових розрахунків вартості цих послуг.

Структура магістерської роботи складається з вступу, чотирьох розділів, висновків, переліку посилань на 16 найменувань, додатків. Повний обсяг роботи становить 81 сторінка, містить 20 рисунків і 9 таблиць.

КЛЮЧОВІ СЛОВА: модель, імітаційне моделювання, агент, мережа зв'язку, якість обслуговування.

SUMMARY

for the master's work "Development of a Simulation Model for the
Functioning of a Communication Network",

student Taleb Ilyas

The relevance of the work lies in the need to determine the channel load for various network parameters to ensure the required quality of service.

The aim of the work is to develop a simulation model of the communication network functioning, on the basis of which it is possible to study the influence of various initial parameters on the performance indicators of the network for their assessment and decision-making to improve the quality of network service.

Research objectives: to substantiate the choice of the simulation package; develop models of typical communication network objects; develop a simulation model of the communication network functioning; to investigate the influence of the initial parameters on the performance of the network to assess and make decisions to improve the quality of network service.

The object of research is a communication network model.

The subject of the research is a technique for modeling the process of functioning of a communication network.

Research methods: methods of simulation, object-oriented programming.

Results, their novelty, theoretical and practical significance: the research results can be useful for communication service providers, and used for financial calculations of the cost of these services.

The structure of the master's work consists of an introduction, four chapters, a conclusion, a list of references to 16 titles, applications. The total volume of work is 81 pages, contains 20 figures and 9 tables.

KEYWORDS: model, simulation, agent, communication network, QoS

ЗМІСТ

СПИСОК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	8
ВСТУП.....	9
1 МЕТОДОЛОГІЧНІ ОСНОВИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ	11
1.1 Види імітаційного моделювання.....	12
1.2 Класифікація систем комп'ютерного моделювання.....	16
1.3 Проектування і розробка імітаційних моделей складних об'єктів	19
2 МОДЕЛЮВАННЯ МЕРЕЖ ПЕРЕДАЧІ ДАНИХ.....	22
3 ОГЛЯД СИСТЕМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ANYLOGIC....	27
3.1 Загальні відомості про систему моделювання AnyLogic.....	27
3.2 Етапи імітаційного моделювання в AnyLogic	29
3.3 Основні концепції, реалізовані в AnyLogic	31
3.4 Об'єкти Enterprise Library	36
4 ПОБУДОВА МОДЕЛІ ФУНКЦІОНУВАННЯ МЕРЕЖІ ЗВ'ЯЗКУ	41
4.1 Постановка задачі	41
4.2 Опис параметрів моделі.....	42
4.3 Моделювання абонента мережі.....	46
4.3.1 Вихідні дані.....	46
4.3.2 Перегляд результатів моделювання	46
4.3.3 Побудова подієвої частини об'єкту «Абонент».....	50
4.4 Моделювання маршрутизатора	52
4.4.1 Вихідні дані.....	52
4.4.2 Побудова подієвої частини об'єкту «Маршрутизатор»	54
4.5 Моделювання каналу зв'язку	59
4.5.1 Вихідні дані.....	59
4.5.2 Побудова подієвої частини об'єкту «Канал»	60
4.6 Запуск і налагодження моделі	61
4.7 Аналіз результатів моделювання	66

ВИСНОВКИ	70
ПЕРЕЛІК ПОСИЛАНЬ	71
Додаток А Загальна схема роботи маршрутизатора	74
Додаток Б Алгоритм блоку Абонент	75
Додаток В Алгоритми роботи блоків маршрутизатора	76
Додаток Г Програмний код класу Message.....	80

СПИСОК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Скорочення

ІЗІМ	– Інструментальні засоби імітаційного моделювання
ІМ	– Імітаційна модель
ІМСС	– Імітаційне моделювання складних систем
ОЗП	– Оперативний запам'ятовуючий пристрій
ОК	– Обчислювальний комплекс
ОККП	– Обчислювальний комплекс комутації повідомлень
ОКП	– Одноканальний пристрій
ПЗ	– Програмне забезпечення
СМО	– Система масового обслуговування
GPSS	– General Purpose Simulation System (Загальноцільові системи моделювання)
QoS	– Quality of Service (Якість обслуговування)
HDD	– Hard Disk Drive (Жорсткий диск)
HTTP	– HyperText Transfer Protocol (Протокол передачі гіпертексту)
FIFO	– First In, First Out
IEEE	– Institute of Electrical and Electronic Engineers (Інститут інженерів з електротехніки та електроніки)
LAN	– Local Area Network (Локальна мережа)

Умовні позначення

оп/с – операцій в секунду

Мб – мегабайт

ВСТУП

На сьогоднішній день забезпечення необхідної якості обслуговування для різних видів трафіку є одним з головних завдань при створенні будь-якої мережі зв'язку. Тому особливо важливе значення набуває дослідження поведінки трафіку різних класів з метою визначення завантаження каналів при заданих параметрах мережі. Подібні дослідження актуальні для постачальників послуг зв'язку, і можуть допомогти під час фінансових розрахунків вартості цих послуг.

Якість обслуговування характеризують такі показники:

- ймовірність (коефіцієнт) пропускнуої здатності мережі;
- середній час передачі (затримки) повідомлень (пакетів);
- варіація затримки;
- ймовірність втрати повідомлень (пакетів).

Для пошуку «вузьких» ділянок у мережі потрібен її детальний аналіз, який небажано проводити на працюючій системі зв'язку, тому що будь-який зовнішній вплив підвищує вірогідність відмови, що призводить до фінансових втрат підприємства. Заміна натурних експериментів можлива за допомогою засобів імітаційного моделювання, коли робота об'єкту відтворюється за допомогою програмного пакету. В результаті за декілька хвилин можна відтворити роботу мережі протягом декілька днів, що дає можливість оцінити роботу мережі у широкому діапазоні варійованих параметрів.

Метою магістерської роботи є розробка імітаційної моделі функціонування мережі зв'язку, на підставі якої можна дослідити вплив різних вихідних параметрів на показники функціонування мережі для їх оцінки та прийняття рішень щодо поліпшення якості обслуговування мережі.

Для досягнення поставленої мети в роботі необхідно вирішити наступні завдання:

- провести огляд методологічних основ імітаційного моделювання;

- провести порівняльний аналіз сучасних пакетів імітаційного моделювання;
- обґрунтувати вибір пакету імітаційного моделювання для побудови моделі функціонування мережі зв'язку;
- розробити моделі типових об'єктів мережі зв'язку: абонент, канал зв'язку, маршрутизатор;
- на підставі розроблених моделей типових об'єктів мережі розробити імітаційну модель функціонування мережі зв'язку;
- провести запуск і налагодження моделі;
- дослідити вплив ємностей буферів, інтервалів часу надходження повідомлень, їх обчислювальних труднощів і інших параметрів на показники функціонування мережі з метою їх оцінки та прийняття рішень при необхідності щодо поліпшення якості обслуговування мережі.

1 МЕТОДОЛОГІЧНІ ОСНОВИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

Математичне моделювання поділяється на аналітичне та імітаційне (рис.1.1).

Під аналітичним моделюванням розуміють процес формалізації реального об'єкта і знаходження його рішення в аналітичних функціях.

Процедуру побудови математичної моделі будь-якого реального явища або процесу і знаходження чисельного рішення за допомогою ітераційних формул часто називають чисельним моделюванням [1, 2]¹⁾.

Комп'ютерне моделювання – це застосування комп'ютерних технологій розв'язування математичних моделей на електронно-обчислювальних машинах.

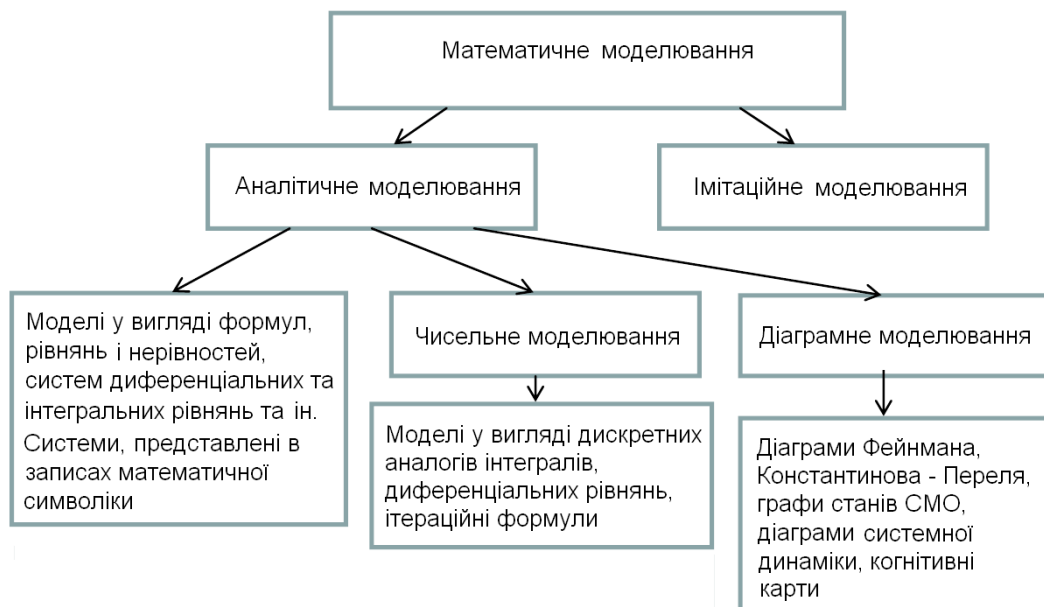


Рисунок 1.1 – Види математичного моделювання

¹⁾ [1] Боголюбов А.Н. Основы математического моделирования. Учеб. пособие. М.: МГУ, 2003. 137с.

[2] Введение в математическое моделирование. Учеб. пособие. Логос, 2004. 440с.

Імітаційне моделювання застосовується, коли неможливо побудувати аналітичну модель системи, яка враховує причинні зв'язки, нелінійності, стохастичні змінні, коли необхідно імітувати поведінку системи в часі, розглядаючи різні можливі сценарії її розвитку при зміні зовнішніх і внутрішніх умов.

Таким чином, імітаційне моделювання – це високорівнева інформаційна технологія із застосуванням комп'ютерів і найчастіше використовується при моделюванні складних систем [3]¹⁾.

1.1 Види імітаційного моделювання

Імітаційне моделювання умовно може бути представлено різними різновидами або напрямками, що відповідно мають свої методології (рис. 1.2).

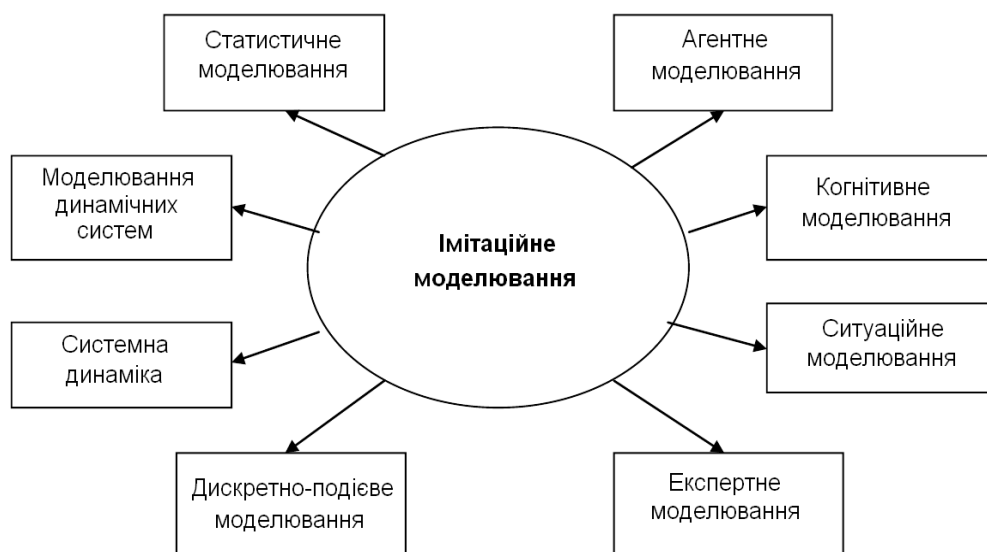


Рисунок 1.2 – Напрямки імітаційного моделювання

¹⁾ [3] Аверченков В.И., Федоров В.П., Хейфец М.Л. Основы математического моделирования технических систем. Учеб. пособие. М: Изд-во «Флинта», 2011. 271с.

Статистичне (чисельне) моделювання є різновидом імітаційного моделювання [4]¹⁾. Моделі тут будуються для явищ і систем об'єктів, входи і (або) функціональні співвідношення між різними компонентами якої містять елементи випадковості або повністю випадкових процесів, що підкоряються імовірнісним законам.

Статистичне моделювання – застосування математичної статистики для статистичного оцінювання і прогнозування, кореляційно-регресійного і багатовимірною статистичного аналізу, оптимізації систем, визначення екстремуму функцій великого числа змінних в різних галузях виробництва і науки. Результатом застосування статистичного моделювання є можливість проведення прогнозу. Прогнозування можна вважати одним з найбільш цінних додатків імітаційного моделювання.

Динамічні системи. Під динамічною системою розуміють будь-який об'єкт, процес або явище, для якого однозначно визначено поняття стану як сукупності деяких величин і заданий закон, який описує зміну початкового стану з плином часу, що рухається в просторі і змінюється в часі. Динамічними об'єктами можуть бути механічні, виробничі, фізичні, хімічні, біологічні об'єкти, обчислювальні процеси і ін.

Динамічні системи описуються різними способами: диференціальними рівняннями, дискретними відображеннями, марковськими ланцюгами, графічними образами і ін. Вони класифікуються залежно від виду оператора відображення та структури фазового простору. Розрізняють лінійні і нелінійні, безперервні і дискретні оператори відповідно визначаються системи лінійні і нелінійні, системи з дискретним часом і системи з безперервним часом.

Для моделювання динамічних систем використовуються середовища схемотехнічного моделювання: VISSIM, SIMULINK + MATLAB, PowerSim, Multisim, LabView, Easy5, MvStudium і ін.

¹⁾ [4] Бусленко, Н.П. Метод статистического моделирования. М.: Статистика, 1970. 112 с.

Системна динаміка – парадигма моделювання, де для досліджуваної системи 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 [5]¹⁾. За допомогою системної динаміки будують моделі суспільства, світової динаміки, бізнес-процесів, розвитку міста, моделі виробництва, динаміки популяції, екології та розвитку епідемії та інші [6]²⁾.

Агентне моделювання (agent-based model (ABM)) – різновид імітаційного моделювання, сучасний метод, що дозволяє досліджувати роботу децентралізованих агентів і то, як така поведінка визначає поведінку всієї системи в цілому [7]³⁾.

На відміну від системної динаміки аналітик визначає поведінку агентів на індивідуальному рівні, а глобальне поведінка виникає як результат діяльності безлічі агентів (моделювання «знизу вгору»).

Агент – це сутність, що володіє активністю, автономною поведінкою, може приймати рішення відповідно до деяких наборів правил, взаємодіяти з оточенням, а також самостійно змінюватися. Агентами можуть бути: люди, транспорт, обладнання, нематеріальні речі, організації.

Середовище – деякий простір, в якому знаходяться агенти, яке характеризується своїми станами і факторами, агенти знаходяться в певному місці цього простору, з можливістю орієнтування і пересування в даному просторі.

Правила взаємодії – закони взаємодії агентів у довкіллі, з процедурами прийняття рішення і вибору стратегії при черговому кроці взаємодії.

Агентний підхід дозволяє досліджувати завдання колективної взаємодії, ефективно вирішувати завдання прогнозування. Агентні системи

¹⁾ [5] Сидоренко, В.Н. Системная динамика. М.: Эконом. факультет МГУ, ТЕИС, 1998. 200 с.

²⁾ [6] AnyLogic. Навчальний посібник з системної динаміки URL: <http://www.xjtek.com> (Дата звернення 12.11.2020)

³⁾ [7] Ивашкин, Ю.А. Агентные технологии и мультиагентное моделирование систем. Учеб. пособие. М.: МФТИ, 2013. 268с.

дозволяють досліджувати процеси самоорганізації, дають можливість природного опису складних систем, мають високу гнучкість.

Під *когнітивними технологіями* розуміється широкий спектр технологій раціоналізації і формалізації інтелектуальних систем створення і функціонування знань, експертизи, комунікації і прийняття рішення [8]¹⁾. Когнітивні інформаційні технології являє собою сукупність методів, прийомів, дій, процесів, здійснюваних у певній послідовності, інструментальних засобів, що дозволяють перетворити вхідну інформацію в варіанти управлінського рішення.

В даний час когнітивний підхід є напрямком дослідження великих систем, до яких відноситься соціально-економічні, політичні, екологічні системи. Методологія когнітивного моделювання розвивається в напрямку вдосконалення апарату аналізу, моделювання і пошуку рішень в слабоформалізуємих і погано структурованих ситуаціях при відсутності або неповної інформації про процеси, що відбуваються в таких ситуаціях і умовах швидких змін.

Ситуаційне управління – це метод управління складними технічними та організаційними системами, заснований на ідеях теорії штучного інтелекту: уявлення знань про об'єкт управління і способи управління ним на рівні логіко-лінгвістичних моделей, використання навчання та узагальнення в якості основних процедур при побудові процедур управління за поточними ситуаціями, використання дедуктивних систем для побудови багатокрокових рішень [9]²⁾.

Одним з напрямків ситуаційного моделювання є розробка комп'ютерних та ігрових імітаційних моделей своєрідних тренажерів, що дозволяють проводити навчання по виробничим, транспортним, психологічним, педагогічним, управлінським та ін. ситуаціям. В основі даних

¹⁾ [8] Абрамова, Н.А., Авдеева З.К. Когнитивный анализ и управление развитием ситуаций: проблемы методологии, теории и практики. Проблемы управления. 2008. № 3. С. 85-87

²⁾ [9] Болотова Л.С. Системы искусственного интеллекта: модели и технологии, основанные на знаниях. М.: Финансы и статистика, 2012. 664 с.

тренажерів лежить імітаційна модель, яка дозволяє учням застосовувати певний набір інструментів впливу і управління та відтворює реакцію соціальної системи на відповідне керуючий вплив.

1.2 Класифікація систем комп'ютерного моделювання

Імітаційні моделі можуть створюватися у вигляді програм технологією прямого програмування, а також за допомогою систем комп'ютерного моделювання, представлених на рис. 1.3.

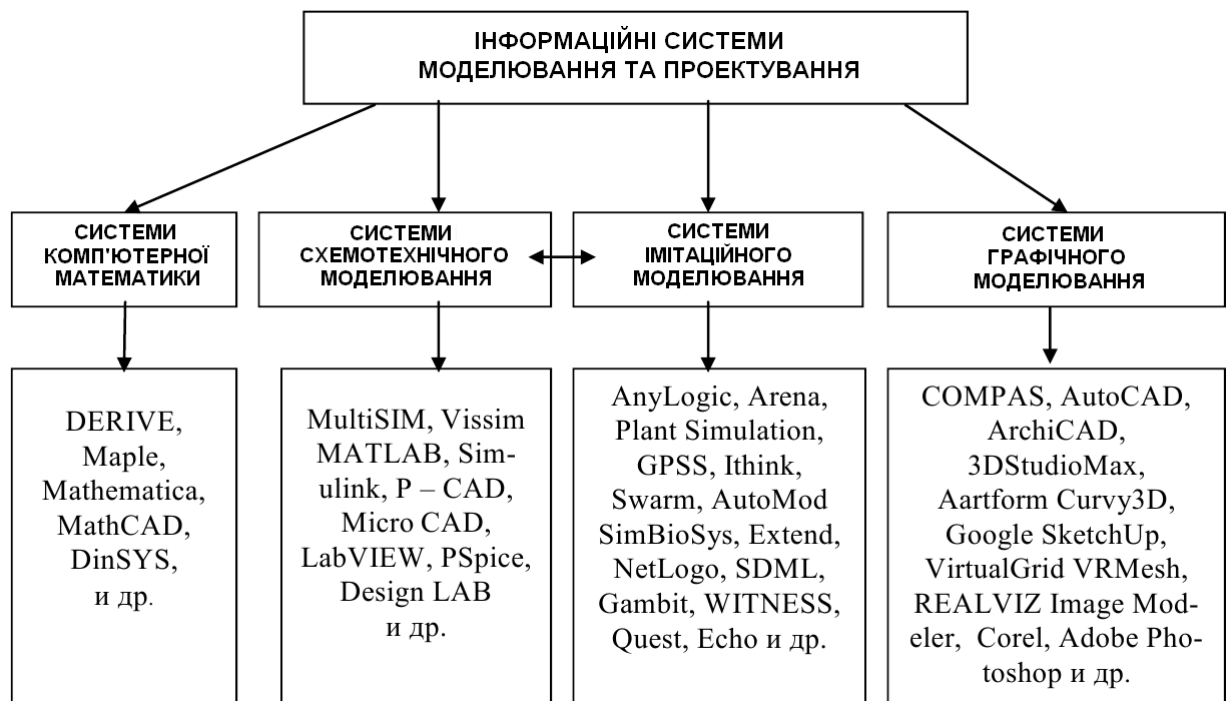


Рисунок 1.3 – Умовна класифікація інформаційних систем за типом вирішуваних завдань

В даний час йде стрімкий розвиток напрямку розробки інструментальних засобів імітаційного моделювання (ІЗІМ) цілеспрямовано підтримують ті чи інші методології та напрямки імітаційного моделювання складних систем (ІМСС):

– AnyLogic – програмне забезпечення для імітаційного моделювання складних систем і процесів, що дозволяє підтримувати напрямок агентного моделювання, дискретно-подієвого моделювання та розробки моделей системної динаміки (розробляється російською компанією XJ Technologies);

– GPSS (англ. General Purpose Simulation System – общецельова система моделювання) – мови об'єктно-орієнтованого програмування, що використовується для імітаційного моделювання систем масового обслуговування, різних інформаційних процесів і розробки імітаційних моделей в мережі інтернет;

– Arena – розробляється компанією Systems Modeling Corporation, програмне забезпечення для імітаційного моделювання, що дозволяє створювати рухомі комп'ютерні моделі, використовуючи які можна адекватно представити дуже багато реальних систем;

– Plant Simulation – програмне середовище імітаційного моделювання систем і процесів, призначеного для оптимізації матеріалопотоків, завантаження ресурсів, логістики та методу управління для всіх рівнів планування від цілого виробництва та мережі виробництв до окремих ліній і дільниць;

– SimBioSys: C++ – оболонки агентно-базового еволюційного моделювання в біологічних і суспільних науках;

– системи моделювання SWARM та його розширення MAML (Multi-Agent Modelling Language) для моделювання штучного світу;

– пакети Ascape (Agent Landscape) і RePast (Recursive Porous Agent Simulation Toolkit), написані на платформі мови Java, для підтримки агентно-базового моделювання;

– NetLogo і MIMOSE (Micro- and Multilevel Modelling Software) інформаційні системи, призначені для створення імітаційних моделей і технологій моделювання в суспільних науках;

– SPSS, Statistica, PilGrim, Z-Tree – систем статистичного моделювання для дослідження економічних, педагогічних і психологічних явищ і процесів.

У галузі імітаційного моделювання реальних об'єктів умовно виділяються чотири основних напрямки: моделювання динамічних систем, дискретно-подієве моделювання, агентне моделювання і системна динаміка. У табл. 1.2 наведені мови і засоби автоматизації імітаційного моделювання, які однозначно, а деякі умовно можна віднести до відповідних підходів (методологій) імітаційного моделювання.

Як видно з табл. 1.2 систем імітаційного моделювання досить багато, однак, не всі перераховані програмні продукти доступні для використання в Україні. При виконанні магістерської роботи цей факт будемо враховувати при виборі пакету імітаційного моделювання. Ще одним з важких факторів є вартість такого пакету.

Таблиця 1.2 – Інструментальні засоби імітаційного моделювання

Напрями імітаційного моделювання	Інструментальні засоби
Динамічні системи	Dynamo, Power-Sim, MIMIC, АРТОН MIDAS, PACTOLUS, CSSL, Слами, GASP, НЕДИС, МІКС, MATLAB+ Simulink, Multisim VisSim, LabView, Easy5, MvStudium та ін.
Системна динаміка	AnyLogic, Arena, SimBioSys, eM-Plant, Tecnomatix, Plant Simulation, SimuLab, VenSim, PowerSim, Pilgrim, Dynamo, Stella, Ithink та ін.
Дискретно-подієве моделювання	AnyLogic, Arena, Extend, PowerSim Studio, Witness, ProModel, Pilgrim, Taylor Simulation, GPSS, SimScript, Quest, SIMULA, SIMUL8, Modelling, SimProcess, AutoMod, Enterprise Dynamics, FlexSim та ін.
Агентне моделювання	AnyLogic, Swarm+MAML, SimAgent, SimBioSys, C++, Java, AgentSpeak, Oz, TeleScript, RePast, NetLogo, Ascape, Mason і ін.

Системна динаміка і дискретно-подієве моделювання – традиційні підходи, агентне моделювання – відносно новий підхід. Підхід динамічного моделювання дозволяє побачити поведінку моделі в часі при русі в минуле (для отримання історичного результату) і в майбутнє (для виявлення можливих результатів). При зміні параметрів моделі можна спостерігати причини успіху або невдачі, знаходити оптимальні рішення. Математично, системна динаміка і динамічні системи оперують в основному з безперервними в часі процесами, тоді як дискретно-подієве і агентне моделювання – в основному з дискретними.

Для розробки моделей складних систем за даними підходам використовуються середовища імітаційного моделювання, розроблені в Росії: AnyLogic, Pilgrim, Rand Model Designer, розширений редактор GPSS та ін.

1.3 Проектування і 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 наступним чином [10]¹⁾:

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

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

2) Стадія проектування та розробки програмного забезпечення:

- етап розробки ескізного і технічного проекту автоматизованої системи;
- етап проектування програмного забезпечення;
- етап проектування інтерфейсу;

¹⁾ [10] Брауде, Э. Технология разработки программного обеспечения. СПб.: Питер, 2004. 655 с.

– етап реалізації програмного забезпечення (створення програмного коду);

– етап створення та оформлення документації.

3) Стадія впровадження.

4) Період супроводу або користувальницький період.

Практично будь-яка сучасна велика програмна система розробляється із застосуванням CASE-технологій принаймні на етапах аналізу і моделювання, що пов'язано з великою складністю даної проблематики і з прагненням підвищити ефективність робіт [10]¹⁾.

Системи проектування баз даних забезпечують логічне моделювання даних, автоматичне перетворення моделей даних в третю нормальну форму, автоматичну генерацію схем бази даних та описів форматів файлів на рівні програмного коду.

При створенні моделі вихідної інформаційної системи зазвичай застосовується функціональна методологія [11]²⁾. Вона передбачає розгляд системи у вигляді набору функцій, що перетворюють вхідний потік інформації у вихідний.

Функціональна модель – опис системи за допомогою IDEF0. Дана модель призначена для опису існуючих бізнес-процесів, в якій використовуються як природні, так і графічні мови. Для передачі інформації про конкретну систему джерелом графічної мови є сама методологія IDEF0.

Методологія IDEF0 призначена для побудови ієрархічної системи діаграм – одиничних описів фрагментів системи. Спочатку проводиться опис системи в цілому та її взаємодії з навколишнім світом (контекстна діаграма), після чого проводиться функціональна декомпозиція – система розбивається на підсистеми і кожна підсистема описується окремо (діаграми

¹⁾ [10] Брауде, Э. Технология разработки программного обеспечения. СПб.: Питер, 2004. 655 с.

²⁾ [11] Черемных, С.В., Семенов И. О., Ручкин В.С. Структурный анализ систем: IDEF – технологии. М.: Финансы и статистика, 2003. 208 с.

декомпозиції). Потім кожна підсистема розбивається на більш дрібні і так далі до досягнення потрібного ступеня деталізації.

Для проведення аналізу та реорганізації складних систем і процесів Logic Works пропонує CASE-засіб верхнього рівня – VPwin, який підтримує 3 методології:

- IDEF0 (функціональна модель),
- IDEF3 (WorkFlow Diagram) – тільки діаграми процесів,
- DFD (DataFlow Diagram) – діаграми потоків даних.

ERwin. Підтримує декілька різновидів методології інформаційного моделювання, заснованої на ER-діаграмах (сутність - зв'язок). Інтеграція моделей VPwin з моделями ERwin виконується шляхом обміну даними через функції експорту/імпорту.

Після детального розгляду загальних методологічних основ імітаційного моделювання доцільно дослідити особливості моделювання мереж передачі даних.

2 МОДЕЛЮВАННЯ МЕРЕЖ ПЕРЕДАЧІ ДАНИХ

На сьогоднішній день забезпечення необхідної якості обслуговування для різних видів трафіку є одним з головних завдань при створенні будь-якої мережі зв'язку. Тому особливо важливе значення набуває дослідження поведінки трафіку різних класів з метою визначення завантаження каналів при заданих параметрах мережі.

Якість обслуговування характеризують такі показники:

- ймовірність (коефіцієнт) пропускної здатності мережі;
- середній час передачі (затримки) повідомлень (пакетів);
- варіація затримки;
- ймовірність втрати повідомлень (пакетів).

Методи забезпечення якості обслуговування (QoS) спрямовані на поліпшення характеристик продуктивності і надійності мережі, ці методи дозволяють зменшити затримки, варіації затримок, а також втрати пакетів в періоди перевантаження мережі, створюючи тим самим необхідні умови для задовільного обслуговування мережею трафіку додатків. Методи забезпечення якості обслуговування спрямовані на компенсацію негативних наслідків тимчасових перевантажень, що виникають в мережах з комутацією пакетів. У цих методах використовуються різні алгоритми управління чергами, резервування і зворотного зв'язку, що дозволяють знизити негативний вплив черг до прийняттого для користувачів рівня [12]¹⁾.

Забезпечення необхідної якості обслуговування є важливим питанням при розрахунках бізнес-процесів надання послуг зв'язку провайдерами. Треба вміти передбачати параметри, що характеризують функціонування основних і резервних каналів зв'язку, їх надійності і ймовірності виходу з ладу.

¹⁾ [12] Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. 4–е изд. СПб.: Питер, 2010. 944 с.: ил.

Для створення моделі мережі (в тому числі завдання її топології, характеристик елементів мережі), а також динамічного моделювання її роботи, аналізу та оптимізації її характеристик, управління трафіком, безсумнівно, потрібно використовувати один з найпотужніших інструментів дослідження складних систем – імітаційне моделювання (ІМ).

В даний час відомий ряд спеціалізованих програмних продуктів (COMNETIII, BONE S DESIGNER, OPNET Modeler, пакети Model Vision Studuim, ns3, AnyLogic, ИСМА, середовище MATLAB сумісно з пакетом Simulink, CPN (Colored Petri Nets і ін.), призначених безпосередньо для моделювання мереж передачі даних. Однак з різних причин деякі з них не знайшли досить широкого поширення в нашій країні. У тому числі через високу вартість, яка не виправдовує економічно рішення задачі оцінки якості обслуговування мережі.

Для пошуку «вузьких» ділянок у мережі потрібен її детальний аналіз, який небажано проводити на працюючій системі зв'язку, тому що будь-який зовнішній вплив підвищує вірогідність відмови, що призводить до фінансових втрат підприємства. Заміна натурних експериментів можлива за допомогою засобів імітаційного моделювання, коли робота об'єкту відтворюється за допомогою програмного пакету. В результаті за декілька хвилин можна відтворити роботу мережі протягом декілька днів, що дає можливість оцінити роботу мережі у широкому діапазоні варійованих параметрів.

Система імітаційного моделювання повинна вирішувати наступні основні завдання:

- 1) Підвищення рівня автоматизації виробництва, націлене на підвищення продуктивності.
- 2) Оцінка і аналіз роботи мережі у широкому діапазоні варійованих параметрів за декілька хвилин, без виведення з експлуатації діючого обладнання.

3) Можливість визначення мінімально необхідного обладнання, що забезпечує потреби передачі, обробки та збереження інформації (якщо воно і не має реальних аналогів у наш час).

4) Оцінка необхідного запасу продуктивності обладнання, що забезпечує можливе збільшення виробничих потреб у найближчий час (один – два роки).

5) Вибір декількох варіантів обладнання з урахуванням поточних потреб, перспективи розвитку на підставі критерію вартості обладнання.

б) Проведення перевірки працездатності обчислювальної системи, складеної з рекомендованого обладнання.

Порівняльний аналіз відомих пакетів імітаційного моделювання, які спеціалізуються безпосередньо на моделюванні мереж зв'язку представлений у табл. 2.1. Як можна побачити більшість пакетів мають високу вартість і потребують знання специфічних мов програмування. Так пакет CPN хоча і розповсюджується безкоштовно, без будь яких обмежень на право використання, модифікації і розповсюдження третіми лицами, але потребує знання вбудованої мови програмування CPN ML для моделювання розмальованих мереж Петрі. Приклад мережі Петрі представлений на рис. 2.1.

Враховуючі перелічені аргументи було вирішено для моделювання мережі зв'язку використовувати сучасний пакет імітаційного моделювання, який би мав можливість безкоштовного використання і побудови моделі з використанням відомих мов програмування високого рівня. Гарним рішенням є використання пакету для ІМ AnyLogic. У бібліотеках AnyLogic немає об'єктів, призначених безпосередньо для побудови ІМ мереж зв'язку. Але є об'єкти для дискретно-подієвих процесів, з яких можна створити такі об'єкти [13]¹⁾.

¹⁾ [13] Офіційний сайт XJ Technologies URL: <http://www.xjtek.com> (Дата звернення 12.11.2020)

При побудові моделі використовуються засоби візуальної розробки (введення станів і переходів стейтчарта, введення піктограм змінних і т.п.), завдання чисельних значень параметрів, аналітичних записів співвідношень змінних і аналітичних записів умов настання подій. Основною технологією програмування в AnyLogic є візуальне програмування.

Таблиця 2.1 – Порівняння ПЗ моделювання мереж зв'язку

Продуктивність і назва	Ціна, \$	Вимоги до пам'яті комп'ютера	Операційні системи	Прикмітка
Caci Products Co. COMNET III	35.000	від 32 Мб ОЗП від 100 Мб HDD	Windows, SunOS, Solaris	LANs, X.25, ATM, Frame Relay, протоколи маршрутизації IP. Реалізація власного коду SIMSCRIPT. Анімація.
Cadence Lie. BONE S DESIGNER	20.000	від 32 Мб ОЗП від 80 Мб HDD	SunOS, Solaris, HP-UX	LANs, X.25, ATM, Frame Relay. Реалізація коду на C++. Анімація.
MIL3 Inc. OPNET Modeler	40.000	від 16 Мб ОЗП від 150 Мб HDD	Windows, Solaris, HP-UX	Fixed/wireless LANs, X.25, ATM, Frame Relay, Web caching, http, усі різновиди TSP. Реалізація коду на C++. Анімація. Вихідний код частково відкритий.
CPN	–	від 8 Мб ОЗП від 250 Мб HDD	Windows, SunOS, Solaris, Linux, FreeBSD HP-UX	Fixed/wireless LANs, X.25, ATM, Frame Relay, Web caching, http, усі різновиди TSP. Реалізація власного коду на CPN ML. Анімація. Вихідний код повністю відкритий.

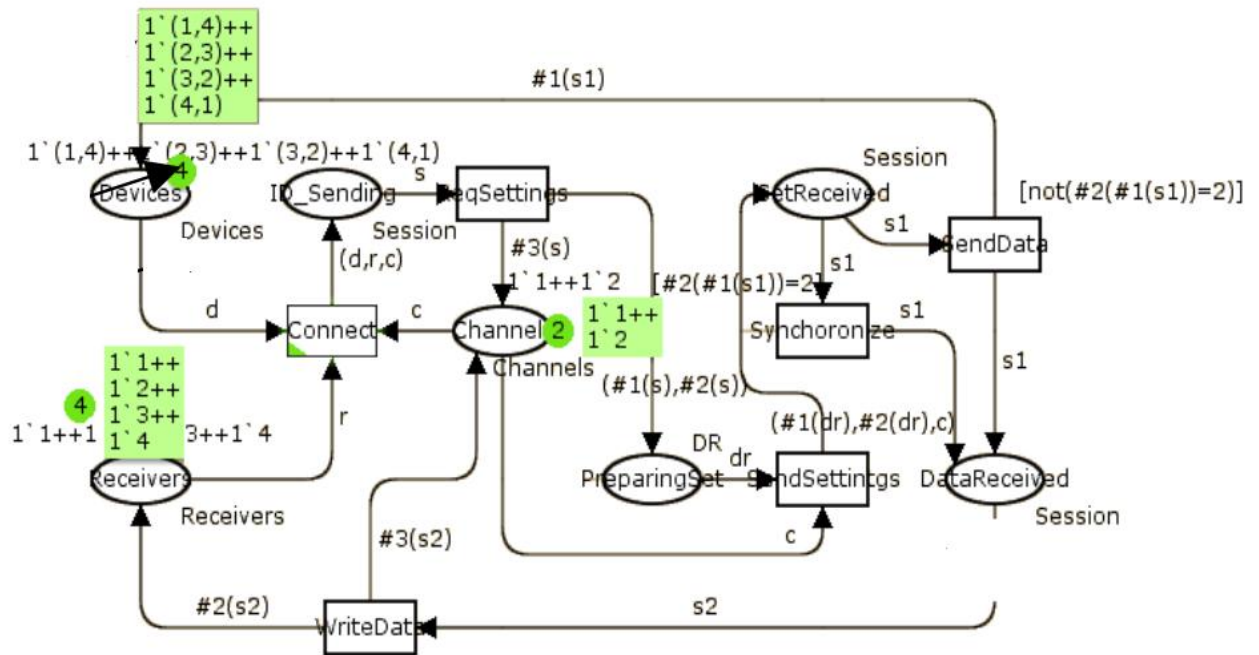


Рисунок 2.1 – Приклад мережі Петрі

AnyLogic створений у середовищі Eclipse і є надбудовою над мовою Java – однієї з найпотужніших і в той же час найпростіших сучасних об'єктно-орієнтованих мов. Всі об'єкти, визначені користувачем при розробці моделі за допомогою графічного редактора, компілюються у конструкції мови Java, а потім відбувається компіляція всієї зібраної програми на Java, яка задає модель, в виконуваний код. Хоча програмування зведено до мінімуму, розробнику моделі необхідно мати навички програмування цією мовою [14]¹⁾.

У наступних главах магістерської роботи наводяться основні характеристики пакету імітаційного моделювання AnyLogic, а також приклади створення окремих класів типових об'єктів мереж зв'язку.

¹⁾ [14] Java For AnyLogic Users URL: http://www.xjtek.com/files/book/Java_for_AnyLogic_users.pdf (Дата звернення 12.11.2020)

3 ОГЛЯД СИСТЕМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ANYLOGIC

3.1 Загальні відомості про систему моделювання AnyLogic

Система AnyLogic, розроблена компанією XJTechnologies (Росія), це середа комп'ютерного моделювання загального призначення. Це комплексний інструмент, що охоплює основні напрями моделювання: дискретно-подієве, системної динаміки, агентне. Використання AnyLogic дає можливість оцінити ефект конструкторських рішень в складних системах реального світу.

Інструментальна система AnyLogic не обмежує користувача однієї єдиної парадигмою моделювання, що є характерним для існуючих на ринку інструментів моделювання. У AnyLogic розробник може гнучко використовувати різні рівні абстрагування і різні стилі і концепції і змішувати їх при створенні однієї і тієї ж моделі.

Програмний продукт AnyLogic заснований на об'єктно-орієнтованій концепції. Об'єктно-орієнтований підхід до подання складних систем є кращим методом управління складністю інформації, ця концепція дозволяє простим і природним чином організувати і представити структуру складної системи. Таким чином, ідеї та методи, спрямовані на управління складністю, вироблені в останні десятиліття в області створення програмних систем, дозволяють розробникам моделей в середовищі AnyLogic організувати мислення, структурувати розробку і, в кінцевому рахунку, спростити і прискорити створення моделей.

Іншою базовою концепцією AnyLogic є представлення моделі як набору взаємодіючих паралельно функціонуючих активностей. Такий підхід до моделювання інтуїтивно дуже зрозумілий і природний в багатьох додатках, оскільки системи реального життя складаються з сукупності активностей, взаємодіючих з іншими об'єктами. Активний об'єкт AnyLogic –

це об'єкт зі своїм власним функціонуванням, що взаємодіє з оточенням. Він може включати в себе будь-яку кількість екземплярів інших активних об'єктів. Активні об'єкти можуть динамічно породжуватися і зникати відповідно до законів функціонування системи [13]¹⁾.

Графічне середовище моделювання AnyLogic підтримує проектування, розробку, документування моделі, виконання комп'ютерних експериментів з моделлю, включаючи різновиди аналізу – від аналізу чутливості до оптимізації параметрів моделі щодо деякого критерію (рис.3.1).

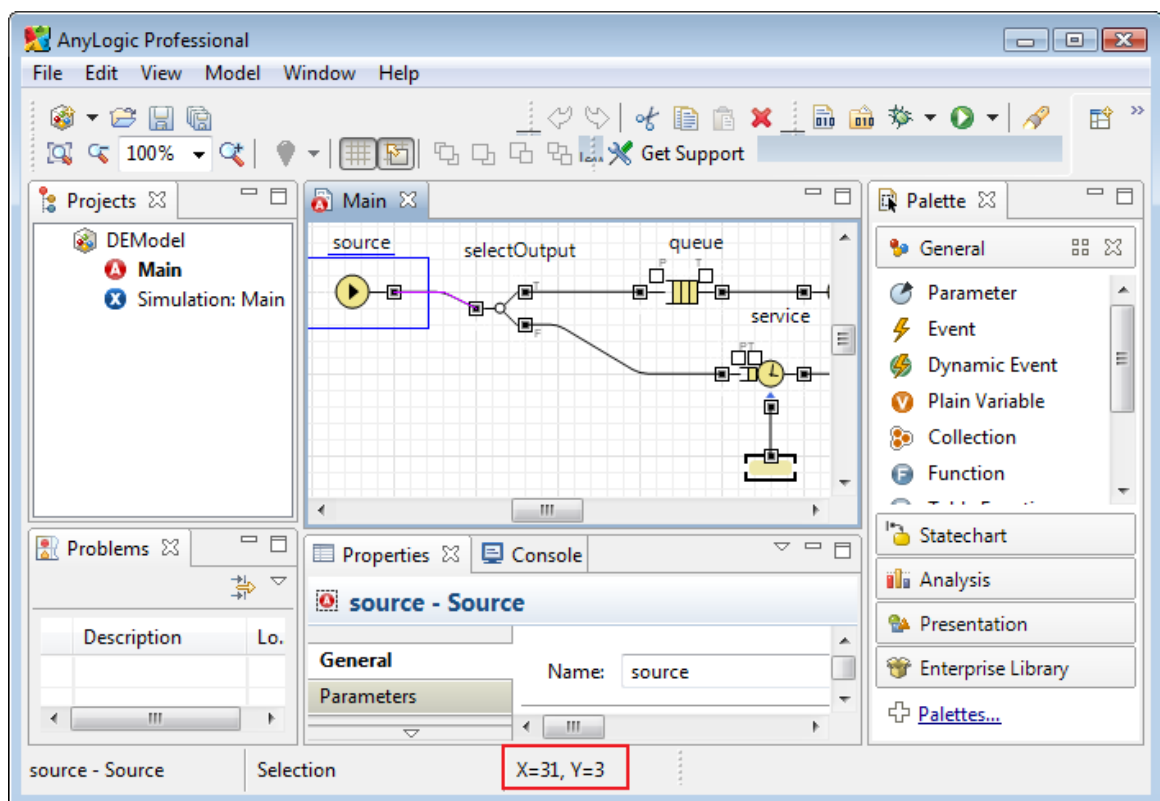


Рисунок 3.1 – Вигляд середовища редактору пакету AnyLogic

У результаті AnyLogic не обмежує користувача однією єдиною парадигмою моделювання, що є характерним фактично для всіх інструментів моделювання, існуючих сьогодні на ринку. У AnyLogic розробник може гнучко використовувати різні рівні абстрагування, різні стилі та концепції,

¹⁾ [13] Офіційний сайт XJ Technologies URL: <http://www.xjtek.com> (Дата звернення 12.11.2020)

будувати моделі в рамках тієї чи іншої парадигми і змішувати їх при створенні однієї і тієї ж моделі використовувати раніше розроблені модулі, зібрані в бібліотеки, доповнювати і будувати свої власні бібліотеки модулів. При розробці моделі на AnyLogic можна використовувати концепції з декількох «класичних» областей моделювання, наприклад, в агентній моделі використовувати методи системної динаміки для представлення змін стану середовища або в безперервній моделі динамічної системи врахувати дискретні моменти. Наприклад, аналіз IT-інфраструктури компанії (аналіз продуктивності серверів, вузьких місць локальної мережі і т. п.). Він легко проводиться за допомогою методів дискретного-подієвого моделювання, має небагато користі, якщо в моделі не виражений вплив можливих змін параметрів цієї інфраструктури на бізнес-процеси і, в кінцевому рахунку, на прибуток компанії, а такий зв'язок в моделі не може бути реалізований тільки засобами дискретно-подієвого моделювання.

У AnyLogic легко будуються подібні моделі з необхідним рівнем адекватності, що дозволяють відповісти на багато питань, що цікавлять дослідника. Багаті можливості анімації і візуального представлення результатів в процесі роботи моделі дозволяють зрозуміти суть процесів, що відбуваються в системі, що моделюється, спростити налагодження моделі. Зручний інтерфейс і численні засоби підтримки розробки моделей в AnyLogic роблять не тільки використання, а і створення комп'ютерних імітаційних моделей в цьому середовищі моделювання доступним навіть для початківців.

3.2 Етапи імітаційного моделювання в AnyLogic

Імітаційне моделювання складається з двох великих етапів: створення моделі та аналізу отриманих за допомогою моделі результатів з метою прийняття рішення.

При детальному розгляді, побудова дійсно корисної імітаційної моделі вимагає великої роботи. Спочатку розробник моделі повинен визначити, які завдання будуть вирішуватися з її допомогою, тобто моделюванню в будь-якій його формі повинне передувати формулювання мети моделювання. Від мети залежить те, які процеси в реальній системі слід виділити і відобразити в моделі, а від яких процесів абстрагуватися, які характеристики цих процесів враховувати, а які – ні, які співвідношення між змінними і параметрами моделі повинні бути відображені в моделі. Даний етап можна охарактеризувати як створення концептуальної (змістовної) моделі. На ньому відбувається структуризація моделі, тобто виділення окремих підсистем, визначення елементарних компонентів моделі та їх зв'язків на кожному рівні ієрархії. У імітаційному моделюванні структура моделі відображає структуру реального об'єкта моделювання на деякому рівні абстракції, а зв'язки між компонентами моделі є відображенням реальних зв'язків. Елементи системи, їх зв'язки, параметри і змінні, а також їх співвідношення і закони їх зміни повинні бути виражені засобами середовища моделювання, тобто в цьому середовищі повинні бути визначені змінні і параметри моделі, побудовані процедури обчислення змінних і характеристик моделі в часі.

При необхідності для більшого розуміння процесів, що протікають в моделі, має бути розроблено анімаційне уявлення цих процесів. Потім побудована модель повинна бути перевірена з точки зору коректності її реалізації.

Наступний етап – це калібрування або ідентифікація моделі, тобто збір даних і проведення вимірювань тих характеристик в реальній системі, які повинні бути введені в модель у вигляді значень параметрів і розподілів випадкових величин.

Далі, необхідно виконати перевірку правильності моделі (її валідацію), яка полягає в тому, що вихід моделі перевіряється на декількох тестових режимах, в яких, характеристики поведінки реальної системи відомі або

очевидні. Останнім етапом роботи з моделлю є комп'ютерний експеримент, тобто власне те, заради чого і створювалася модель.

У найпростішому випадку експеримент – це виконання моделі при різних значеннях її існуючих параметрів (факторів) і спостереження її поведінки з реєстрацією характеристик поведінки. Цей вид використання моделі називається прогнозом, або експериментом типу «що буде, якщо ...». Комп'ютерне моделювання дозволяє не тільки отримати прогноз, але й визначити, які керуючі впливи на систему приведуть до сприятливого розвитку подій. Більш складні експерименти дозволяють виконати аналіз чутливості моделі, оцінку ризиків різних варіантів керуючих рішень, а також оптимізацію для визначення параметрів і умов раціонального функціонування моделі.

Одне з важливих питань – представлення та аналіз результатів моделювання. Для цього в інструментальному середовищі можуть бути використані спеціальні засоби для обробки статистичної інформації, для представлення в структурованому або графічному вигляді отримання даних, інтеграція з зовнішніми базами даних і т.п.

Часто імітаційна модель використовується як модуль більшої системи прийняття рішення, що одержує в режимі реального часу дані моніторингу стану керованої системи, що оцінює, до яких наслідків може призвести поточна ситуація, і що пропонує оптимальне (або просто раціональне) рішення для мінімізації негативних наслідків розвитку системи в майбутньому. Для цього зазвичай потребується інтеграція моделі з іншими інформаційними системами та розробка спеціального інтерфейсу користувача.

3.3 Основні концепції, реалізовані в AnyLogic

Дві фази імітаційного моделювання. У AnyLogic дві фази імітаційного моделювання – розробка моделі та її аналіз – явно розділені. Розробка моделі

виконується в середовищі редактора AnyLogic, аналіз моделі відбувається в середовищі виконання. У кожній фазі існують свої засоби управління. Перехід з однієї фази в іншу провадиться дуже легко. Можна, можливо багаторазово використовувати перехід між фазами редагування і виконання моделі при розробці моделі [13]¹⁾.

Активні об'єкти, класи і екземпляри активних об'єктів. Клас в програмуванні є потужним засобом, що дозволяє структурувати складну систему. Клас визначає шаблон, відповідно до якого будуються окремі екземпляри класу. Ці екземпляри можуть бути визначені як об'єкти інших активних об'єктів.

У AnyLogic основним структурним блоком при створенні моделей є класи активних об'єктів. Використання активних об'єктів є природним засобом структуризації моделі складних систем: світ складається з безлічі паралельно функціонуючих і взаємодіючих між собою сутностей. Різні типи цих сутностей і представляють різні активні об'єкти.

Щоб створити модель AnyLogic, потрібно створити класи активних об'єктів (або використовувати об'єкти бібліотек AnyLogic). Визначення активного об'єкта задає шаблон, і окремі об'єкти, побудовані відповідно з цим шаблоном (екземпляри активного об'єкта), можуть використовуватися потім як елементи інших активних об'єктів. У будь-який клас можуть бути включені декілька екземплярів інших класів, які можуть відрізнятися своїми параметрами. Завжди один клас в моделі є кореневим. Для нього в моделі AnyLogic породжується один екземпляр з наперед визначеним ім'ям root, він і запускається виконавчою системою AnyLogic на виконання. Ім'я класу кореневого активного об'єкта можна міняти у вікні його властивостей.

Кожен активний об'єкт має структуру (сукупність включених до нього активних об'єктів та їх зв'язки), а також поведінку, обумовлену сукупністю змінних, параметрів, стейтчартів і т. п. Кожен екземпляр активного об'єкта в

¹⁾ [13] Офіційний сайт XJ Technologies URL: <http://www.xjtek.com> (Дата звернення 12.11.2020)

моделі має свою власну поведінку. Він може мати свої значення параметрів, він функціонує незалежно від інших об'єктів, взаємодіючи з ними і з зовнішнім середовищем.

Об'єктно-орієнтований підхід. AnyLogic використовує об'єктно-орієнтований підхід до подання складних систем. Цей підхід дозволяє простим і природним чином організувати і представити структуру складної системи за допомогою ієрархії абстракцій. Наприклад, на деякому рівні абстракції автомобіль можна вважати певним єдиним об'єктом. Але більш детально його можна представити як сукупність взаємодіючих підсистем: двигуна, рульового управління, гальмової системи і т. п. Кожна з цих підсистем може бути представлена, якщо це необхідно, своєю структурою взаємодії підсистем.

Саме таку ієрархію абстракцій дозволяє створити AnyLogic при розробці моделей. Всю модель можна розглядати як єдиний об'єкт root. При детальному розгляді видно, що цей об'єкт може містити, наприклад, два екземпляри класу MyClass з іменами myClass і myClass1. Сам клас MyClass може мати свою складну будову, яка буде прихована в класі Root. Така ієрархія структури може бути довільної глибини.

Візуальна розробка моделі. При побудові моделі в середовищі AnyLogic не передбачено можливості використання ніяких інших засобів, крім засобів візуальної розробки (введення станів і переходів стейтчарта, введення піктограм змінних і т. п.), завдання чисельних значень параметрів, аналітичних записів співвідношень змінних та аналітичних записів умов настання подій. Основною парадигмою, прийнятою в AnyLogic при розробці моделей, є візуальне проектування – побудова за допомогою графічних об'єктів і піктограм ієрархій структури та поведінки активних об'єктів.

Вбудована мова Java. AnyLogic є надбудовою над мовою Java – однієї з найбільш потужних і в той же час простих сучасних об'єктно-орієнтованих мов програмування. Всі об'єкти, визначені користувачем при розробці моделі на AnyLogic за допомогою його графічного редактора, транслюються в конструкції

мови Java, а потім відбувається компіляція всієї зібраної програми на Java, яка задає модель, в виконуваний код.

Хоча при побудові моделі на AnyLogic розробник використовує конструкції мови Java в більшій чи меншій мірі, насправді він ніколи не розробляє повні програми, він не програмує, а лише вставляє фрагменти коду в спеціально передбачені для цього поля вікна *Код* і вікон властивостей об'єктів моделі. Ці фрагменти висловлюють логіку конкретних кроків або дій в моделі. Як і всі інші включені в модель програмні фрагменти, ці вирази повинні бути синтаксично правильними конструкціями Java, тому розробник моделей AnyLogic повинен мати уявлення про цю мову.

Засоби опису поведінки об'єктів. Основним засобом специфікації поведінки об'єктів в AnyLogic є змінні, таймери і стейтчарти. Змінні відображають характеристики об'єкта, що змінюються. Таймери можна заводити на певний інтервал часу і після закінчення цього інтервалу виконувати задану дію. Стейтчарти дозволяють візуально уявити поведінку об'єкта в часі під впливом подій або умов, вони складаються з графічного зображення подій і переходів між ними. Будь-яка складна логіка поведінки об'єктів моделі в AnyLogic може бути виражена за допомогою комбінації стейтчартів, диференціальних та алгебраїчних рівнянь, змінних, таймерів і програмного коду на Java. Алгебраїчні і диференціальні рівняння, як і логічні вирази, записуються в AnyLogic аналітично.

Імітація декількох паралельно протікаючих процесів. Інтерпретація будь-якого числа паралельно протікаючих процесів в моделі AnyLogic прихована від користувача. Ніякого календаря подій розробник моделі на AnyLogic не веде, відстеження подій у всіх процесах, визначених у моделі, виконується системою автоматично. Ніяких зусиль для організації квазіпаралелізму інтерпретації при цьому не вимагається.

Модельний і реальний час. Поняття модельного часу є базовим в системах імітаційного моделювання. Модельний час – це умовний логічний час, в одиницях якого визначено поведінку всіх об'єктів моделі.

У моделях AnyLogic модельний час може змінюватися або безперервно, якщо поведінка об'єктів описується диференційними рівняннями, або дискретно, перемикаючись від моменту настання однієї події до моменту настання наступної події, якщо в моделі присутні тільки дискретні події. Моменти настання всіх планованих подій в дискретній моделі виконавча система зберігає в так званому календарі подій, вибираючи звідти найбільш ранню подію для виконання пов'язаних з нею дій. Значення поточного часу в моделях AnyLogic може бути отримано зверненням до функції `getTime()`.

Одиницю модельного часу розробник моделі може інтерпретувати як будь-який відрізок часу: секунду, хвилину, годину або рік. Важливо тільки, щоб всі процеси, що залежать від часу, були виражені в одних і тих же одиницях. При моделюванні фізичних процесів всі параметри і рівняння повинні бути виражені в одній і тій же системі фізичних величин.

Інтерпретація моделі виконується на комп'ютері. Фізичний час, що витрачається процесором на імітацію дій, які повинні виконуватися в моделі протягом однієї одиниці модельного часу, залежить від багатьох факторів. Тому одиниця фізичного і одиниця модельного часу незбігаються.

У AnyLogic прийняті два режими виконання моделей: режим віртуального часу і режим реального часу. У режимі віртуального часу процесор працює з максимальною швидкістю без прив'язки до фізичного часу. Цей режим використовується для факторного аналізу моделі, набору статистики, оптимізації параметрів моделі і т. п. Оскільки анімація та інші вікна спостереження за поведінкою моделі зазвичай істотно знижують продуктивність комп'ютера, для підвищення швидкості виконання моделі ці вікна потрібно закрити.

У режимі реального часу користувач задає зв'язок модельного часу з фізичним часом, тобто встановлюється обмеження на швидкість роботи процесора при виконанні моделі. У цьому режимі задається кількість одиниць модельного часу, які повинні виконуватися процесором в одну секунду. Зазвичай даний режим включається для того, щоб візуально уявити

функціонування системи в реальному темпі настання подій, проникнути в суть процесів, що відбуваються в моделі.

Співвідношення фізичного і модельного часу при роботі моделі в режимі реального часу можна зрозуміти на наступному прикладі. При коефіцієнті прискорення 4, якщо процесор встигає виконати менш ніж за 1 с. всі операції, які в моделі визначені протягом чотирьох одиниць модельного часу, він буде чекати до кінця секунди. Якщо ж процесор не встигає зробити це, то у нього не буде інтервалу очікування, і коефіцієнт прискорення буде менше того, який встановлений користувачем.

Анімація поведінки моделі. Зручні засоби розробки анімаційного представлення моделі в AnyLogic дозволяють уявити функціонування модельованої системи в живій формі динамічної анімації, що дозволяє «побачити» поведінку складної системи.

Засоби анімації дозволяють користувачеві легко створити віртуальний світ, сукупність графічних образів, керований динамічними параметрами моделі за законами, визначеним користувачем за допомогою рівняння і логіки модельованих об'єктів. Візуальне подання поведінки системи допомагає користувачеві проникнути в суть процесів, що відбуваються в системі.

Інтерактивний аналіз моделі. Багато систем моделювання дозволяють змінювати параметри моделі тільки до запуску моделі на виконання. Система AnyLogic надає користувачеві можливість змінювати параметри моделі в ході її функціонування. Тому вікно анімації можна назвати «стендом» для проведення комп'ютерного експерименту з моделлю. Основним прикладом засобів, які змінюють параметри моделі, є слайдери (бігунки).

3.4 Об'єкти Enterprise Library

Бібліотека AnyLogic Enterprise Library підтримує дискретно-подієвий підхід моделювання. За допомогою об'єктів Enterprise Library можна моделювати системи реального світу, динаміка яких представляється як

послідовність операцій (прибуття, затримка, захоплення чи недолік ресурсу, поділ) над якимись сутностями (entities, транзакти, заявки).

Сутностями можуть бути, повідомлення, запити, документи, дзвінки, пакети даних, транспортні засоби, технічні пристрої і т.п. Ці сутності пасивні і самі не контролюють свою динаміку. Але вони можуть володіти певними атрибутами, що впливають на процес їх обробки (наприклад, тип дзвінка, складність роботи, довжина повідомлення) або накопичують статистику (загальний час очікування, вартість, час обробки).

Процеси задаються у формі потокових діаграм (блок-схем). Тобто в графічному поданні, прийнятому в багатьох областях (виробництві, бізнес-процесах, центрах обробки дзвінків, комутації повідомлень, логістиці, охороні здоров'я і т.п.). Потокові діаграми AnyLogic ієрархічні, масштабуються. Вони також об'єктно-орієнтовані, що дозволяє користувачеві моделювати складні системи будь-якого рівня детальності. Іншою важливою особливістю Enterprise Library є можливість створення досить складною анімації моделей.

У Enterprise Library також входять об'єкти, розроблені для моделювання процесів, що відбуваються в просторі. Тобто таких процесів, де об'єкти-заявки та ресурси переміщуються в якоїсь мережі. Це підмножина об'єктів значно спрощує моделювання деяких типів систем, наприклад, виробництва, внутрзаводської логістики, супермаркету, складу, госпіталю, майстерні.

Для використання цього підходу, званого мережевим моделюванням, необхідно визначити топологію мережі (наприклад, використовуючи векторну графіку AnyLogic поверх плану або креслення будівлі або споруди), безліч ресурсів (статичних, рухомих або що переміщуються), і власне процес. Процес в даному випадку – це комбінація об'єктів типу «переміститися туди» або «приєднати до себе ресурс» (таких, як NetworkMoveTo, NetworkSeize, NetworkSendTo) і звичайних об'єктів Enterprise Library (Queue, Delay, Service). Заявки та ресурси автоматично анімуються рухомими об'єктами за

сегментами мережі або знаходяться в її вузлах. Ця анімація може також комбінуватися із звичайною анімацією.

Ієрархічні моделі і повторно використовувані модельні компоненти. Якщо система, що моделюється, складна, має сенс розбити її модель на компоненти (підпроцеси) і помістити кожен з них в окремий активний об'єкт. Ви можете визначити входи і виходи з підпроцесу, помістити їх на зовнішній інтерфейс активного об'єкта і приховати його реалізацію. На верхньому рівні такими об'єктами оперують як блоками, поєднуючи їх входи і виходи. Можна створити кілька екземплярів активного об'єкта з різними параметрами, у тому числі і в інших проектах.

Розширення процесних моделей. Як і все в AnyLogic, процесна модель може бути розширена до будь-якого необхідного рівня детальності з необхідною нестандартною функціональністю.

По-перше, базові класи Entity і ResourceUnit можуть бути розширені власними класами з будь-якими додатковими полями і методами. Вказавши ці класи в параметрах класів об'єктів, можна дозволити простіший доступ до власних полів (без явного приведення типів, як це було в AnyLogic 5). Наприклад, в моделі аеропорту може знадобитися створити спеціальні класи, що моделюють пасажирів, багаж, офіцерів безпеки і т. п.

По-друге, у кожного об'єкта Enterprise Library є спеціальні «точки розширення» – місця, де можна задати якісь дії або вирази. Такими точками розширення є динамічні параметри (помічені в описах об'єктів як [динамічні]), обчислювані під час виконання моделі при проходженні заявок через процесну діаграму.

Створення анімації для процесних моделей. Бібліотека Enterprise Library тісно інтегрована з анімаційними засобами AnyLogic і дозволяє створювати анімації процесів будь якого ступеня складності, в тому числі ієрархічні і з кількома різними графічними уявленнями процесу. Наприклад, можна визначити глобальний погляд на процес виробництва з декількома агрегованими індикаторами, а також детальні анімації конкретних операцій –

і перемикається між ними. У мережевому моделюванні користувач малює мережу вузлів і сегментів, і заявки автоматично відображаються рухомими уздовж сегментів або залишаються в вузлах [15]¹⁾.

Комбінування процесних моделей з моделями і конструкціями інших типів. Одним з головних переваг AnyLogic є можливість комбінування різних стилів моделювання, що дозволяє відображати комплексність і неоднорідність систем реального світу. Можна комбінувати власні процесні моделі, побудовані за допомогою Enterprise Library, з моделями системної динаміки або агентними моделями, або ж просто створювати власні об'єкти за допомогою базових елементів AnyLogic і включати їх в діаграму, що описує процес.

Є безліч способів комбінування підходів моделювання, наведемо лише деякі приклади. Вплив системно-динамічної діаграми потоків і накопичувачів на процесну модель може бути реалізований, наприклад, як об'єкт Source, що створює заявки з інтенсивністю, керуючи динамічною змінною.

Зворотній зв'язок може бути представлений, наприклад, системно-динамічним накопичувачем, що збільшує своє значення при кожному прибутті заявки в об'єкт процесної діаграми Sink.

У певний момент свого життєвого циклу агент може ставати заявкою і вставлятися в процес за допомогою блоку Enter (і навпаки), або ж агент може існувати паралельно із заявкою і взаємодіяти з нею.

Якщо потрібен об'єкт зі специфічною функціональністю, відмінною від функціональності, що надається об'єктами Enterprise Library (і недосяжною шляхом комбінування цих об'єктів), можна створити власний клас активного об'єкта і використовувати об'єкти Exit і Enter в якості інтерфейсних елементів. Тоді є можливість вставляти екземпляри цього класу активного

¹⁾ Карпов, Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб.: БХВ-Петербург, 2005. 400 с.

об'єкта у власну процесну діаграму. У такому активному об'єкті можна використовувати діаграми станів, події, змінні і т. п.

Класи активних об'єктів бібліотеки AnyLogic Enterprise Library є блоками, за допомогою яких будуються блок-схеми, що моделюють процеси, які відбуватимуться із заявками. Бібліотечні класи Entity і Resource Unit є базовими класами для заявок і ресурсів відповідно [16]¹⁾.

¹⁾ [16] Борщев А.В. Практическое агентное моделирование и его место в арсенале аналитика URL: <http://www.gpss.ru/immod05/p/borshev/> (Дата звернення 12.11.2020)

4 ПОБУДОВА МОДЕЛІ ФУНКЦІОНУВАННЯ МЕРЕЖІ ЗВ'ЯЗКУ

4.1 Постановка задачі

Мережа зв'язку є СМО виду: багатозафазна, багатоканальна, з декількома неоднорідними потоками заявок на обслуговування, розімкнута, кінцевою надійністю, з чергами обмеженою ємності на окремих фазах обслуговування.

Мережа зв'язку має такі типові об'єкти (рис. 4.1): абонент (обладнання користувача); канал зв'язку; вузол зв'язку (комутаційне обладнання – маршрутизатор).

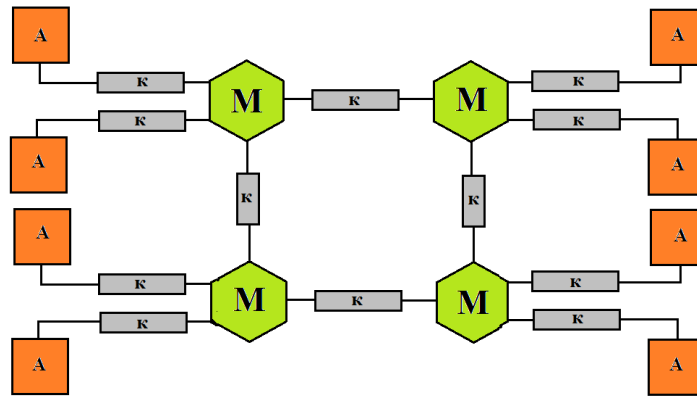


Рисунок 4.1 – Структура мережі зв'язку з типовими об'єктами

На маршрутизатори (М) по каналах (К) мережі надходять від абонентів (А) повідомлення через випадкові проміжки часу, розподілені по експонентному закону. Повідомлення відповідно до еталонної моделі взаємодії відкритих систем поділяються на пакети по 1458 байт. Повідомлення можуть бути чотирьох категорій: 1 – передача простих даних, 2 – аудіоінформація, 3 – відеоінформація, 4 – електронна пошта. Передаються пакети по мережі через маршрутизатори. Шлях проходження по мережі різних пакетів одного повідомлення може бути різний. Вибір шляху передачі здійснюється за «пріоритетом», який визначається спеціальним алгоритмом, закладеним в маршрутизатори.

Потрібно розробити класи ІМ типових об'єктів мережі зв'язку, здатних функціонувати без заздальгідь підготовленої і вкладеної в них користувачем інформації про структуру мережі, тобто структура мережі повинна збиратися об'єктами самостійно.

Маршрутизація на вузлах повинна проводитися за інформацією про структуру мережі, зібраної маршрутизаторами самостійно. Таким чином, досягається самостійність моделі в області її роботи. Розробнику тільки потрібно з'єднати об'єкти і визначити їх параметри, «прив'язавши» до характеристик реального модельованого обладнання мережі зв'язку.

Таким чином, необхідно розробити імітаційну модель функціонування мережі зв'язку. Дослідити вплив ємностей буферів, інтервалів часу надходження повідомлень, їх обчислювальних труднощів і інших параметрів на показники функціонування мережі з метою їх оцінки та прийняття рішень при необхідності щодо поліпшення якості обслуговування мережі.

Побудову моделі почнемо з створення активних об'єктів – примірників типів агентів. Згідно з рис. 4.1 в мережі зв'язку виділені наступні сегменти (компоненти): абонент; маршрутизатор; канал.

Реалізуємо ці сегменти засобами AnyLogic, а потім з них будемо будувати мережу на кореновому об'єкті Main.

Імітаційна модель об'єктів класу складається з трьох частин: характеристики (вихідні дані), результати моделювання, подієва частина. Розглянемо призначення і особливості побудови цих частин імітаційних моделей трьох класів.

4.2 Опис параметрів моделі

Розглянемо функціонування мережі, вигляд якої наведений на рис. 4.2. Повідомлення надходять від 5 джерел. Інтервали надходження повідомлень, інтервали між відмовами і час відновлення працездатності розподіляються за експоненціальним законом (exponential), а обчислювальні складності

повідомлень в залежності від категорій – за нормальним законом (Normal). Для деяких однакових параметрів з метою спрощення прийнято, що вони мають рівні значення, наприклад, середні значення інтервалів надходження повідомлень. Модель же буде побудована універсальною так, що всі ці значення можуть бути будь-якими.

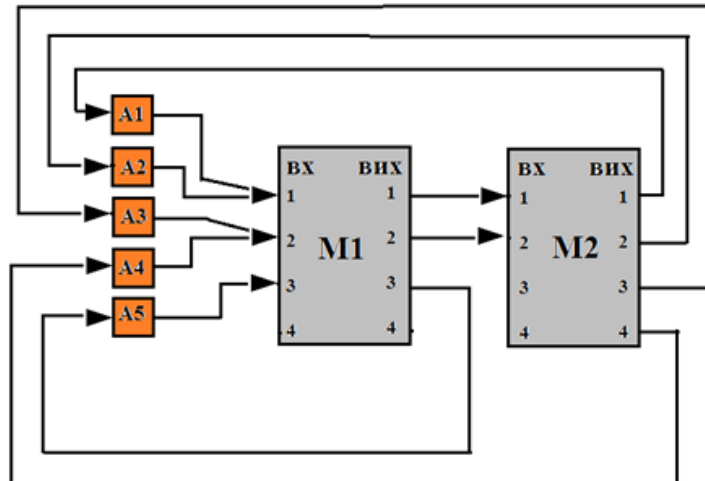


Рисунок 4.2 – Схема мережі зв'язку

Повідомлення абонентів A1 і A2 надходять на вхід 1 маршрутизатору M1, абонентів A3 і A4 – на вхід 2 маршрутизатора M1, абонента A5 – на вхід 3 маршрутизатора M1.

Маршрутизатор M1 налаштований таким чином, що повідомлення, адресовані абонентам A1 і A2 надходять на вхід 1 маршрутизатору M2, а абонентам A3 і A4 – на вхід 2 маршрутизатора M2.

Маршрутизатор M2 налаштований так, що його виходи 1 ... 4 підключені до каналів зв'язку, по яких передаються повідомлення, що адресовані абонентам A1 ... A4 відповідно.

Видно, що система зв'язку є багатофазною, багатоканальною системою масового обслуговування замкнутого типу з обмеженими ємностями буферів (накопичувачів), тобто з відмовами.

Маршрутизатор в загальному вигляді можна також представити як систему масового обслуговування, блок-схема якої наведена у додатку А.

У маршрутизаторі виділені блок контролю 1, накопичувач (буфер) обчислювального комплексу ВК, безпосередньо ВК, блок контролю 2, накопичувачі напрямків зв'язку.

Повідомлення, що надходять на маршрутизатор, повинні мати наступні параметри:

- numAbOtr – номер абонента-відправника повідомлення;
- numAbPol – номер абонента-отримувача повідомлення;
- numKat – номер категорії повідомлення;
- timeOtr – час відправлення повідомлення абонентом;
- dlina – довжина повідомлення, байт;
- timeObr – час обробки повідомлення ВК, с;
- timePered – час передачі повідомлення по каналу зв'язку, с.

Ці параметри виходять шляхом розіграшу за такими вихідними даними:

- kolAbonent – кількість абонентів;
- timeAbonent – середній час відправлення повідомлень абонентом, с;
- verKat = {verKat1, verKat2, verKat3, verKat4} – ймовірності розподілу видів повідомлень першої, другої, третьої і четвертої категорій відповідно,
- verKat1 + verKat2 + verKat3 + verKat4 = 1;
- dlKat = {dlKat1, dlKat2, dlKat3, dlKat4} – середні довжини повідомлень, байт, першої, другої, третьої і четвертої категорій відповідно;
- dlKat0 = {dlKat01, dlKat02, dlKat03, dlKat04} – стандартні відхилення довжин повідомлень, байт, першої, другої, третьої і четвертої категорій відповідно;
- proizvod – продуктивність ВК при обробці повідомлень, оп/с;
- skorPeredKan – середній час передачі повідомлень по каналу зв'язку, біт/с;
- skorPeredKanR – середній час передачі повідомлень по резервному каналу зв'язку, біт/с;

– $timeBklKanR$ – час включення резервного каналу зв'язку.

В ході моделювання збирається наступна статистика:

– $kolOtrKat1$, $kolOtrKat2$, $kolOtrKat3$, $kolOtrKat4$, $kolOtr$ – кількість відправлених абоненту повідомлень першої, другої, третьої, четвертої та всіх категорій відповідно;

– $kolPolKat1$, $kolPolKat2$, $kolPolKat3$, $kolPolKat4$, $kolPol$ – кількість отриманих абонентом повідомлень першої, другий, третій, четвертій і всіх категорій відповідно;

– $всьогоOtrKat1$, $всьогоOtrKat2$, $всьогоOtrKat3$, $всьогоOtrKat4$, $всьогоOtr$ – кількість відправлених в мережі повідомлень першої, другої, третьої, четвертої та всіх категорій відповідно;

– $всьогоПолKat1$, $всьогоПолKat2$, $всьогоПолKat3$, $всьогоПолKat4$, $всьогоПол$ – кількість отриманих в мережі повідомлень першої, другої, третьої, четвертої та всіх категорій відповідально;

– кількість відправлених повідомлень кожним абонентом кожному абоненту мережі;

– кількість отриманих повідомлень кожним абонентом від кожного абонента мережі.

За цим статистичними даними розраховуються:

– коефіцієнти пропускної здатності між усіма абонентами мережі;

– $коэфПропСпособ$ – коефіцієнт пропускної здатності мережі зв'язку;

– $врПередачі$ – середній час передачі абоненту одного повідомлення;

– $врПередСооб$ – середній час передачі в мережі зв'язку одного повідомлення.

Коефіцієнти пропускної здатності визначаються як відносна кількості отриманих повідомлень до відправлених повідомлень.

У моделі також для виконання умов обмеження ємностей буферів необхідно використовувати:

– $emkBuferVx$ – ємність вхідного буфера абонента, байт;

– $tekEmkBuferVx$ – поточну ємність вхідного буфера абонента, байт;

- emkBufer1 – ємність вхідного буфера маршрутизатора, байт;
- tekEmkBufer1 – поточна ємність вхідного буфера, байт;
- emkBuferNap1, emkBuferNap2, emkostBuferNap3, emkBuferNap4 – ємності на входах каналів зв'язку першого, другого, третього і четвертого напрямків відповідно, байт;
- tekEmkNap1, tekEmkNap2, tekEmkNap3, tekEmkNap4 – поточні ємності на входах каналів зв'язку першого, другого, третього і четвертого напрямків відповідно, байт.

Відмови і відновлення ВК і каналів зв'язку розігруються за наступними даними:

- timeOtkBK – середній час між відмовами ВК;
- timeVosstBK – середній час відновлення ВК;
- timeOtkKan – середній час між відмовами каналу зв'язку;
- timeVosstKan – середній час відновлення каналу зв'язку

4.3 Моделювання абонента мережі

4.3.1 Вихідні дані

Створимо три нових типи агентів: Абонент, Канал, Маршрутизатор.

На кожному типі агентів Main, Абонент, Канал, Маршрутизатор створимо області перегляду, на яких будемо розміщувати елементи подієвої частини моделі, а також вихідні дані, змінні для проміжних обчислень і результати моделювання.

В області перегляду для агента Абонент додамо необхідні для моделювання параметри і змінні, призначення яких розглянуте у п.4.2 (рис.4.3).

Змінні призначені для проміжних обчислень і збору статистичних даних для одного абонента мережі. Значення властивостей параметрів моделювання наведені у табл. 4.1.

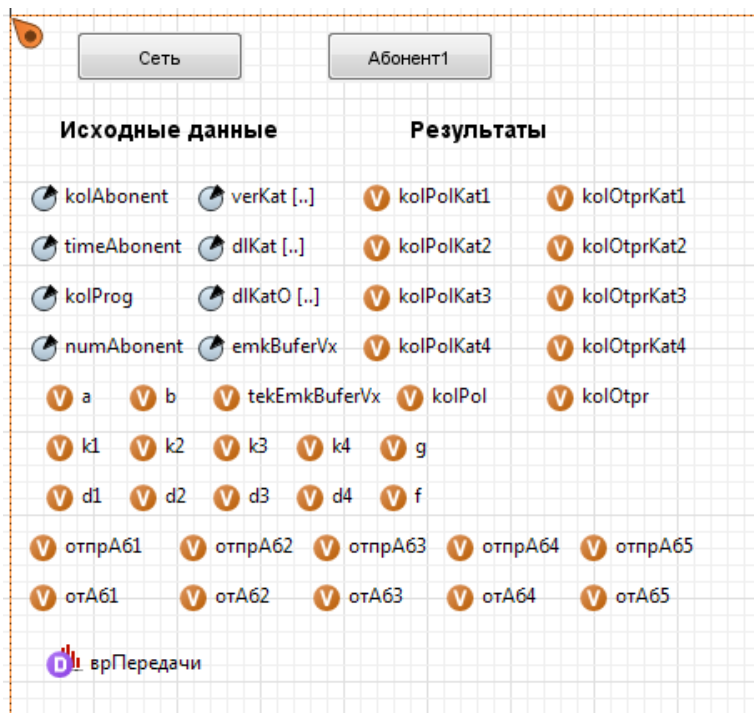


Рисунок 4.3 – Розміщення параметрів і змінних області перегляду агента Абонент

Таблиця 4.1 – Значення властивостей параметрів моделювання об'єкту Абонент

Ім'я	Масив	Розмірність	Значення за замовчуванням
kolAbonent	–	–	6
timeAbonent	–	–	30
kolProg	–	–	100
numAbonent	–	–	1
emkBuferVx	–	–	80000
verKat	так	KolKat	{0.3, 0.5, 0.7, 1.0}
dlKat	так	KolKat	{53000,86000,66000,50000}
dlKatO	так	KolKat	{6100,5000,7000,500}

4.3.2 Перегляд результатів моделювання

Варіант розміщення елементів для виведення результатів моделювання по кожному абоненту показаний на рис. 4.4.

Перш ніж перейти до побудови подієвої частини сегмента Абонент, розмістимо на кореновому агенті Main необхідні елементи для збору статистичних даних, обробки, розрахунків показників якості обслуговування мережі та їх виведення (рис.4.5).

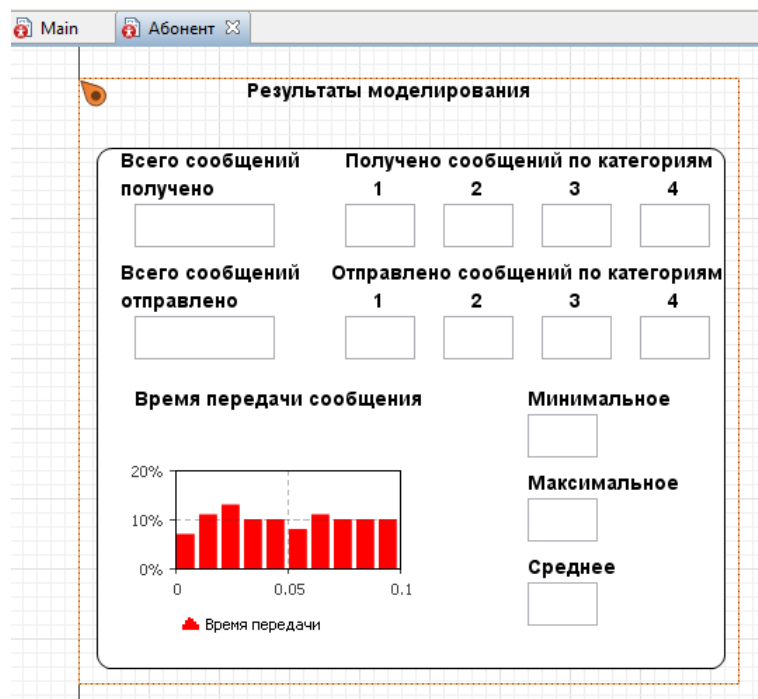


Рисунок 4.4 – Розміщення елементів для виведення результатів моделювання

Тип всіх змінних `double`. Змінні `отпр11 ... отпр55` призначені для накопичення статистичних даних про кількість відправлених повідомлень абонент – абонент, наприклад, `отпр23` – кількість повідомлень, відправлених абонентом 2 абоненту 3, а `отпр32` – кількість повідомлень, відправлених абонентом 3 абоненту 2. За цими даними розраховуються коефіцієнти пропускної здатності `кПрСп11 ... кПрСп55` абонент – абонент.



Рисунок 4.5 – Элементы сбора статистики функціонування мережі

Далі організуємо зручне розташування елементів для виведення кількісних показників якості обслуговування мережі зв'язку. Частина результатів моделювання будемо виводити в такому ж вигляді і такі ж, як і для кожного абонента (рис.4.6). В області перегляду додамо текстові поля, які будуть відображати отримані в результаті експерименту значення коефіцієнтів пропускної здатності між абонентами. Зрозуміло, що текстові поля: КПрСп11, КПрСп22, КПрСп33, КПрСп44, КПрСп55 залишаться порожніми, тому що пакети, що призначені тому ж самому абоненту не передаються у канал зв'язку.

Всего сообщений получено в сети	Получено сообщений по категориям			
<input type="text"/>	1 <input type="text"/>	2 <input type="text"/>	3 <input type="text"/>	4 <input type="text"/>
Всего сообщений отправлено в сети	Отправлено сообщений по категориям			
<input type="text"/>	1 <input type="text"/>	2 <input type="text"/>	3 <input type="text"/>	4 <input type="text"/>
Время передачи сообщения	Минимальное		Коэффициент пропускной способности сети	
<p>Время передачи</p>	Максимальное			
	Среднее			
	<input type="text"/>			
Коэффициенты пропускной способности абонент - абонент				
КПрСп11 <input type="text"/>	КПрСп12 <input type="text"/>	КПрСп13 <input type="text"/>	КПрСп14 <input type="text"/>	КПрСп15 <input type="text"/>
КПрСп21 <input type="text"/>	КПрСп22 <input type="text"/>	КПрСп23 <input type="text"/>	КПрСп24 <input type="text"/>	КПрСп25 <input type="text"/>
КПрСп31 <input type="text"/>	КПрСп32 <input type="text"/>	КПрСп33 <input type="text"/>	КПрСп34 <input type="text"/>	КПрСп35 <input type="text"/>
КПрСп41 <input type="text"/>	КПрСп42 <input type="text"/>	КПрСп43 <input type="text"/>	КПрСп44 <input type="text"/>	КПрСп45 <input type="text"/>
КПрСп51 <input type="text"/>	КПрСп52 <input type="text"/>	КПрСп53 <input type="text"/>	КПрСп54 <input type="text"/>	КПрСп55 <input type="text"/>

Рисунок 4.6 – Виведення результатів якості обслуговування мережі

4.3.3 Побудова подієвої частини об'єкту «Абонент»

Подієва частина активного об'єкту Абонент, призначена для імітації відправника-одержувача повідомлень, надходження повідомлень через випадкові інтервали часу, розіграшу параметрів повідомлень та рахунку кількості всього відправлених-отриманих повідомлень за категоріями і абонентам, запам'ятовування часу надходження кожного повідомлення, для використання в подальшому розрахунку мінімального, максимального і середнього часу передачі одного повідомлення.

Алгоритм блоку Абонент наведено у додатку Б, а його реалізація засобами AnyLogic – на рис. 4.7.

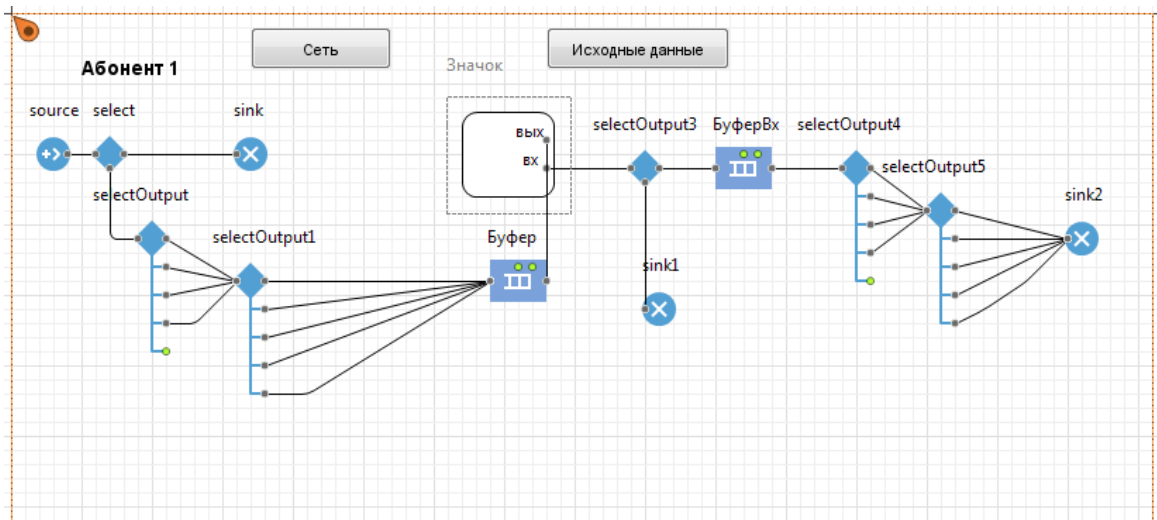


Рисунок 4.7 – Реалізація блоку Абонент засобами AnyLogic

При побудові моделі необхідно створити нестандартний тип заявки з додатковими полями для запису і зберігання параметрів моделювання. Скористуємося для цього Java-кодом. Створимо клас заявки Message, який буде спадкуватися від базового класу Entity. Програмний код класу заявки Message наведений у додатку Г.

Схема, що реалізована на рис.4.6 включає окремі компоненти бібліотеки моделювання процесів AnyLogic. Властивості цих компонентів, що були задані в моделі наведені у додатку Д.

Об'єкт source буде генерувати повідомлення з інтервалами часу, розподілених за експоненціальним законом. Прийнято, що повідомлення від даного абонента-відправника з рівною ймовірністю можуть бути відправлені будь-якому абоненту мережі. Тому розігрується номер абонента-отримувача повідомлення, і заноситься в numAbPol. В поле timeOtrp заноситься час відправлення повідомлення.

Об'єкт select виділяє і направляє об'єкту sink на знищення повідомлення, адресовані абоненту-відправнику, тобто самому собі.

Об'єкт `selectOutput` призначений для розіграшу категорій повідомлень, що відправляються. Він дозволяє провести рахунки відправлених повідомлень по мережі зв'язку в цілому і за категоріями повідомлень, визначити кількості відправлених повідомлень за один прогін, розіграшу довжин повідомлень за категоріями, виведення розрахунків по кожному абоненту, а також по мережі зв'язку в цілому.

Об'єкт `selectOutput1` призначений для розділення і рахунку повідомлень, що відправляються по абонентам. Повідомлення, що відправляються надходять в об'єкт буфер, далі воно має потрапити до каналу зв'язку. Створимо порти для відправлення та прийому повідомлень. Отже, повідомлення, що відправляються, надходять в канал зв'язку, підключений до порту вих. Через порт `vx` повідомлення надходять з каналу зв'язку. Об'єкт `selectOutput3` призначений для контролю поточної ємності вхідного буферу, у випадку заповнення буфера, повідомлення губляться. Об'єкт `selectOutput4` необхідний для поділу потоку отриманих повідомлень за категоріями. Він призначений для: рахунку отриманих повідомлень по мережі зв'язку в цілому і за категоріями повідомлень, визначення числа отриманих повідомлень за один прогін, виведення розрахунків для кожного абонента, а також по мережі зв'язку в цілому. Об'єкт `selectOutput5` призначені для поділу і рахунку отриманих повідомлень по абонентам. Код об'єкту `sink2` призначений для розрахунку коефіцієнта пропускної спроможності мережі зв'язку і часу передачі одного повідомлення.

4.4 Моделювання маршрутизатора

4.4.1 Вихідні дані

В області перегляду агенту Маршрутизатор додамо параметри і змінні необхідні для моделювання, призначення яких було розглянуто у п.4.2

(рис.4.8). Значення параметрів встановимо згідно табл. 4.2. Тип timeOtkBK, timeVosstBK, proizvod встановимо double, типи інших параметрів і простих змінних – int.

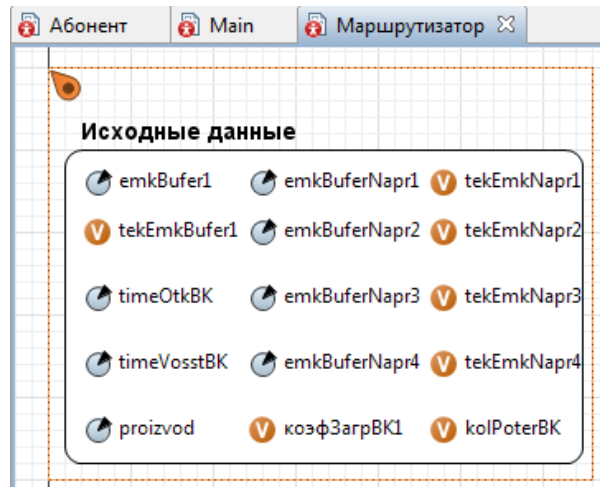


Рисунок 4.8 – Розміщення параметрів і змінних області перегляду агента
Маршрутизатор

Таблиця 4.2 – Значення властивостей параметрів моделювання об'єкту
Маршрутизатор

Ім'я	Значення за замовчуванням	Ім'я	Значення за замовчуванням
emkBufer1	5000000	emkBuferNap1	emkBuferNap2
timeOtkBK	3600	emkBuferNap3	emkBuferNap4
timeVosstBK	3.7	250000	250000
proizvod	40000	250000	250000

Далі при виконанні різних експериментів моделювання будемо мати можливість змінювати параметри моделі маршрутизатора, що дасть змогу дослідити зміну коефіцієнтів завантаження мережі, кількості загублених пакетів та інших показників якості мережі зв'язку, і дозволить оптимізувати їх. Оберемо час відмови обчислювального комплексу – 3600 с., а час його відновлення – 3.7 с. Продуктивність ОК при обробці повідомлень – 40000 оп/с.

4.4.2 Побудова подієвої частини об'єкту «Маршрутизатор»

Елементи подієвої частини об'єкту Маршрутизатор приведені на рис. 4.9. Об'єкт включає обчислювальний комплекс (ОК), буфер 2, порти входу-виходу, імітатор відмов обчислювального комплексу. У свою чергу обчислювальний комплекс містить блок контролю 1, буфер 1, блок обробки повідомлень.

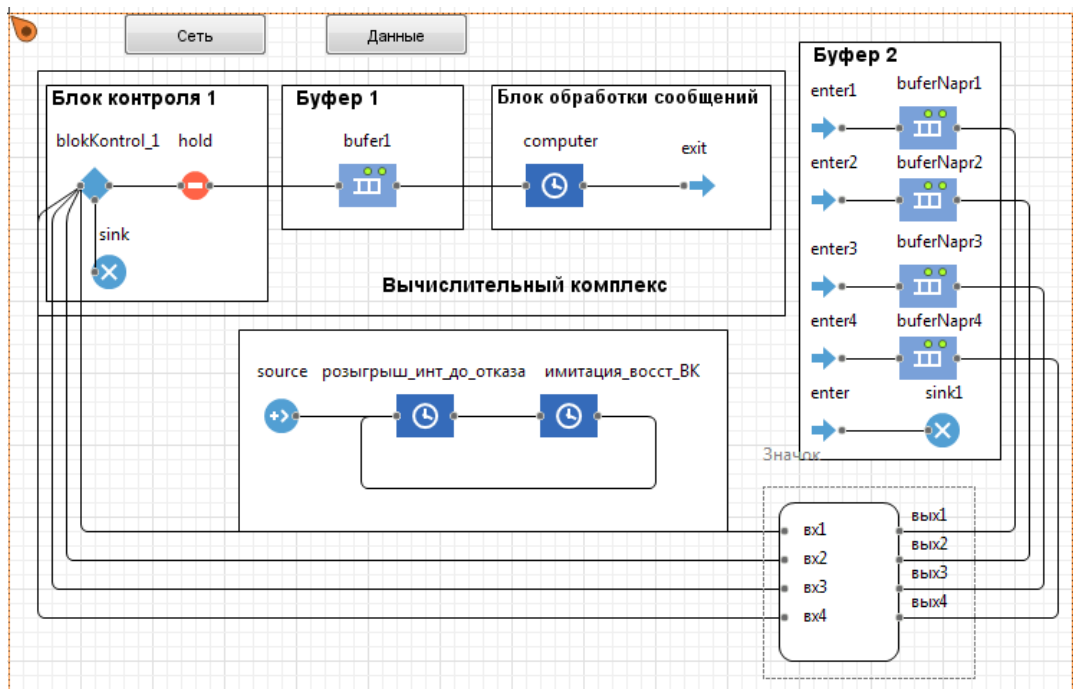


Рисунок 4.9 – Реалізація блоку Маршрутизатор засобами AnyLogic

Розглянемо докладніше подієву частину об'єкту Маршрутизатор.

Блок контролю 1 призначений для контролю поточної ємності буфера 1 маршрутизатора. Він аналізує наявність в буфері 1 вільної пам'яті, достатньої для зберігання повідомлення, що надійшло, і в залежності від результату аналізу повідомлення або розміщується в буфер 1, або знищується.

Алгоритм роботи Блоку контролю 1 представлений у додатку В.1. У AnyLogic цей алгоритм реалізується блоками `selectOutput`, `hold` і `sink`. Властивості цих компонентів наведені у табл. 4.3.

Таблиця 4.3 – Властивості компонентів Блоку контролю 1

Властивості	selectOutput
Ім'я:	blokKontrol_1
Тип заявки:	Message
Вихід true вибирається	При виконанні умови
Умова:	(EmkostBufer1-tekEmkostBufer1 > = entity.dlina) && (Hold.isBlocked () == false)
Властивості	sink
Тип заявки:	Message
Дія при вході:	kolPotBK ++;

Блок sink виконує розрахунок кількості повідомлень, втрачених при відмові обчислювального комплексу.

Блок *Буфер 1* призначений для прийому, розміщення і зберігання повідомлень, що надходять на обробку. Алгоритм роботи блоку Буфер 1 наведено у додатку В.2. У AnyLogic алгоритм блоку Буфер 1 реалізується об'єктом queue, який виконує функції черги (FIFO).

Поточна ємність буфера emkBufer1. При приміщенні повідомлення в буфер його поточна ємність збільшується, тому у полі «Дії при вході» потрібно записати:

```
tekEmkBufer1 += entity.dlina;
```

При виході повідомлення з буфера його поточна ємність зменшується, тому у полі «Дії при виході» слід записати:

```
tekEmkBufer1 -= entity.dlina;
entity.timeObr = entity.dlina / proizvod;
```

Блок *обробки повідомлень* призначений для імітації обробки повідомлень. Алгоритм роботи блока наведений на у додатку В.3.

Для реалізації алгоритму Блоку обробки повідомлень в AnyLogic використовується об'єкт delay. Час затримки розподіляється за експоненціальним законом: `exponential(1/entity.timeObr)`.

Блок контролю 2 призначений для розподілу повідомлень за напрямками і контролю поточних ємностей буферів (накопичувачів) напрямків передачі повідомлень.

Алгоритм роботи Блоку контролю 2 наведено у додатку В.4 Спочатку визначається номер напрямку, за яким має бути передано повідомлення, що надійшло. Потім визначається наявність достатньої вільної пам'яті в буфері цього напрямку. При відсутності потрібного обсягу пам'яті повідомлення втрачається.

Для розподілу повідомлень за напрямками можна було б використовувати об'єкти `selectOutput5` і `sink`. Але в цьому випадку будемо використовувати інші об'єкти `AnyLogic`: `exit` і `enter`. Вони дозволяють організувати складну маршрутизацію, тому на схемі у рис.4.8 блок контролю 2 не виділено, хоча функціонально він існує.

У об'єкта `exit` в полі Дії при виході додамо наступний код:

```
int i;
i=entity.numIstPol;
{
  switch (i) {
  case 1:if (emkBuferNaprl- tekEmkNaprl>=entity.dlina)
    { enter1.take(entity); break;}
    else {enter.take(entity);
    break;}
  case 2:if (emkBuferNaprl-tekEmkNaprl>=entity.dlina)
    { enter1.take(entity); break;}
    else {enter.take(entity);
    break;}
  case 3:if (emkBuferNaprl2-tekEmkNaprl2>=entity.dlina)
    { enter2.take(entity); break;}
    else {enter.take(entity);
    break;}
  case 4:if (emkBuferNaprl2-tekEmkNaprl2>=entity.dlina)
    { enter2.take(entity); break;}
    else {enter.take(entity);
```



```

        break;}
    case 5:if (emkBuferNapr3-tekEmkNapr3>=entity.dlina)
        { enter3.take(entity); break;}
        else {enter.take(entity);
            break;}
    } }

```

Маршрутизатор налаштовується певним чином, наприклад, таблицею маршрутизації. В даному випадку він налаштовується програмним шляхом так, що повідомлення першого і другого відправників передаються по першому напрямку, третього і четвертого відправників – за другим напрямом, п'ятого відправника – по третьому напрямку. Такий варіант прийнятий з урахуванням побудови в подальшому мережі зв'язку (рис. 4.2).

Блок Буфер 2 призначений для прийому і зберігання повідомлень, переданих по каналах напрямків. Він складається з чотирьох буферів – для кожного напрямку свій буфер.

Алгоритм роботи буфера кожного з напрямів такої ж, як і алгоритм роботи буфера 1 (додаток В.2). Реалізується кожен з буферів також об'єктом *queue*.

Кожний елемент *buferNaprN* має ємність *emkBuferNaprN*. У полі «Дії при вході» введемо:

```
tekEmkNaprN + = entity.dlina.
```

У полі «Дії при виході»:

```
tekEmkNaprN - = entity.dlina, де N = {1, 2, 3, 4}.
```

Також як і для джерела повідомлень, для маршрутизатора потрібно створити виходи, через які відправляти повідомлення, і входи, за якими отримувати повідомлення.

Імітатор відмов обчислювального комплексу. Прийнято, що обчислювальний комплекс може виходити з ладу і відмовляти в обробці повідомлень. Можна було б побудувати імітатор відмов так, що генератор виробляє заявки-відмови у кількості, визначеній тривалістю часу моделювання. Однак ми зробимо так, що генератор виробляє одну заявку, а

потім цей процес повторюється через інтервали часу, що дорівнюють напрацюванню до чергової відмови. Задамо властивості компонентів блоку відповідно з даними табл.4.4.

Таблиця 4.4 – Властивості об’єктів блоку Імітатор відмов обчислювального комплексу

Властивість	source
Прибувають згідно	Інтенсивності
Інтенсивність прибуття	1
Кількість заявок, що прибувають за один раз	1
Обмежена кількість прибуттів	1
Максимальна кількість прибуттів	1
Властивість	розыгрыш_инт_до_отказа
Затримка задається	Певний час
Час затримки	<code>exponential (1/timeOtkBK)</code>
Місткість	1
Дії при виході	<code>hold.setBlocked (true); if (computer.size () != 0) { computer.remove ((Message) computer.get (0)); kolPoterBK ++;}</code>
Властивість	имитация_восст_ВК
Затримка задається	Певний час
Час затримки	<code>exponential (1/timeVosstBK)</code>
Місткість	1
Дії при виході	<code>hold.setBlocked (false)</code>

4.5 Моделювання каналу зв'язку

Даний сегмент призначений для імітації передачі повідомлень по каналах зв'язку. Для його реалізації в AnyLogic побудуємо імітаційну модель напрямку зв'язку, яка складається з основного і резервного каналів.

Напрямок зв'язку являє собою систему масового обслуговування разомкнутого типу з очікуванням і з відмовами через обмежену ємність вхідного буфера. А також з виходами з ладу (тимчасового нефункціонування) основного каналу. У моделі повідомлення слід подавати заявками, основний і резервний канал – одноканальними пристроями (ОКП), вхідний буфер (накопичувач) – чергою. У черзі слід використовувати дисципліну обслуговування FIFO.

4.5.1 Вихідні дані

В області перегляду агенту Канал додамо параметри і змінні необхідні для моделювання, призначення яких було розглянуто у п.4.2 (рис.4.10). Значення параметрів встановимо згідно табл. 4.5.

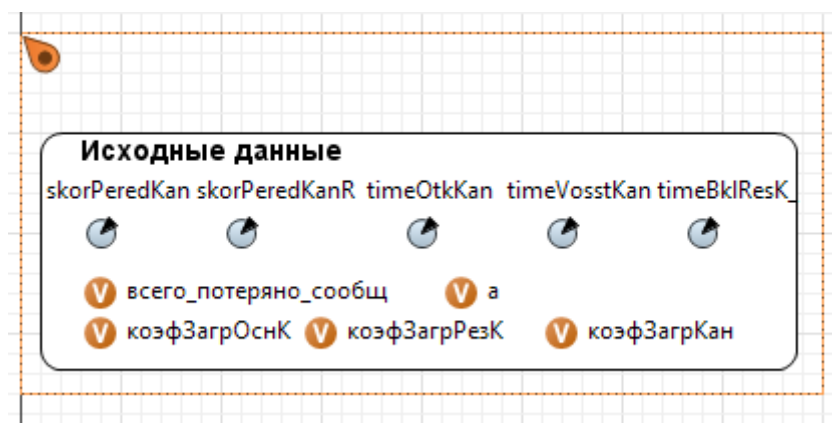


Рисунок 4.10 – Вихідні дані об'єкту Канал

Таблиця 4.5 – Значення властивостей параметрів моделювання об'єкту Канал

Ім'я	Тип	Значення за замовчуванням
skorPeredKan	double	5000
skorPeredKanR	double	5000
timeOtkKan	double	360
timeVosstKan	double	3.2
timeBklResK_	double	0.1
всього_потеряно_сообщ	int	0

4.5.2 Побудова подієвої частини об'єкту «Канал»

Об'єкти подієвої частини сегменту Канал наведені на рис. 4.11. Властивості об'єктів задані відповідно до табл. 4.6. Імітатор відмов основного каналу зв'язку побудований аналогічно імітаторові відмов обчислювального комплексу. Аналогічно були використані вихідні дані для моделювання цього блоку. Генератор виробляє одну заявку, а потім цей процес повторюється через інтервали часу, що дорівнюють напрацюванню до чергової відмови (табл.4.4).

Подієва частина призначена для імітації передачі пакетів повідомлень, виходу з ладу і відновлення каналу зв'язку, втрати пакетів, переданих по каналу в випадку виходу його з ладу.

Роботу ІМ об'єкта Канал, як і Абонента, можна розділити на 2 етапи. Перший – генерація сигналу присутності в обидві сторони, щоб мережеве обладнання зафіксувало наявність зв'язку на певному вході. У момент передачі сигналів про структуру затримки сигналів (повідомлень) немає, і розіграш поломки каналу не проводиться. Фіксація з'єднань на маршрутизаторах відбувається раніше, ніж передача сигналів присутності від абонентів. Другий етап – передача інформаційних пакетів з розіграшем поломок.

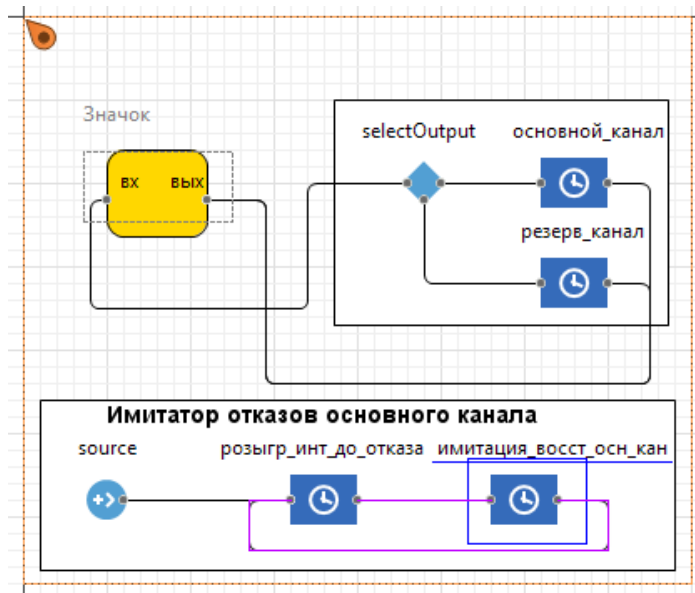


Рисунок 4.11 – Реалізація блоку Канал засобами AnyLogic

4.6 Запуск і налагодження моделі

Зараз, коли побудовані активні об'єкти Абонент, Канал і Маршрутизатор, приступимо до побудови моделі мережі зв'язку. Для цього, по-перше, створимо ще чотири типи агентів: Абонент2, Абонент3, Абонент4, Абонент5. Скопіюємо зміст агента абонент1 в інші. В коді треба внести зміни. У властивостях Абонент1 записаний код для розрахунку коефіцієнтів пропускної здатності абонентів 2...5 з абонентом 1. Тому, абонент 2 повинен мати код для розрахунку коефіцієнтів пропускної здатності абонентів 1,3...5 з абонентом 2 і так далі.

На main додамо п'ять об'єктів для абонентів, 11 об'єктів каналів зв'язку і два об'єкта для маршрутизаторів, як це показано на рис. 4.11

Виходи об'єктів абонент1...абонент5 з'єднані з відповідними входами об'єктів канал1...канал5. Виходи першого і другого абонентів з'єднані з вх1 маршрутизатора1, виходи третього і четвертого – з вх2, п'ятого – з вх3 об'єкта маршрутизатор1.

Таблиця 4.6 – Значення властивостей об'єктів подієвої частини сегмента

Канал

Властивості	selectOutput
Виход true обирається:	При виконанні умови
Умова:	основной_канал_работает
При виході (true):	entity.timePered=entity.dlina/skorPeredKan;
При виході (false):	<pre> if (a==0) entity.timePered=entity.dlina/skorPeredKanR; if (a==1) {entity.timePered=entity.dlina/skorPeredKanR + timeBklResK; a=0;} a=1; </pre>
Властивості	резерв_канал
Действия при выходе	<pre> коэфЗагр- РезК=резерв_канал.statsUtilization.mean(); коэфЗагрКан=коэфЗагрОснК+коэфЗагрРезК; </pre>
Властивості	розыг_инт_до_отказа
Затримка задається Час затримки	exponential(1/timeOtkKan)
Дії при виході:	<pre> основной_канал_работает = false; if (основной_канал.size() !=0) { Message m = основной_канал.get(0); основной_канал.stopDelayForAll(); всего_потеряно_сообщ ++; } </pre>
Властивості	имитация_восст_осн_кан
Затримка задається Час затримки	<p>Визначений час</p> <pre> exponential(1/timeVosstkKan) </pre>
Дії при виході:	основной_канал_работает = true;

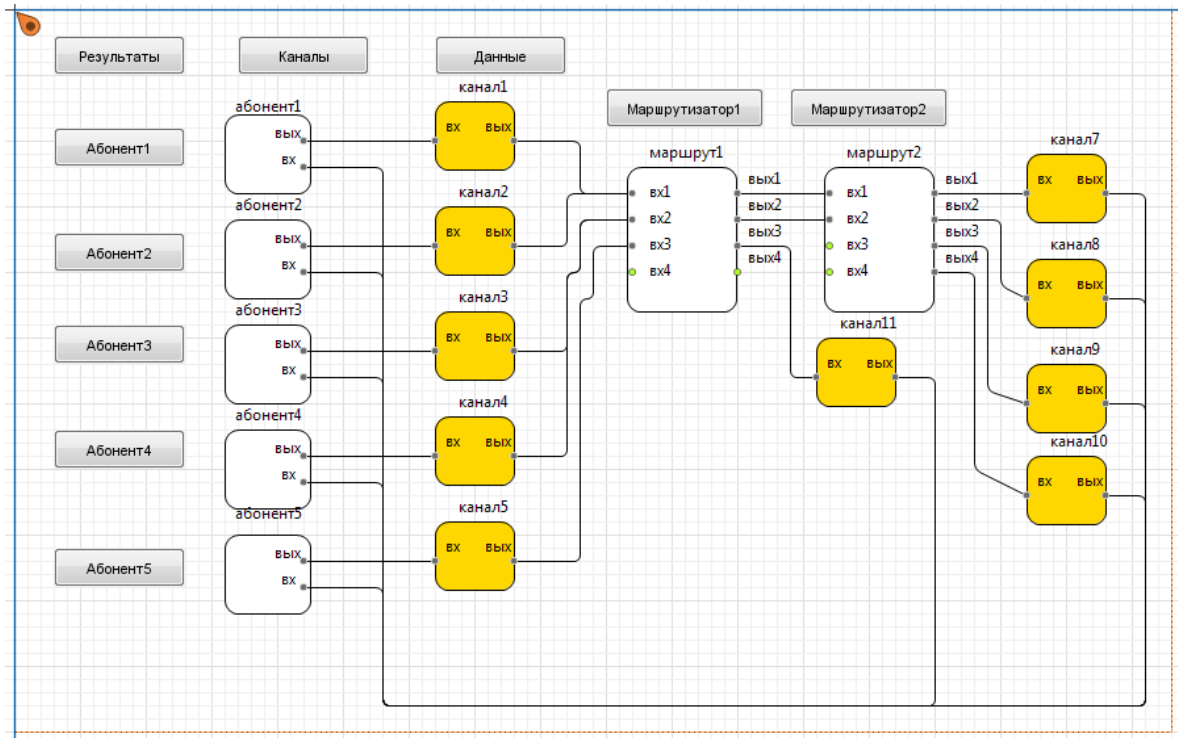


Рисунок 4.11 – Элементы модели функционирования сети зв'язку

Для того щоб зв'язок був між усіма абонентами і вони могли б обмінюватися повідомленнями, потрібен ще один маршрутизатор. Для цього не можна використовувати другий примірник, тому що програмно він налаштований саме на власний варіант організації зв'язку. Тому необхідно було створити ще один тип агента Маршрутизатор1, у який були скопійовані об'єкти Маршрутизатора і додані зміни у код. Так елементу exit у полі «Дії При виході» доданий наступний код:

```
int i;
i=entity.numAbPol;
{
switch (i) {
case 1:if (emkBuferNap1-tekEmkNap1>=entity.dlina)
{enter1.take(entity); break;}
else {enter.take(entity); break;}
case 2:if (emkBuferNap2-tekEmkNap2>=entity.dlina)
{enter2.take(entity); break;}
else {enter.take(entity); break;}
```

```

case 3:if (emkBuferNapr3-tekEmkNapr3>=entity.dlina)
    {enter3.take(entity); break;}
    else {enter.take(entity); break;}
case 4:if (emkBuferNapr4-tekEmkNapr4>=entity.dlina)
    {enter4.take(entity); break;}
    else {enter.take(entity); break;}
} }

```

Таким чином Маршрутизатор1 налаштований так, що повідомлення від абонентів 1...4 будуть направлятися на його виходи 1...4. Усі інші зв'язки встановлені так, як це показано на рис. 4.11.

В моделі була налаштована можливість перемикання між різними областями перегляду таким чином, щоб можна було переходити до любого абоненту, каналу, маршрутизатору і назад. У кожному активному об'єкті – до даних і назад до подієвої частини об'єкту або до мережі.

Для перемикання був використаний елемент button з палітри Елементи управління. Перемикання організовано за допомогою методу navigateTo().

Для запуску моделі виконаємо наступні налаштування: встановимо одиниці модельного часу – секунди, режим виконання – віртуальний час (максимальна швидкість), зупинити в заданий час (кінцевий час – 3600000). Час моделювання збільшено в 1000 разів по числу прогонів моделі.

При запуску моделі були отримані результати моделювання наведені на рис. 4.12.

Отримані наступні показники якості обслуговування мережі зв'язку: коефіцієнт пропускної здатності 0.816 і середній час передачі одного повідомлення 5.21. Коефіцієнт пропускної здатності, наприклад, абонент 2 – абонент 3 дорівнює 0.814. Максимальний час передачі повідомлень дорівнює 35.8, мінімальний час – 0.77. Велика розбіжність і різниця між мінімальним і максимально часом передачі повідомлення пояснюється прийнятим експоненціальним законом розподілу часу передачі повідомлень.

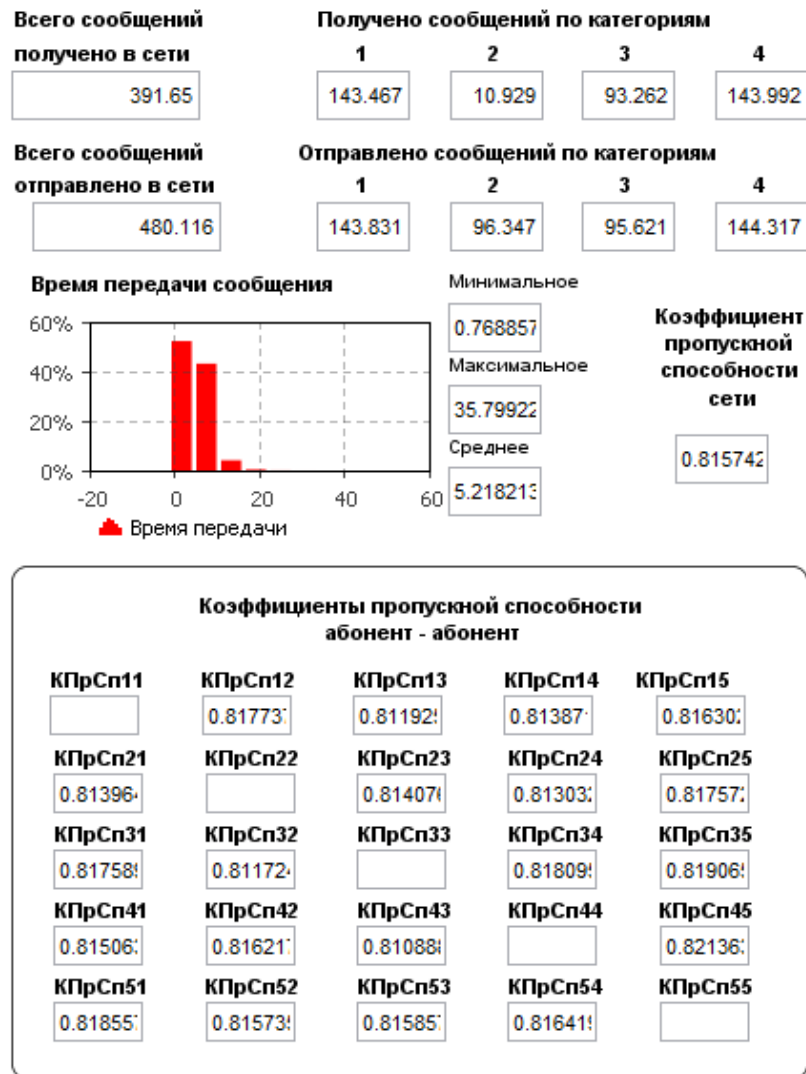


Рисунок 4.12 – Результаты моделирования сети зв'язку

Кількість відправлених та отриманих повідомлень всього та за категоріям розрахована за один прогін моделі, тобто за 3600 сек.

Для розрахунку був введений параметр $kolProg = 1000$ і в 1000 разів було збільшено модельний час. Кількість відправлених повідомлень 480.116, отримано всього абонентами – 391.65. Якщо розділити число отриманих повідомлень на число відправлених, то і буде отриманий коефіцієнт пропускної здатності мережі. Точно такі ж результати моделювання, коефіцієнти пропускної здатності, виводяться і по кожному абоненту мережі.

Результати моделювання по абоненту 2 мережі зв'язку представлені на рис. 4.13.



Рисунок 4.13 – Результати моделювання мережі зв'язку по Абоненту2

4.7 Аналіз результатів моделювання

Після налаштування моделі мережі зв'язку було виконано декілька експериментів. Результати моделювання наведені у табл. 4.7. Параметри мережі зв'язку в кожному наступному експерименті залишаються такими ж, якщо не вказано їх зміна. У четвертому експерименті в зв'язку зі зменшенням ємкості вхідних буферів ВК1 і ВК2 повідомлення другої категорії не передаються, так як їх середня довжина з усіх категорій максимальна. У третьому і четвертому експериментах крім зазначених параметрів були також змінені середні інтервали надходження повідомлень від всіх абонентів (timeAbonent) з 30 до 20.

Таблиця 4.7 – Показники функціонування мережі зв'язку

Показники	Значення
Експеримент 1: згідно постановки завдання	
Коефіцієнт пропускної здатності мережі зв'язку	0,816
Час передачі одного повідомлення	5,218
Коефіцієнт загрузки ОК1	0,255
Коефіцієнт загрузки ОК2	0,170
Коефіцієнт загрузки основних каналів	0,042
Кількість отриманих/відправлених пакетів	391,65/480,116
Експеримент 2: ємність вхідного буферу 600000	
Коефіцієнт пропускної здатності мережі зв'язку	0,816
Час передачі одного повідомлення	5,218
Коефіцієнт загрузки ОК1	0,255
Коефіцієнт загрузки ОК2	0,170
Коефіцієнт загрузки основних каналів	0,042
Кількість отриманих/відправлених пакетів	391,65/480,116
Експеримент 3: ємність буферів усіх напрямків 70000	
Коефіцієнт пропускної здатності мережі зв'язку	0,741
Час передачі одного повідомлення	4,974
Коефіцієнт загрузки ОК1	0,255
Коефіцієнт загрузки ОК2	0,170
Коефіцієнт загрузки основних	0,042

каналів		
Кількість відправлених пакетів	отриманих/	355,006/479,205
Експеримент 4: ємність буферів усіх напрямків 60000		
Коефіцієнт пропускної здатності мережі зв'язку		0,5999
Час передачі одного повідомлення		4,823
Коефіцієнт загрузки ОК1		0,255
Коефіцієнт загрузки ОК2		0,170
Коефіцієнт загрузки основних каналів		0,042
Кількість відправлених пакетів	отриманих/	288,441/480,763

Як це можна побачити з наведених експериментів при збільшенні буферу напрямків маршрутизаторів збільшується пропускна здатність мережі. На рис. 4.15 видно, що при ємності буфера близько 40000 байтів працездатність мережі порушається. Це пов'язано з тим, що така ємність занадто маленька для збереження пакетів, нагадаємо що довжина пакетів різних категорій складає 53000, 86000, 66000 і 50000 байтів відповідно. Прийнятна пропускна здатність досягається при ємності буфера напрямків від 100000 байтів і вище.

Результати моделювання дають можливість простежити діапазони змін важливих мережевих параметрів і передбачити негативні зміни якості обслуговування мережі. Так для підвищення пропускної здатності мережі, модель якої була побудована, оптимальними значеннями ємності вхідних буферів є 5000000 байтів, буферів вихідних напрямків 100000 байтів. Незначне підвищення пропускної здатності можна спостерігати при зменшенні інтервалу надходження повідомлень від абонентів. При ємності

буферів напрямків 80000 байтів повідомлення другої категорії не передаються, так як їх середня довжина з усіх категорій максимальна.

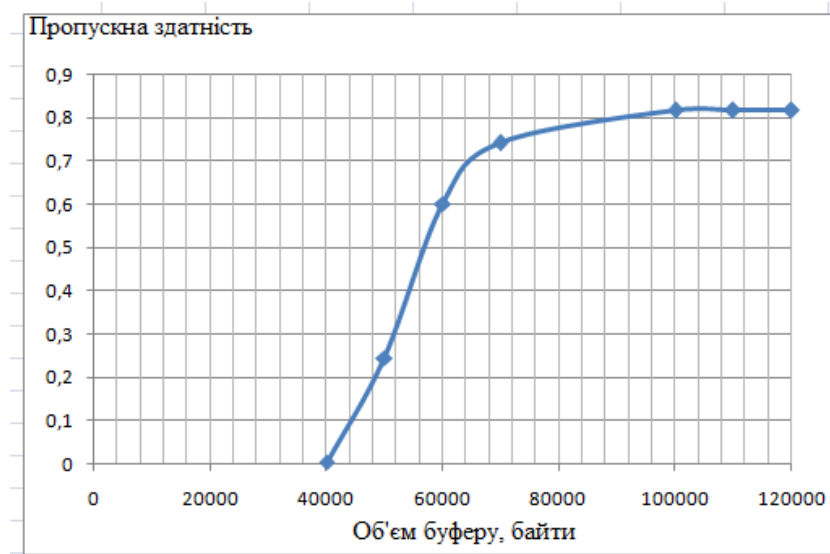


Рисунок 4.15 – Графік залежності коефіцієнту пропускної здатності мережі від ємності буферів напрямків маршрутизаторів

Найвища пропускна здатність, яка досягається за результатами експериментів при змінах вищезазначених параметрів складає 0.815.

ВИСНОВКИ

В магістерській роботі була розроблена імітаційна модель функціонування мережі зв'язку, яка дозволяє дослідити вплив ємностей буферів, інтервалів часу надходження повідомлень, їх обчислювальних потужностей і інших параметрів на показники функціонування мережі з метою їх оцінки та прийняття рішень при необхідності щодо поліпшення якості обслуговування мережі.

Особливістю даної моделі є те, що вона була створена з активних об'єктів – агентів, що моделюють роботу наступних компонентів мережі: абонента, маршрутизатора, каналу. Тобто використовуючи ці об'єкти можна будувати імітаційні моделі функціонування мережі зв'язку довільної топології. Сегменти реалізовані засобами пакету AnyLogic на мові програмування Java.

За допомогою наведеної методики розробки імітаційної моделі функціонування мережі зв'язку можуть бути вирішені наступні завдання:

- підвищення рівня автоматизації виробництва, націлене на підвищення продуктивності;
- оцінка і аналіз роботи мережі у широкому діапазоні варійованих параметрів за декілька хвилин, без виведення з експлуатації діючого обладнання;
- можливість визначення мінімально необхідного обладнання, що забезпечує потреби передачі, обробки та збереження інформації;
- оцінка необхідного запасу продуктивності обладнання, що забезпечує можливе збільшення виробничих потреб у найближчий час ;
- вибір декількох варіантів обладнання з урахуванням поточних потреб, перспективи розвитку на підставі критерію вартості обладнання;
- проведення перевірки працездатності обчислювальної системи, складеної з рекомендованого обладнання.

ПЕРЕЛІК ПОСИЛАНЬ

1. Боголюбов А.Н. Основы математического моделирования. Учеб. пособие. М.: МГУ, 2003. 137с.
2. Введение в математическое моделирование. Учеб. пособие. Логос, 2004. 440с.
3. Аверченков В.И., Федоров В.П., Хейфец М.Л. Основы математического моделирования технических систем. Учеб. пособие. М: Изд-во «Флинта», 2011. 271с.
4. Бусленко, Н.П. Метод статистического моделирования. М.: Статистика, 1970. 112 с.
5. Сидоренко, В.Н. Системная динамика. М.: Эконом. факультет МГУ, ТЕИС, 1998. 200 с.
6. AnyLogic. Навчальний посібник з системної динаміки URL: <http://www.xjtek.com> (Дата звернення 12.11.2020)
7. Ивашкин, Ю.А. Агентные технологии и мультиагентное моделирование систем. Учеб. пособие. М.: МФТИ, 2013. 268с.
8. Абрамова, Н.А., Авдеева З.К. Когнитивный анализ и управление развитием ситуаций: проблемы методологии, теории и практики . Проблемы управления. 2008. № 3. С. 85-87
9. Болотова Л.С. Системы искусственного интеллекта: модели и технологии, основанные на знаниях. М.: Финансы и статистика, 2012. 664 с.
10. Брауде, Э. Технология разработки программного обеспечения. СПб.: Питер, 2004. 655 с.
11. Черемных, С.В., Семенов И. О., Ручкин В.С. Структурный анализ систем: IDEF – технологии. М.: Финансы и статистика, 2003. 208 с.
12. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. 4–е изд. СПб.: Питер, 2010. 944 с.: ил.

13.Офіційний сайт XJ Technologies URL: <http://www.xjtek.com> (Дата звернення 12.11.2020)

14.Java For AnyLogic Users URL: http://www.xjtek.com/files/book/Java_for_AnyLogic_users.pdf (Дата звернення 12.11.2020)

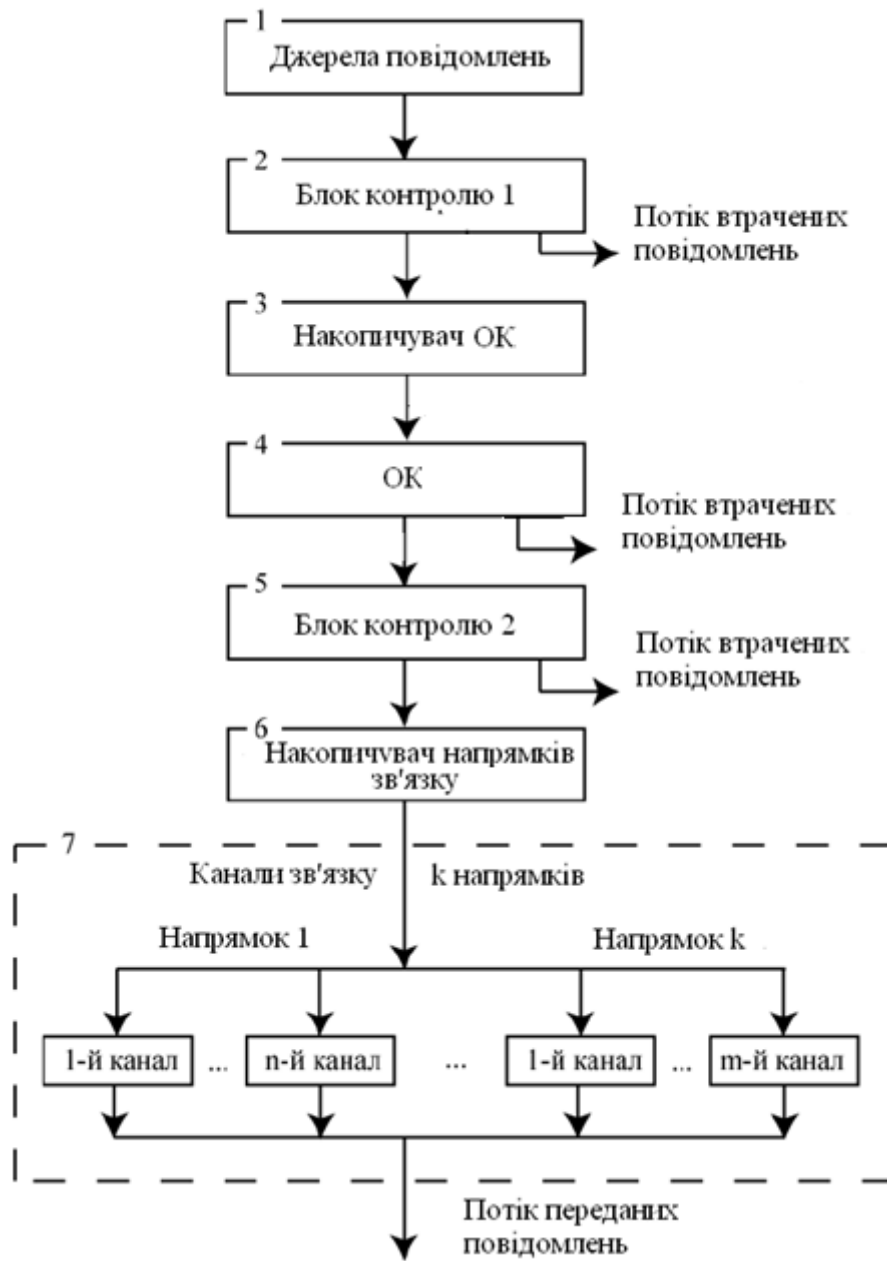
15.Карпов, Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб.: БХВ-Петербург, 2005. 400 с.

16.Борщев А.В. Практическое агентное моделирование и его место в арсенале аналитика URL: <http://www.gpss.ru/immod05/p/borshev/> (Дата звернення 12.11.2020)

Д О Д А Т К И

Додаток А

Загальна схема роботи маршрутизатора



Додаток Б
Алгоритм блоку Абонент



Додаток В

Алгоритми роботи блоків маршрутизатора

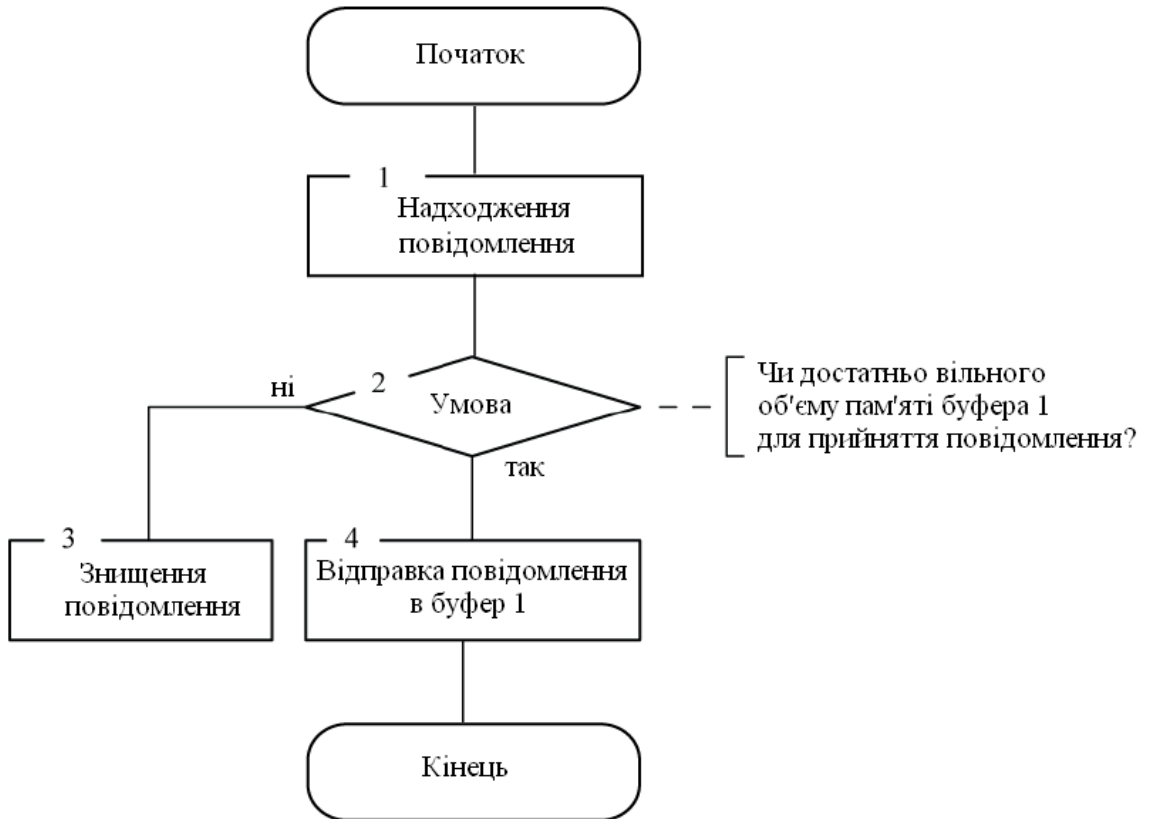


Рисунок В.1 – Алгоритм роботи блоку контролю 1

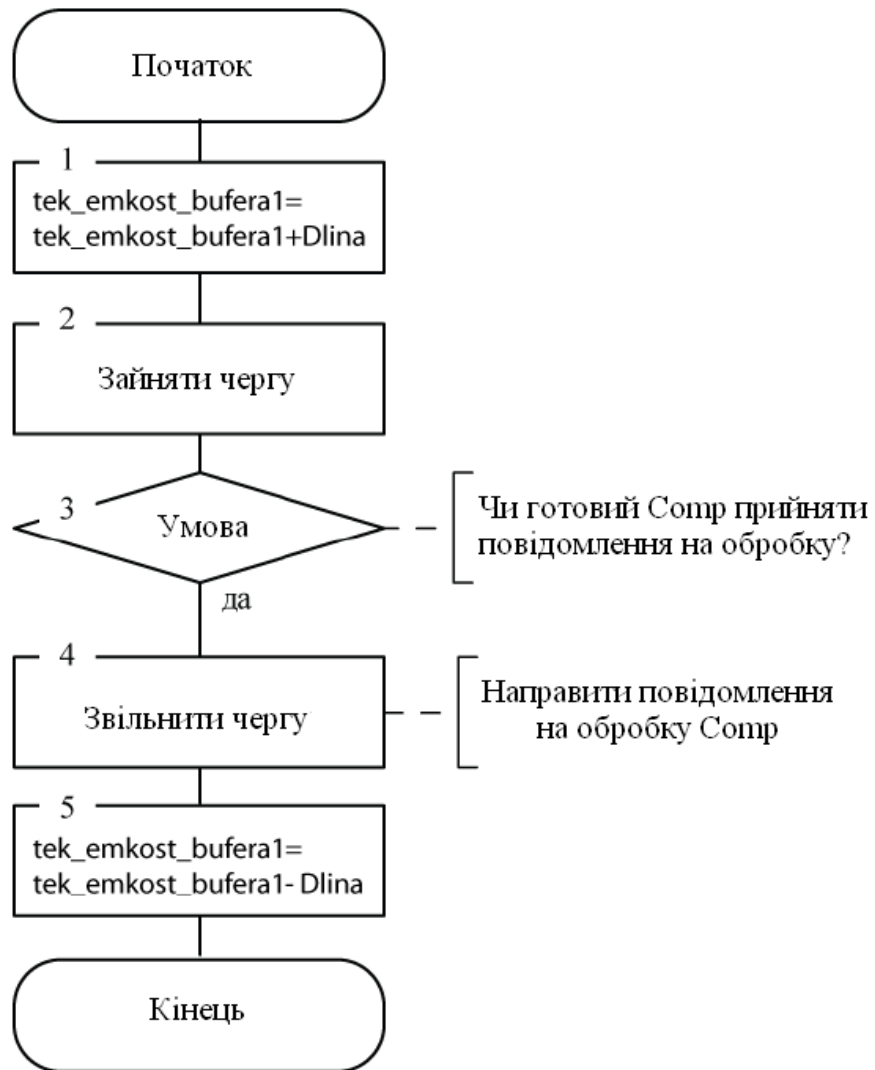


Рисунок В.2 – Алгоритм роботи Блоку буфер 1



Рисунок В.3 – Алгоритм роботи Блоку обробки повідомлення



В.4 – Алгоритм роботи Блоку контролю 2

Додаток Г
Програмний код класу Message

```
/**
 * Message
 */
public class Message extends
com.anylogic.libraries.enterprise.Entity implements Serializable
{

    int numAbOtp;
    int numAbPol;
    int numKat;
    double timeOtp;
    double dlina;
    double timeObr;
    double timePered;

    /**
     * Конструктор по умолчанию
     */
    public Message() {
    }

    /**
     * Конструктор, инициализирующий поля
     */
    public Message(int numAbOtp, int numAbPol, int numKat,
double timeOtp, double dlina, double timeObr, double timePered)
    {

        this.numAbOtp = numAbOtp;
        this.numAbPol = numAbPol;
        this.numKat = numKat;
        this.timeOtp = timeOtp;
        this.dlina = dlina;
        this.timeObr = timeObr;
        this.timePered = timePered;
    }

    @Override
    public String toString() {
        return
            "numAbOtp = " + numAbOtp + " " +
            "numAbPol = " + numAbPol + " " +
            "numKat = " + numKat + " " +
            "timeOtp = " + timeOtp + " " +
            "dlina = " + dlina + " " +
            "timeObr = " + timeObr + " " +
            "timePered = " + timePered + " ";
    }
}
```



```
/**
 * Это число используется при сохранении состояния
 модели<br>
 * Его рекомендуется изменить в случае изменения класса
 */
private static final long serialVersionUID = 1L;
}
```