

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Тернопільський національний економічний університет  
Навчально-науковий інститут інноваційних освітніх технологій  
Кафедра комп'ютерної інженерії

**БОЙКІВ Назар Ігорович**

**Імітаційна модель обслуговування магазину  
комп'ютерної техніки / Imitation model of the  
computer equipment store maintenance**

спеціальність: 123 - Комп'ютерна інженерія  
магістерська програма - Комп'ютерна інженерія

Магістерська робота

Виконав студент групи КІм-21  
Н.І. БОЙКІВ

Науковий керівник:  
к.т.н., Л.О. Дубчак

Магістерську роботу допущено до захисту:

ТЕРНОПІЛЬ -2018

## ЗМІСТ

Вступ.....	3
1 Сучасні системи масового обслуговування.....	5
1.1 Застосування сучасних систем масового обслуговування.....	5
1.2 Типи систем масового обслуговування.....	11
1.3 Постановка задачі при моделюванні СМО .....	15
2 Імітаційне моделювання систем масового обслуговування.....	19
2.1 Опис алгоритму та будови розробленої моделі системи масового обслуговування.....	19
2.2 Практичні особливості імітаційного моделювання.....	24
2.3 Simulink як середовище імітаційного моделювання.....	33
3 Практична реалізація поставленого завдання.....	36
3.1 Опис розроблювальної схеми та її складових.....	36
3.2 Реалізація блоків СМО в Simulink .....	40
3.3 Тестування, верифікація та захист моделі.....	47
Висновки.....	67
Список використаних джерел.....	68

## ВСТУП

У багатьох сферах практичної діяльності люди стикаються з необхідністю в обслуговування. Подібні ситуації виникають у багатьох закладах та установах, наприклад, в касах у великих аеропортах, коли обслуговуючий персонал авіаційного складу чекає на посадку або висадку, на телефонних станціях, що очікують очищення абонентської лінії, в ремонтних майстернях, коли клієнти очікують на ремонт машин та обладнання, на складах організації розвантаження або завантаження транспортних засобів. У всіх цих випадках фігурує явище масового обслуговування. Вивчення таких ситуацій займається теорія систем масового обслуговування.

У теорії систем обслуговування (QS) обслуговуваний об'єкт називається вимогою. Взагалі, попит, як правило, розуміється як запит на задоволення певної потреби, наприклад, розмова з клієнтом, посадка літака, покупка продукції або завантаження матеріалів на складі.

Послуги, що обслуговують вимоги, називаються сервісними пристроями або сервісними каналами. Наприклад, це телефонні канали, посадочна смуга, майстри-ремонтники, квиткові касири та точки завантаження та розвантаження на базах і складах.

У теорії QS такі випадки розглядаються, коли отримання вимог відбувається з випадковими інтервалами часу, а тривалість обслуговування вимог не є постійною, тобто є випадковою.

Основним завданням теорії QS є вивчення режиму роботи сервісної системи та вивчення явищ, що виникають під час процесу технічного обслуговування. Отже, однією з характеристик сервісної системи є час, коли клієнт перебуває в черзі. Очевидно, що цей час можна зменшити, збільшивши кількість обслуговуючого пристрою. Однак кожен додатковий пристрій вимагає певної вартості, а час простою сервісного пристрою зростає через відсутність вимог до технічного обслуговування, що також є

негативним явищем. Отже, в теорії QS виникають проблеми оптимізації: як досягти певного рівня сервісу (максимально зменшуючи чергу або втрату вимог) за мінімальну вартість, пов'язану з простою обслуговування пристроїв.

Загалом, моделі QS дуже поширені і використовуються у багатьох сферах людської діяльності, а також у комп'ютеризації. Моделі QS зручні для опису окремих сучасних обчислювальних систем, таких як процесор – жорсткий диск, вхідний-вихідний канал і т. д. Обчислювальна система в цілому являє собою сукупність взаємопов'язаних систем. Наприклад: програма для вирішення певного завдання проходить кілька етапів обробки, доступ до зовнішніх пристроїв зберігання даних на пристроїв введення/виведення. Після виконання послідовності таких дій додаток вважається обслуженим, і він залишає систему.

Метою розробки цього дипломного проекту є імітаційна модель системи масового обслуговування магазину комп'ютерної техніки за допомогою Matlab Simulink.

# 1 СУЧАСНІ СИСТЕМИ МАСОВОГО ОБСЛУГОВУВАННЯ

## 1.1 Застосування сучасних систем масового обслуговування

Під системою масового обслуговування (СМО) розуміють динамічну систему, призначену для ефективного обслуговування потоку заявок (вимог на обслуговування) при обмеженнях на ресурси системи.

Моделі СМО зручні для опису окремих підсистем сучасних обчислювальних систем, таких як підсистема-процесор – основна пам'ять, канал введення-виведення і т. д. Обчислювальна система в цілому являє собою сукупність взаємопов'язаних підсистем, взаємодія яких носить імовірнісний характер. Заявка на рішення деякої задачі, що надходить в обчислювальну систему, проходить послідовність етапів обслуговування, звернення до зовнішніх запам'ятовуючих пристроїв і пристроїв введення-виведення. Після виконання деякої послідовності таких етапів, число і тривалість яких залежить від трудомісткості програми, заявка вважається обслуженою і залишає обчислювальну систему. Таким чином, обчислювальну систему в цілому можна представляти сукупністю СМО, кожна з яких відображає процес функціонування окремого пристрою або групи однотипних пристроїв, що входять до складу системи.

У багатьох сферах практичної діяльності люди стикаються з необхідністю бути в очікуванні. Подібні ситуації виникають на складах, в касах у великих аеропортах, коли обслуговуючий персонал авіаційного складу чекає на посадку або висадку, на телефонних станціях, що очікують очищення абонентської лінії, в ремонтних майстернях, коли очікують на ремонт машин та обладнання, на складах організації розвантаження або завантаження транспортних засобів. У всіх цих випадках ми маємо справу з масовим сервісом. Вивченням таких ситуацій займається теорія систем масового обслуговування.

У теорії систем обслуговування (QS) обслуговуваний об'єкт називається вимогою. Взагалі, попит, як правило, розуміється як запит на задоволення певної потреби, наприклад, розмова з клієнтом, посадка літака, покупка продукції або завантаження матеріалів на складі.

Послуги, що обслуговують вимоги, називаються сервісними пристроями або сервісними каналами. Наприклад, це телефонні канали, посадочна смуга, майстри-ремонтники, квиткові касири та точки завантаження та розвантаження на базах і складах.

У теорії QS такі випадки розглядаються, коли отримання вимог відбувається з випадковими інтервалами часу, а тривалість обслуговування вимог не є постійною, тобто є випадковою.

Основним завданням теорії QS є вивчення режиму роботи сервісної системи та вивчення явищ, що виникають під час процесу технічного обслуговування. Отже, однією з характеристик сервісної системи є час, коли клієнт перебуває в черзі. Очевидно, що цей час можна зменшити, збільшивши кількість обслуговуючих пристроїв. Однак кожен додатковий пристрій вимагає певної вартості, а час простою сервісного пристрою зростає через відсутність вимог до технічного обслуговування, що також є негативним явищем. Отже, в теорії QS виникають проблеми оптимізації: як досягти певного рівня сервісу (максимально зменшуючи чергу або втрату вимог) за мінімальну вартість, пов'язану з простою обслуговування пристроїв.

Загалом, моделі QS дуже поширені і використовуються у багатьох сферах людської діяльності, а також у комп'ютеризації. Моделі QS зручні для опису окремих сучасних обчислювальних систем, таких як процесор – жорсткий диск, вхідний-вихідний канал і т. д. Обчислювальна система в цілому являє собою сукупність взаємопов'язаних систем. Наприклад: програма для вирішення певного завдання проходить кілька етапів обробки, доступ до зовнішніх пристроїв зберігання даних та пристроїв та пристроїв

введення / виведення. Після виконання послідовності таких дій додаток вважається поданим, і він залишає систему.

Сукупність взаємопов'язаних СМО називається мережею масового обслуговування (стохастичною мережею).

Обслуговування – це затримка програми протягом певного фіксованого часу в обслуговуючому пристрої.

Тривалість обслуговування – час затримки заявки в приладі. Дуже часто є одним із основних параметрів вимог, які надходять в СМО.

Зберігач – це пристрій, в якому вимоги зберігаються перед доступом до каналу обслуговування. Кількість місць вважається основним параметром потужності пристрою.

Заява, прийнята до СМО, може бути у двох різних станах:

1. в стані обслуговування, тобто в пристрої обслуговування;
2. в режимі очікування (на пристрої зберігання). Це відбувається, якщо всі пристрої обслуговують одночасно.

Черга вимог формує заявки, що знаходяться в накопичувачі в очікуванні обслуговування. У цьому випадку службова дисципліна є умовою вибору вимоги з черги для обслуговування в сервісному каналі. Вхідний потік вимог визначає послідовність моментів отримання вимог до послуг та вказує кількість таких вимог у кожному регулярному доході. Для того, щоб описати вхідний потік додатків, необхідно встановити закон розподілу вірогідності, який регулює порядок прийому вимог до послуг та визначає кількість таких вимог у кожному наступному пристрої. Наприклад, вимоги в бібліотеці або податковій службі можна отримати в середньому кожні 4 хвилини. У той же час в бібліотеці кожного разу існує одна вимога (клієнти приходять в бібліотеку один за іншим), а в умовах служби таксі можуть отримувати як індивідуальні, так і групові вимоги (пасажери можуть їхати по одному або з компанією).

Управління чергою визначається як сукупність принципів, спрямованих на управління потоком клієнтів та спрощення процесу

обслуговування. Крім цього, управління чергою – це не одинарний процес. Для того, щоб отримати результати, управління чергою повинно бути безперервним процесом .

Величезна кількість відвідувачів – це апріорі добре для будь-якого бізнесу. Це дуже багато потенційних продажів, чи надання послуг, а це не що інше як прямий результат усього процесу обслуговування та сенс створення систем обслуговування.

Проблема виникає, коли приплив клієнтів перевищує можливості працівників. Чим менше діючих службовців (каналів обслуговування), тим менша кількість клієнтів, якими вони можуть керувати.

Залишившись самостійно, службовці не можуть не бути перевантаженими, коли стикаються з завеликою кількістю відвідувачів. Уникнути цього вони можуть тільки тоді, коли вони оснащені інструментами або методами управління чергою.

Управління чергою його передусім стосується областей черги, тобто місць, де відбуваються черги. Але що саме така черга? Можна було б сказати, що це ряд людей, які чекають надання послуг. Але також це поняття можна сформулювати як застосування "проблеми, що перевищує пропозицію".

Як сказано вище, приплив споживачів (вимог) перевищує можливості працівників (каналів обслуговування).

Управління чергою – це те, що допомагає зберегти цю крихку рівновагу.

Спочатку можна подумати, що системи черги є такими, що працюють тільки в роздрібній торгівлі, де найважливіша кількість клієнтів, які треба обслужити обслуговувати в обмежений час. Тим не менш, управління натовпом – це питання, яке вимагає рішення в багатьох інших галузях – від охорони здоров'я до державного управління чи освіти.

Теорія чергування – це не що інше математичне вивчення черг очікування. Модель черги побудована так, щоб можна було передбачити тривалість черги та час очікування. Теорія чергування загалом розглядається



як галузь досліджень операцій, оскільки результати часто використовуються при прийнятті ділових рішень про ресурси, необхідних для надання послуг.

Мережі черг – це системи, в яких ряд черг пов'язаний тим, що називається маршрутизацією клієнтів.

Найпростіша нетривіальна мережа черг називається тандемними чергами. Перші значні результати в цій області були мережами Джексона, в яких є ефективний стаціонарний розподіл продукту форми, аналіз середнього значення, який виводить середні показники, такі як пропускна здатність чи тимчасове перебування виомги в черзі, щоб бути обчислено. Якщо загальна кількість клієнтів в мережі залишається незмінною, мережа називається замкненою мережею.

Було також досліджено мережі клієнтів, мережі Kelly, де клієнти різних класів мають різні пріоритетні рівні в різних сервісних вузлах.<sup>[31]</sup> Інший тип мережі – це G-мережі, які вперше були запропоновані Еролом Геленбе в 1993 році: ці мережі не приймають експоненціальні розподіли часу, подібні до класичної мережі Джексона.

У дискретних часових мережах, де існує обмеження на те, які сервісні вузли можуть бути активними в будь-який час, алгоритм планування максимальної ваги вибирає політику обслуговування, щоб забезпечити оптимальну пропускну здатність у тому випадку, коли кожна вимога відвідує лише один службовий вузол. У більш загальному випадку, коли виомги можуть відвідати більше одного вузла, маршрутизація зворотного потоку забезпечує оптимальну пропускну здатність таких вимог. План системи обслуговування повинен мати алгоритм формування черги, оскільки це впливає на роботу моделі в цілому.

Характеристика потоку вимог в математичному моделюванні позначається символом  $\lambda$ .

$\lambda$  – інтенсивність вхідних в систему замовлень, тобто середня кількість замовлень, що надходять в систему за одиницю часу.

Потік вимог є правильним, якщо замовлення потрапляють до системи по одному за певний проміжок часу. Наприклад, потік вимог до конвеєра при течії молока (при постійній швидкості руху) є регулярним.

Потік вимог називається стаціонарним, якщо його імовірнісні характеристики не залежать від часу. Зокрема, інтенсивність стаціонарного потоку є постійною  $\lambda(t) = \lambda$ . Наприклад, рух автомобілів на світлофорі не є стаціонарним протягом дня, але його можна вважати стаціонарним у години пік.

Потік вимог називається потоком без наслідків, якщо за будь-які два часові інтервали, які не перекриваються, кількість замовлень, що надходять в систему за цими інтервалами, не залежить від кількості замовлень, отриманих в інших часових інтервалах. Наприклад, потік підземних пасажирів практично не має наслідків.

Потік замовлень називається звичайним, якщо події настають одинарному порядку, а не в групах. Наприклад, потік поїздів до метро є звичайним, в той час як потік автомобілів не є звичайним.

Потік замовлень називається найпростішим (або стаціонарним Пуассоном), якщо вони або стаціонарні, або звичайні, або не мають наслідків.

Моделювання – найефективніший спосіб дослідження складних систем різних цілей – технічних, економічних, екологічних, соціальних, інформаційних – як на етапі їх проектування, так і в процесі експлуатації. Можливості систем моделювання далеко не вичерпані, тому нові методи та технології моделювання постійно з'являються.

Створення моделі – це кропіткий та творчий процес, який вимагає, щоб дослідник не тільки мав глибокі теоретичні знання різних математичних та технічних дисциплін, але і творчий підхід до вирішення завдань, можливість генерувати певні евристики, що мають відношення до глибинної сутності досліджуваного об'єкту. Моделювання як метод пізнання використовувався ще з давніх часів. Але з появою комп'ютера симуляція систем збагатилася

появою принципово нових методів моделювання, таких як: моделювання, еволюційне моделювання, методи групового розгляду аргументів. Моделі та методи моделювання використовуються для створення автоматизованих систем проектування, систем прийняття рішень, автоматизованих систем управління, систем штучного інтелекту. Необхідність вирішення завдань системного моделювання виникає не тільки у вченого, але й у дизайнера, виробника, ділової людини в повсякденній роботі.

Сучасна технологія моделювання не тільки полегшила і прискорила процес проектування та дослідження моделі, але також значно спростила методи спеціаліста в системах моделювання та фахівця, що працює в області моделювання. Результати моделювання, представлені інструментами 3D-анімації, допомагають знайти загальну мову та взаєморозуміння між фахівцями системного дизайну та фахівцями галузі.

Метою дипломної роботи є моделювання системи масового обслуговування. Розробка моделі магазину, на вхід якої заявки надходять випадково із середньою швидкістю. Замовлення магазину також обслуговуються за певної середньої швидкості.

Для досягнення мети в дипломній роботі вирішуються наступні завдання:

1. формування цілей і завдань дослідження, а також аналіз об'єкта дослідження;
2. вибір програмного забезпечення для представлення моделі;
3. реалізація моделі – найбільш відповідальний етап моделювання системи.

Побудувати модель – лише частина роботи дослідника, щоб мати можливість отримати результати симуляції. Мистецтво дослідника полягає саме в тому, щоб отримати в процесі моделювання корисні з точки зору всього моделювання результати.

Математична модель системи масового обслуговування включає такі елементи:

1. вхідний потік вимог, що надходять на обслуговування;
2. черга, яка складається з вимог, що очікують на обслуговування;
3. система обслуговування;
4. вихідні потоки обслужених, втрачених вимог та вимог, що надходять на повторне обслуговування;
5. характеристики якості системи;
6. механізм (дисципліна) обслуговування.

Типовим методом вивчення складних систем на першому етапі є імітація їх на комп'ютері. В результаті моделювання отримуються залежності, які характеризують вплив структури та параметрів системи на її ефективність, надійність та інші властивості. Ці залежності використовуються для отримання оптимальної структури та системних параметрів.

Модель, сформульована на мові математики з використанням математичних методів, називається математичною моделлю. Моделювання характеризується відтворенням явищ, описаних математичною моделлю, з збереженням їх логічної структури, послідовністю чергування в часі. Будь-яка відповідна інформація, що циркулює в моделі, може бути використана для оцінки потрібних значень, якщо вона доступна лише для реєстрації та подальшої обробки.

Необхідні значення при вивченні процесів методом імітації зазвичай визначаються як середні значення за великою кількістю реалізацій процесу. Якщо кількість реалізацій  $N$ , що використовується для оцінки невідомих величин, досить велика, то за законом великих чисел отримані оцінки набувають статистичної стійкості і з достатньою точністю для практичних цілей можна вважати приблизними значеннями невідомі кількості.

Сутність методу моделювання стосовно завдань масового обслуговування полягає в наступному.

Побудовано алгоритми, за допомогою яких можна генерувати випадкові реалізації визначених потоків однорідних подій, а також

моделювати процеси функціонування систем обслуговування. Ці алгоритми використовуються для багаторазового відтворення реалізації випадкового процесу обслуговування в умовах фіксованого завдання. Отримана інформація про стан процесу піддається статистичній обробці для оцінки значень, які є показниками якості обслуговування.

При вивченні складних систем методом імітаційного моделювання значна увага приділяється розгляду випадкових чинників. Випадкові математичні події, випадкові величини та випадкові процеси (функції) використовуються як математичні схеми, що використовуються для формалізації дії цих факторів. Формування реалізацій випадкових об'єктів будь-якої природи на комп'ютері зводиться до розробки та перетворення випадкових чисел.

Системи чергування (QS) – це системи певного типу. Основою QS є певна кількість сервісних пристроїв – сервісні канали. Роль каналів насправді може виконуватися пристроями, операторами, постачальниками, лініями зв'язку тощо.

Мета QS полягає в обслуговуванні потоку додатків (вимог), що представляють собою послідовність подій, що надходять нерівномірно та в невідомих і випадкових точках за часом. Сама послуга обслуговування також непостійна, виникає з випадковими інтервалами і залежить від багатьох і навіть невідомих причин. Випадковий характер потоку запитів та час, протягом якого вони обслуговуються, призводять до нерівномірного навантаження QS: невраховані запити можуть бути накопичені на вході (перевантаження QS) або немає запитів, або є менше вільних каналів (недогрузка QS). Потік запитів входить в QS; Деякі з них приймаються для обслуговування в каналах, деякі чекають у черзі служби, а деякі залишають систему без нагляду.

## 1.2 Типи систем масового обслуговування

Системи масового зберігання поділяються на два основних типи (класи): системи очікування (з чергами) та сервісні системи з відмовами. Система обслуговування з відмовами – це система, в якій передбачено відмову в обслуговуванні програми, яка поступає на вхід, що виникає через переповнення сервісних каналів. У наступних процесах ця програма може не бути корисною. Наприклад, можна посилатися на випадок, коли вхідний вилік під час телефонного дзвінка заблоковано через те, що всі канали обслуговування є зайняті, а потім виклик виходить з системи. У сервісних системах з очікуванням, якщо програма з'являється в той момент, коли всі сервісні пристрої зайняті, то вони не залишають систему а формують чергу.

Системи з очікуваннями – це різні типи, які відрізняються за порядком організації черг. І вони приходять з обмеженою або необмеженою довжиною черги або обмеженим часом очікування тощо.

У клієнтській обстановці важливими є ефективні процедури управління чергою. Важливість систем масового обслуговування саме цим фактором і описується. З точки зору функціонування, погані системи обслуговування можуть призвести до падіння продуктивності персоналу, оскільки вони витрачають час на організацію черги. З грошової точки зору, клієнти можуть вирішити залишити чергу, а це призводить до втрати прибутку. У деяких випадках клієнт не розглядає питання про повернення до системи, у якому є некеровані черги. Впровадження хороших систем управління чергою допомагає автоматизувати процес чергування, покращуючи сервіс та закріплюючи лояльність клієнтів.

Багато клієнтських середовищ, незалежно від того, в магазині, поштовому або залізничному вокзалі, вважають, що коли хтось перебуває в черзі, ефективність зростає. Строго кажучи, це не так. Обслуговування по суті відбувається там, де людина виходить з черги, не завершуючи покупку або не вирішуючи мету візиту. Важко оцінити справжню вартість послуг при

такому процесі з ресурсної точки зору, але встановлення хороших систем чергування може допомогти зменшити можливість виходу заявок з черги, що прямо впливає на збільшення ефективності СМО. І це вже, до речі, є доведеним фактом.

Черги трапляються, коли ресурси обмежені. Фактично, черги мають економічний сенс; жодна черга не буде прирівнюватися до дорогої надлишкової потужності. Теорія чергування допомагає в розробці збалансованих систем, які обслуговують клієнтів швидко і ефективно, але не коштують занадто багато. Усі системи масового обслуговування це не що інше як керування вимогами в та поза чергами, і черги тут мають найбільший вплив на результати роботи моделі системи.

Застосовуючи теорію обслуговування, бізнес може розвивати більш ефективні системи обслуговування, процеси, механізми ціноутворення, кадрові рішення та стратегії управління прибуттям, щоб зменшити час очікування клієнтів та збільшити кількість клієнтів, які можуть обслуговуватися.

Теорія чергування як методика управління операціями зазвичай використовується для визначення та оптимізації кадрових потреб, планування та інвентаризації, що допомагає поліпшити загальне обслуговування клієнтів.

Найчастіше використовуються такі службові дисципліни:

1. LIFO – останній ввійшов, перший пішов;
2. FIFO – перший прийшов, перший вийшов.

Якщо черга є пріоритетною для певної кількості  $n$  значень, вимоги будуть надані в залежності від рівня пріоритету та важливості їх вартості.

У подібних випадках розрізняють такі поняття як абсолютний і відносний пріоритет. Відносний пріоритет – це те, коли вимога з вищим пріоритетом чекає завершення роботи програми, яка передує йому, і лише потім надходить у службовий блок.

Пріоритет вважається найвищим і абсолютним, якщо вимога з найвищим пріоритетом при вході в СМО негайно перериває послугу програми з меншим пріоритетом і займає канал обслуговування. Замінена вимога в цьому випадку може знову перейти в чергу, і, можливо, залишити систему.

Сервісний механізм визначається тривалістю процедур обслуговування ( $t$ ) та кількістю вимог ( $\mu$ ), що обслуговуються за одиницю часу. Сервісний механізм може складатися з декількох пристроїв (сервісних каналів). У цьому випадку ці пристрої можуть бути організовані паралельно (наприклад, у супермаркеті є кілька кас) або послідовно (наприклад, послідовна обробка деталей у магазині на токарному та фрезерному автоматах).

Тип служби часу може бути як детермінованим, так і випадковим. Наприклад, обслуговування клієнтів на об'єкті обслуговування вважається завершеним, коли клієнт (або група клієнтів) покидає систему. Тривалість робочого часу ( $t$ ) залежить від запитів клієнта (або груп клієнтів) і є випадковою змінною. Обробка того ж типу деталей, наприклад, на токарному верстаті майстерні, характеризується детерміністичним терміном служби.

Відтік вимог характеризується інтенсивністю.  $\mu$  – інтенсивність обслуговування, тобто кількість вимог, що подаються на одиницю часу, протягом якого пристрій займає послуга.

При детальному вивченні розроблювальної СМО вирішуються наступні завдання:

1. проблеми аналізу СМО – визначення характеристик якості сервісу в залежності від параметрів і властивостей вимог до потоку витрати, параметрів і структури системи обслуговування та дисципліни обслуговування;

2. завдання параметричного синтезу – визначення параметрів системи обслуговування в даній структурі залежно від параметрів і властивостей потоку вимог, дисципліни та якості обслуговування;



3. завдання синтезу структури системи з оптимізацією її параметрів таким чином, що при заданих потоках, дисципліні та якості обслуговування, вартість розробки СМО була мінімальною або була мінімальною втратою замовлень за заданими потоками, дисципліною та вартістю системи.

Ефективність роботи черг обслуговування визначається її пропускною спроможністю – відносною кількістю обслуговуваних запитів. За кількістю каналів  $n$  всі QS поділяються на одноканальні ( $n = 1$ ) і багатоканальні ( $n > 1$ ). Багатоканальна QS може бути однорідною (по каналах) та неоднорідною (за тривалістю обслуговування додатків). Відповідно до дисципліни обслуговування, існує три класи QS.

СМО з очікуванням (необмежене очікування або черга). Коли всі канали зайняті, програма входить до черги і в кінцевому рахунку буде виконана.

СМО змішаного типу (обмежене очікування). Існує обмеження на тривалість черги. Інший тип обмеженого очікування – це обмеження на час застосування в MSS.

Якщо під час вступу заявки на вході в QS принаймні один канал вільний, його служба буде негайно запущена. Однак цілком імовірно, що всі канали зайняті технічним обслуговуванням. У цьому випадку послуга служби затримується, а додаток заходить в чергу. Отже, другим важливим компонентом є черга, параметром якої є кількість місць у черзі  $n$ . У пріоритетних системах чергу можна розділити на кілька черг, для кожної з яких потрібно вказати кількість місць. Кількість місць у черзі може бути обмежена, це можна зробити для кожної черги окремо, і для всіх черг в цілому. Можуть бути конфлікти. У такому разі програмою може бути неприйнятна система прийому заявки.

Залежно від кількості місць у черзі, вони розрізняють QS з відмовами, і, відповідно, QM без збоїв. У QS з невдачею кількість місць у черзі є кінцевою, і через імовірнісний характер як вхідного потоку, так і сервісних процесів існує ненульова ймовірність того, що заявка, отримана на вході СМО, знайде

всі канали, зайняті служба і всі місця в черзі зайняті, тобто вона отримає відмову. У QS, без відмови, заявка негайно призначається на сервіс, якщо на момент її отримання є щонайменше один сервісний канал або він беззастережно прийнятий у черзі служби.

### 1.3 Постановка задачі при моделюванні СМО

Загальне формулювання проблеми полягає в наступному. Система має  $n$  сервісних каналів, кожен з яких може одночасно обслуговувати лише одну вимогу.

Система отримує найпростіший (Пуассона) потік вимог з параметром  $\lambda$ . Якщо в момент отримання чергової вимоги в системі при обслуговуванні вже є щонайменше  $n$  вимог (тобто всі канали зайняті), то ця вимога перетворюється в чергу і чекає початку служби.

Час служби для кожної вимоги  $t$  – це випадкова величина, яка підлягає закону експоненційного розподілу з параметром  $\mu$ .

СМО з очікуваннями можна розділити на дві великі групи: закриті та відкриті. Закриті системи включають системи, в яких потік вимог виникає в самій системі і є обмеженим. Наприклад, майстер, завданням якого є встановлення терміналів в магазині, повинен періодично обслуговувати їх. Кожна налаштована машина стає потенційним джерелом вимог. У таких системах загальна кількість циркулюючих вимог зазвичай і найчастіше постійно.

Якщо джерело має нескінченну кількість вимог, то система називається відкритою. Прикладами таких систем можуть бути магазини, каси, станції, порти тощо. Для цих систем потік вимог можна вважати необмеженим.

Ці особливості функціонування систем цих двох типів накладають певні умови на використовуваний математичний апарат. Розрахунок характеристик роботи СМО різних типів може бути зроблений на основі розрахунку ймовірностей станів СМО (так званих формул Ерланга).

В даному проекті поставлене завдання створення СМО, метою якого є імітація роботи магазину комп'ютерної техніки з певними середньостатистичними умовами прийому додатків, розміром черг та часом обслуговування для кожної програми.

Процес моделювання складається із етапів, поданих на рисунку 1.1.



Рисунок 1.1 – Процес дипломного проектування

Дуже часто необхідно розробляти або вдосконалювати системи, метою яких є багаторазове відтворення завдань одного типу в їх власних умовах для їх вирішення. І в більшості випадків часто трапляється, що рішення даного завдання в системі реального часу виникає з різних непередбачених або очевидних причин, недоцільних і часто навіть неможливих. У таких випадках створюються масові сервісні системи з використанням інструментів моделювання для вирішення конкретної задачі. Така процедура може бути виконана для виконання ряду обов'язкових вимог: збір даних, визначення слабких місць системи та, звичайно, визначення ефективності роботи системи.

Запропонована СМО розроблена в середовищі Matlab Simulink, а саме в додатку SimEvents. SimEvents забезпечує дискретне ядро для моделювання подій, яке управляє та обробляє послідовностями асинхронних подій. Ці події можуть допомогти імітувати зміни режимів роботи та викликати переходи між системами в Simulink, які пов'язані з часом.

За допомогою програми SimEvents можна створювати об'єкти, які можуть бути такими дискретними компонентами, як пакети в системі зв'язку або літаки на злітно-посадковій смузі аеропорту.

Створення, переміщення та обробка об'єктів у системі викликає такі події, як отримання пакета або від'їзд літака. У свою чергу ці події змінюють стан системи і впливають на поведінку системи. Присутня також можливість охарактеризувати свій об'єкт, використовуючи такі атрибути, як адреса призначення, час обробки або латентність сервера.

SimEvents розрізняє лінії для суб'єктів, що представляють собою переміщення об'єктів по певних структурах чи блоках, і рядки подій, які представляють розрахунки на основі подій. Кожен тип лінії використовує унікальний портовий стиль.

SimEvents пропонує бібліотеку блоків, яка дозволяє створювати, обробляти, зберігати та переміщати об'єкти в системі. Є також можливість моделювати як прості, так і складні мережі черг і серверів.

Підключаючи шлюзи та перемикаючі блоки, можна встановити шляхи, якими об'єкти рухаються у відповідь на події. Ці шляхи можуть включати затримки та вибірккові критерії перемикавання. Багато параметрів у SimEvents можна визначити статистично, щоб моделювати імовірнісні варіації в системі.

Присутня і можливість створювати власні конкретні бібліотечні блоки, використовуючи ключові функції Simulink, такі як маскування підсистем та створення бібліотек. А також додавання спеціальних портів сигналів SimEvents до підсистем Simulink, які генерують або отримують події.

Розробка проекту вимагає виконання наступних завдань:

1. дослідити системи масового обслуговування, що імітують роботу магазинів, а також висвітлити їх основні характеристики, якість та параметри продуктивності;
2. проаналізувати подібні системи масового обслуговування та засоби їх моделювання;
3. вибрати найбільш ефективний та найменш дорогий інструмент моделювання для реалізації проекту;
4. модифікувати систему масового обслуговування, що відповідає всім вимогам, та використовувати основні параметри ефективності, взяті з дослідження реальних систем.

Розроблена симуляційна модель може бути використана для розробки програмного продукту або апаратного забезпечення для впровадження на сучасних програмованих логічних інтегральних схемах для подальшого застосування та впровадження в магазинах комп'ютерних технологій.

## 2 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ

### 2.1 Опис алгоритму та будови розробленої моделі системи масового обслуговування

Модель є умовним зображенням або спрощеним зображенням певного об'єкта, який є процесом. Модель створена для більш детального вивчення об'єкта. Модель є теоретично або матеріально створеним об'єктом, який замінює реальний об'єкт, є відносно схожим на нього або має ті ж параметри або набагато зручніші в плані дослідження.

Розглянемо банк або поштове відділення або митниці в аеропорту: кожне з цих місць включають одну або кілька людей в черзі, в той час як за ними приходять й інші люди, які будуть формувати подальшу чергу. Залежно від кількості людей, що вступають в чергу, вони повинні бути оброблені враховуючи часовий фактор і кількість людей на підході. При цьому сама черга може бути довгою або короткою так само як і час очікування може бути довгим або коротким. Це все відноситься до властивостей системи. Такі задачі якраз і покликане вирішувати комп'ютерне моделювання. Нехай придумана для прикладу модельна система складається з наступного:

1. у нас є фіксована кількість позицій, де допомагають клієнтам;
2. кожен клієнт, який входить до системи, має свій певний час обслуговування – деякі вимагають 1 хвилину, інші 5 або 20 хвилин. Ймовірність клієнта, що потребує 1, 5 або 20 хвилин, фіксується відповідно до 50%, 40% та 10%;
3. клієнти виходять випадковим чином, але середня швидкість цього процесу фіксована, скажімо 10 або 15 людей на годину;
4. вони приєднуються до однієї черги, а той, хто очолює чергу, переходить у вільну позицію, якщо така є.

Властивості системи, які необхідно вказати знати:

1. середній час очікування;
2. середня довжина черги;
3. середній час простою (якщо такий є) позицій.

Перше, що необхідно зробити – вивчити, що відбувається в цій системі, перш ніж можна буде реалізувати для неї модель. В даному прикладі будуть використовуватись об'єктно-орієнтовані методи програмування для реалізації різних елементів – осіб, що займаються клієнтами, чергою тощо.

Якщо особа, яка займається клієнтами, простоює (нехай вона буде обробником), він/вона попросить першого клієнта в черзі продовжувати – якщо такий є. Клієнт потребує певного проміжку часу, щоб отримати оброблений запит, і протягом цього часу обробник зайнятий.

Якщо ніхто не чекає, обробник залишається вільним, доки не з'явиться клієнт. В такому випадку потрібно записати, як довго обробник фактично простоює.

Для клієнтів важливо записати час прибуття, довжину черги на той час і час, доки він/вона не буде обслужений. Але нас цікавлять лише середні значення (або, можливо, і максимальне значення). Для цього треба використовувати об'єкт, що представляє чергу, щоб записати, які клієнти чекають і скільки вони чекають. Оскільки клієнти не мають багато властивостей поведінки – просто час прибуття та тривалість їх запиту, то в такому випадку не потрібно моделювати їх через окремі об'єкти.

Тепер розглянемо конкретні приклади інших схожих за принципом роботи та моделювання систем масового обслуговування.

Перший приклад використання алгоритмів імітаційного моделювання масових сервісних систем описано в [4]. Автор пропонує алгоритм моделювання СМО сервера, який приймає програми одного типу блоку, використовуючи наступну послідовність: генератор одиниць на основі часу – FIFO Queue – Single Server – Entity Sink – випадковий номер на основі подій – Час служби, Wait Time, InterTime, Utilization – відображення на дисплеї. Недоліком цієї системи є лише послідовна обробка додатків, спровокована

наявністю лише однієї службової одиниці. Без додаткових блоків та механізмів обробки додатків така система буде неефективною у ситуації, коли через короткий проміжок часу буде потрібно обробляти велику кількість програм.

У [7] автор демонструє більш складну систему, ніж описано раніше. Незважаючи на вищезгаданий алгоритм для запитів на обслуговування, ця СМО також обробляє програми відповідно до пріоритету. Недоліком цієї системи є порівняно високий час обслуговування прикладних програм із високим пріоритетом, що призведе до втрати заявок у випадку, коли необхідно в той самий час обслуговувати велику кількість вимог.

Пропонована система масового обслуговування була створена з урахуванням надійних параметрів та даних існуючої системи, отриманих внаслідок довгострокових та прямих досліджень, на основі інтеграції в механізм роботи установи та спостереження за всіма його ключовими елементами. Варто також зазначити, що кількість каналів обслуговування та потоків вимог також відповідає фактичним даним, отриманим від моніторингу роботи одного магазину. Модель імітує дванадцятигодинний середній робочий день установи.

Магазин включає в себе чотири рівні обслуговування: постійні клієнти, клієнти, орієнтовані на кредити, високопріоритетні клієнти та новий товар.

Також він містить вісімнадцять каналів обслуговування: обслуговування клієнтів вісьмома продавцями, чотирма адміністраторами сервісів (ІТ-майстрами), двома кредитними брокерами та двома касирами. За обслуговування товару відповідальні два вантажники.

Кожен клієнт, щоб отримати статус обслуженого, повинен пройти три або чотири канали обслуговування: кредитний брокер (необов'язково), продавець, каса, та адміністратор сервісів.

У свою чергу, кожна одиниця товару має пройти два канали: складування при прибутті та безпосередньо викладка товару на зал, якою займаються вантажники позмінно.



Тепер більш детально про канали обслуговування.

Якщо форма оплати клієнта кредит, тоді він відправляється до кредитного брокера. У брокерів аналогічний із продавцями та касирами робочий графік. Час обслуговування клієнта брокерами складає від двадцяти до п'ятдесяти хвилин.

Робота продавців заключається у необхідності допомоги клієнтам, які щойно потрапили до системи, тобто магазину. Вони надають необхідні консультації, інформують стосовно особливостей даного конкретного закладу, продають товар та займаються його післяпродажною підготовкою. Кінцевим результатом роботи продавця є клієнт, який вже визначився із вибором товару, необхідності додаткових послуг та способом оплати і готовий до того, щоб розплатитись, отримати виконані додаткові послуги та покинути систему. Продавців у магазині працює вісім: чотири в першу шестигодинну зміну, чотири в другу. У неробочий час вони знаходяться поза магазином і жодної участі в його функціонуванні не приймають. Середній час обслуговування відвідувачів визначений регламентом закладу і коливається від однієї до п'ятнадцяти хвилин. Окремі випадки не враховуються.

Обслужені продавцями покупці відправляються до касирів, або до кредитних брокерів. Якщо клієнт розраховується готівкою, то він автоматично пропускає один канал обслуговування (брокерів) та відразу відправляється до продавців і потім до касирів. Касирів двоє, і в них також шестигодинні робочі зміни. Час обслуговування касирами клієнтів складає від двох до двадцяти хвилин. У випадку кредиту клієнт направляється до каси тільки після того, як його обслужив брокер. Після каси він відправляється до адміністратора сервісів, які працюють також шестигодинними змінами. Час обслуговування клієнтів адміністратором складає від 2 до 10 хвилин. Тільки після цього задоволений покупець покидає магазин, і це символізує вихід обробленої заявки із системи.

Обслуговування клієнта можливе тільки у вказаній вище хронології, тобто клієнт не може бути обслужений касою, доки його не обслужить продавець.

Згідно зазначених часових інтервалів, мінімальний час повного обслуговування одного клієнта – 5 хв, максимальний – 130 хвилин. При мінімальному часу обслуговування магазин за дванадцять годин здатен обслужити 300 покупців. При максимальному – 54. В середньому магазин обслуговує приблизно 120 клієнтів за робочий день, тобто одного клієнта кожні 20 хвилин.

Другий потік заявок у розробленій системі – це новий товар, який привозять до магазину. Необхідність наявності цього елемента обумовлена тим, що розкладкою товару займаються безпосередньо вантажники – два ключових канали обслуговування – і кожна хвилина їхнього робочого часу повинна бути симульована системою. Крім того, кількість приведеного в допродажну готовність товару є також показником успішності та ефективності магазину.

Заявки другого потоку повинні бути обслуженими двома каналами. Перший канал симулює приймання нового товару на магазин. Вказується з якою інтенсивністю товар прибуває в магазин за один робочий день. Згідно досліджень, в один робочий день в магазин може прийти від 40 до 100 одиниць товару (дрібний товар в цьому випадку вимірюється ящиками, в яких прибуває партіями). В середньому це 70 одиниць.

Наступний канал обслуговування якраз імітує розкладку товару вантажниками безпосередньо на полиці чи склад магазину. В середньому вантажник затрачає на розпакування однієї одиниці товару від однієї до десяти хвилин. Тобто за дванадцять годин обидва вантажники спільними зусиллями можуть розпакувати та розкласти від 72 до 720 одиниць товару. Зазвичай вони розпаковують по п'ятдесят в день, тобто затрачають на розкладку однієї одиниці товару близько шести хвилин.

Алгоритм роботи розробленої системи масового обслуговування:

1. початок симуляції;
2. на вхід поступають чотири вищеописаних потоки заявок;
3. перші 360 хвилин працює тільки перша зміна:
  - a. продавці 1, 2, 3, 4 обслуговують всіх клієнтів від 0 до 360 хвилини;
  - b. брокер 1 один обслуговує клієнтів орієнтованих на кредит від 0 до 360 хвилини;
  - c. касир 1 обслуговує клієнтів, які прийшли від брокерів та продавців від 0 до 360 хвилини;
  - d. ІТ-майстри 1 та 2 обслуговують всіх клієнтів обслужених до цього продавцями, брокером та касиром від 0 до 360 хвилини.
4. вантажник 1 приймає товар від 0 до 360 хвилини;
5. від 360 до 720 хвилини працює друга зміна:
  - a. продавці 5, 6, 7, 8 від 360 до 720 хвилини обслуговують клієнтів;
  - b. брокер 2 від 0 до 360 обслуговує клієнтів, орієнтованих на кредит;
  - c. касир 2 та ІТ-майстри 3 та 4 від 360 до 720 хвилини обслуговують всіх клієнтів.
6. вантажник 2 приймає товар від 360 до 720 хвилини.

## 2.2 Практичні особливості імітаційного моделювання

Імітаційне моделювання (скорочено ІМ) у багатьох випадках є підвидом аналогів симуляції, який реалізується за допомогою певного набору математичних інструментів та засобів (часто програмного забезпечення) з метою проведення чіткого та цілеспрямованого вивчення реального об'єкта чи структури, використовуючи апаратне забезпечення можливостей

комп'ютера, використовуючи режим імітації певного програмного забезпечення або виконуючи оптимізацію цих параметрів [13].

Імітаційна модель – це спеціалізований програмний комплекс, що дозволяє відтворити процес активності будь-якого реального об'єкта. Цей комплекс, що використовує комп'ютерну пам'ять, може виконувати паралельні процеси, що мають власні просторово-часові параметри, і є аналогом реального процесу. Моделі використовуються для отримання знань безпосередньо з практичних досліджень, таких як пошук індикаторів продуктивності об'єкта, перевірка надійності системи в різних режимах його роботи або порівняння та існуючих функцій, а також алгоритми роботи об'єктів. Щоб перевірити ймовірність процесу моделювання, нерідко використовується група покращених моделей, які постійно працюють. У цьому випадку, якщо подальша деталізація властивостей моделювання не може вплинути на кінцеві результати, то немає сенсу ускладнювати модель розвитку в майбутньому. Як правило, вони моделюють лише ті властивості реального об'єкта, які безпосередньо впливають на досліджувані показники ефективності за заданим обмеженням. Результати такої імітаційної моделі мають чітке значення і вказують на проблеми та складність системи.

Імітаційні моделі мають свої недоліки:

1. значні витрати машинного часу;
2. дуже відчутні труднощі при оптимізації моделі;
3. мала точність характеристик ймовірності деяких складних подій;
4. неправильні висновки та рекомендації в деяких випадках;
5. інші різноманітні непередбачувані помилки.

Імітаційна модель виправдовує свій розвиток, якщо:

1. остаточні результати відповідають очікуванням або перевершують їх;
2. план перевіряється на припущеннях, які розробник прямо визначає так, що аналітичний метод також може бути використаний;

3. надмірна складність ситуації достатня, щоб її можна було вирішити іншими методами моделювання, зокрема аналітичними;

4. необхідно забезпечити можливість накопичення знань та даних про досліджуваний об'єкт, і ці дані можуть бути підтверджені реальним експериментом.

Щоб створити будь-яку імітаційну модель у своїй складності, потрібно мати спеціалізоване середовище моделювання з набором інструментів, методів, технологій та процедури. Крім того, необхідно чітко зрозуміти логіку моделі розвитку, а також закономірності поведінки відтворюваного процесу в просторово-часовому сенсі, і якщо модель все ще передбачає наявність економічного аспекту, то також фінансова динаміка має бути доступною.

Діяльність комп'ютера при процесі ІМ поділяється на два типи:

1. робота, яка включає в себе безпосереднє моделювання об'єкта та подальшу модифікацію моделі;

2. запуск моделювання розробленої моделі та збір даних.

Імітаційна модель актуальна у двох випадках:

1. якщо є потреба в безпосередньому контролі складного процесу. Тоді вони використовуються як інструмент в межах контуру, здатного адаптувати систему управління;

2. якщо необхідно проводити дослідження з дискретними та безперервними моделями об'єктів, які виділяються своєю складністю, для того, щоб контролювати їх динаміку в особливих умовах та у ризикованих ситуаціях, то будь-яке інше моделювання є недоцільним або навіть неможливим.

Слід зазначити, що комп'ютерні інструменти моделювання вирішують досить багато завдань. Серед них варто відзначити:

1. визначення витрат та часових характеристик при моделюванні логістичних процесів;

2. висновок та систематизація прогнозу фінансових структур підприємств, з урахуванням певного проміжку часу і з урахуванням змін у рахунках;

3. реінжиніринг підприємства, близького до розпаду, у його подальших діях;

4. визначення результатів роботи кредитних груп та профспілок;

5. управління процесом формування різних типів проектів на різних етапах, з урахуванням можливих непередбачених обставин та ризиків;

6. аналіз функціонування таких комплексів як підприємств автотранспорту з урахуванням особливостей грошових та товарних регіональних даних;

7. відтворення детальної діяльності рятувальної бригади в разі аварії або катастрофи, з передбаченням різних ризиків, ситуацій та несприятливих обставин;

8. аналіз ієрархії структур різних систем управління, їх зв'язок з підлеглими системами, з урахуванням пропускнуої здатності каналів зв'язку;

9. моделювання роботи різних систем з метою перевірки їх надійності та вартості обробки інформації. Безпосередньо для банківських систем.

Імітаційна модель використовуються, коли:

1. дорогий або неможливий експеримент на реальному об'єкті;

2. неможливо побудувати аналітичну модель, оскільки в системі є час, причинно-наслідковий зв'язок чи наслідки або нелінійні, стохастичні (випадкові) змінні;

3. необхідно моделювати поведінку системи в часі.

Імітаційне моделювання може застосовуватися в різних сферах діяльності. Особливо ефективно моделювання при вирішенні наступних завдань:

1. проектування та аналіз виробничих систем;

2. оцінка різних систем зброї;

3. визначення вимог до обладнання та протоколів мереж зв'язку;

4. модернізація різних процесів у бізнесі;
5. аналіз фінансово-економічних систем.

Під системою імітаційного моделювання розуміється програмний або апаратно-програмний комплекс, призначений для вирішення завдань з використанням методу імітації. При виборі різновидів імітаційних систем вони базуються на тому, що вони є інструментальними засобами, що забезпечують автоматичну підтримку деяких видів діяльності користувачів.

Імітаційна модель реалізує алгоритм вирішення проблем та надає користувачеві сервісні можливості для керування обчислювальним процесом. Автоматизована підтримка інших етапів системного аналізу за допомогою системи імітації не є обов'язковою. Однак ступінь їх автоматизації визначає можливості імітаційної моделі і є основою їх класифікації. Враховуючи етап системного аналізу та вирішені за допомогою нього технологічні завдання, вибирається можливий набір засобів імітаційної моделі, що автоматизує виконання ряду функцій, реалізованих на цих етапах.

Створення моделі може підтримуватися наступними інструментами автоматизації:

1. частково готові моделі;
2. компілятори алгоритмічної мови високого рівня, спеціалізація якої полегшує процес алгоритмів моделювання креслення;
3. спеціальна мова високого рівня, що дозволяє виконувати інформаційно-математичний опис системи;
4. перетворювачі моделей, що дозволяють перетворити моделі одного виду на модель іншого типу;
5. засоби моніторингу узгодженості різних типів моделей з концептуальним представленням моделі.

Перевірка адекватності та технічної можливості може бути виконана за допомогою:

1. програми розрахунку показників адекватності;

2. автоматизована технологія проведення обмеженого експерименту з імітаційною моделлю;

3. програми обчислення характеристик складності моделі;

4. програми розрахунку ресурсних показників методу вирішення завдань.

Подальша модифікація моделі може забезпечити:

1. автоматизовані технології редагування тексту;

2. програми еквівалентних перетворень математичних та алгоритмічних моделей даного класу.

Створення алгоритму вирішення проблеми може бути підтримане:

1. методичними бібліотеками та програмними пакетами;

2. розробленням алгоритмів вирішення завдань;

3. системами інформаційної підтримки прийняття рішень тощо.

Розробка та вдосконалення схеми вирішення проблеми може бути виконана за допомогою:

1. програмних засобів для контролю сумісності сімейства завдань;

2. текстовими редакторами обчислювальних схем і т. д.

При виборі способу симуляції слід враховувати всі функції, які вони надають та які можна згрупувати у наступні групи:

1. основні характеристики;

2. сумісне програмне забезпечення;

3. анімація;

4. статистична працездатність;

5. звіти з вихідними даними та діаграмами;

6. послуги, що надаються клієнтам та документація.

Найпопулярнішими пакетами моделей імітації є:

1. Rockwell Automation Arena;

2. AnyLogic XJ Technologies;

3. GPSS Всесвітня компанія Minuteman Software;

4. Статус Scittor Process 1.0.2;



5. Powersim 2.01 компанії Modell Data AS;
6. Ithink 3.0.61 виробництва High Performance Systems;
7. Simulink компанії Matlab.

Ці пакети найбільше відрізняються стилем моделювання, тобто середовищем, за допомогою якого створюються моделі. У пакеті Process Charter модель будується за допомогою блок-схеми. Powersim і Ithink використовують систему позначень Systems Dynamics, запропоновану в 1961 р. Джейм Форрестером Массачусетського технологічного інституту. Extend застосовує компоновочні блоки. Всі продукти, крім Process Charter, дозволяють проводити аналіз чутливості, тобто багаторазово виконувати модель із різними вхідними параметрами, щоб зрівняти результати декількох прогонів.

Імітаційне моделювання розроблюваної системи масового обслуговування відбувалось таким чином: в системі є два потоки заявок: товар та клієнти. Клієнтів обслуговують позмінно всі канали обслуговування, товар тільки вантажники (також позмінно).

Спочатку заявка клієнта проходять перший канал обслуговування – продавця. Час обслуговування продавцем та розмір черги до нього приведені вище. Потім обслужена продавцем заявка відправляється або до брокера, або на касу. На це рішення впливає спосіб оплати клієнта. Якщо клієнт оплачує готівкою, тоді він автоматично пропускає один канал обслуговування (брокера) і відразу відправляється на касу. Якщо форма оплати кредит, тоді клієнт мусить спочатку бути обслуженим брокером, і тільки після цього він відправляється до касира. Потім він відправляється до ІТ-майстра, і тільки після цього покидає магазин. У результаті симуляції заявка клієнта має пройти мінімум три канали обслуговування та максимум чотири.

Заявка товару проходить тільки два канали обслуговування (прийом та вітринну викладку), за які відповідають вантажники, і тоді покидає систему.

Всі зв'язки блоків, а також основні та додаткові параметри за допомогою яких виконується імітаційне моделювання описаних процесів наведені у розділі 3.

### 2.3 Simulink як середовище імітаційного моделювання

Simulink – це інтерактивне програмне забезпечення, метою якого є імітація та аналіз систем, які мають динамічний характер. Методи, що пропонує Simulink, можуть бути використані шляхом побудови графічних блок-схем, вивчення продуктивності систем розробки, а також їх моделювання, а також пошук способів вдосконалення моделі реального об'єкта. Simulink повністю інтегрований з MATLAB, і таким чином він здатний надавати негайний доступ до всього спектру інструментів проектування, моделювання та аналізу. Програма Stateflow дозволяє імітувати події та їх поведінку. Переваги цього програмного забезпечення роблять його найбільш популярним інструментом для моделювання складних цифрових процесорів, систем зв'язку та управління.

Simulink – це пакет додатків для Matlab. У цій програмі використовується такий метод моделювання, як візуальне програмування, що передбачає створення моделі з використанням стандартних функціональних блоків тих чи інших бібліотек, які користувач вибирає або створює сам, а також робить обчислення. Відмінність від класичних методів моделювання тут полягає в тому, що немає необхідності відмінно знати мови програмування високого рівня або складні математичні методи та закони. Достатніми вважаються посередні впевнені знання в цих сферах та комп'ютерні навички. Ну, зрозуміло, також знання предметної області навчання та впевненого володіння самим програмним забезпеченням.

Simulink повністю не залежить від пакета Matlab, тому немає потреби знати сам Matlab. Проте доступ до функцій вищезазначеного середовища повністю відкритий для користувача. Деякі з них також є частиною пакетів та інструментами, які спільні з Simulink, такими як LTI-Viewer або Control System Toolbox. Однак існують додаткові блокові пакети, мета яких – різні області застосування. До них відносяться Blockset обробки цифрових сигналів або Power System Blockset.

Робота в середовищі Simulink – це можливість самостійно оновлювати бібліотеки блоків, створюючи унікальні.

У процесі моделювання користувачеві дається можливість вибирати методи для вирішення диференціальних рівнянь та, крім того, методи зміни робочого часу моделі (з змінним або фіксованим кроком). У процесі моделювання можна відслідковувати процеси, що виробляються системою. Для цього існують певні пристрої спостереження, що є частиною самої бібліотеки. Результати моделювання можна інтегрувати в графічний або табличний формат.

Процес розробки моделі передбачає монтаж системи за допомогою блок-схеми. Кожен елемент схеми – це функціональний блок, який має графічне подання, математичний метод дії та синтезу, а також формально-чисельні параметри та характеристики. Всі блоки з'єднані лініями, які є графічним поданням руху потоків даних у об'єкті.

Найбільш поширеною є область, в якій ефективний інструментарій Simulink – це економіка. У такому випадку ця заявка ефективна для використання показників для кількісного визначення та вивчення таких явищ та економічних процесів, як динамічний кризовий аналіз, оподаткування бізнесу, ринкова рівновага або оптимальне планування процесу в банках, різноманітних компаніях та фондах.

Враховуючи всі переваги, середовище Simulink використовується для моделювання розробленої СМО.

### 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПОСТАВЛЕНОГО ЗАВДАННЯ

#### 3.1 Опис розроблювальної схеми та її складових

Мета проекту – розробка імітаційної моделі системи масового обслуговування. Основне завдання розроблювальної моделі – моделювання роботи магазину комп'ютерних технологій. Модель включає в себе певні розраховані вхідні параметри, які точно або приблизно відповідають параметрам реального магазину. Завдання – продемонструвати ефективність чи неефективність практичного використання системи на основі вихідних даних. Ефективна система вважається здатною обслуговувати максимальну кількість вимог без урахування факторів, які не мають прямого відношення до неї і не впливають на його функціонування, але тільки на потік додатків. Параметри системи масового обслуговування, такі як розмір черги або час, який обслуговуючий канал витрачає на підтримку однієї програми, близька до реальної.

Таким чином, розроблена у даному дипломному проекті СМО імітує дванадцятигодинний робочий день магазину комп'ютерної техніки з чотирма потоками вимог – звичайними клієнтами, клієнтами, орієнтованими на кредити, високопріоритетними клієнтами і новим товаром, і яка містить вісімнадцять каналів обслуговування, кожен з яких буде описаний більш детально нижче.

За символікою Кендаля-Башаріна розроблену СМО можна позначити таким чином:

$$D / D_1 / LS / W / FF / S$$

Цей спосіб класифікації обслуговуючих систем включає шість елементів, які характеризують шість ключових параметрів системи.

Для класифікації систем використовується наступне позначення

$$(a / b / c) : (d / e / f)$$

де символи a, b, c, d, e, f, пов'язані з істотними елементами модельного представлення процесів масового обслуговування.

A – розподіл моментів надходження заявок на обслуговування,

B – розподіл часу обслуговування (або вибуття обслужених клієнтів),

C – число паралельних вузлів обслуговування,

D – дисципліна черги.

E – максимальне число, яке допускаються в систему вимог (число вимог в черзі + число вимог, прийнятих на обслуговування),

F – ємність джерела, що генерує заявки на обслуговування.

Для a й b прийняті наступні позначення.

M – пуассонівський розподіл моментів надходжень заявок на обслуговування або вибуття з системи обслужених клієнтів (або експоненціальний розподіл інтервалів часу між моментами послідовних надходжень).

D – фіксований (детермінований) інтервал часу між моментами послідовних надходжень в систему заявок на обслуговування або детермінована (фіксована) тривалість обслуговування.

Ek – розподіл Ерланга або гамма – розподіл інтервалів часу між моментами послідовних надходжень вимог в СМО.

G – розподіл довільного виду моментів надходження в систему заявок на обслуговування (або інтервалів часу між послідовними надходженнями вимог).

G – розподіл довільного виду моментів вибуття з системи обслужених клієнтів.

Приклад позначення (M / D / 10):( GD / N / F) – система масового обслуговування з пуассоновским вхідним потоком, фіксованим часом обслуговування і 10 паралельними вузлами, дисципліна черги не регламентована, незалежно від числа вступників вимог даної системи (черга + обслуговування клієнтів) не може вмістити більше N вимог.

Мета даного аналізу – розробка критеріїв ефективності СМО. Серед режимів роботи СМО явно виділяються перехідні і стаціонарні режими. До перехідних відносяться процеси входу та виходу, які з плином часу переходять в стаціонарні. Якщо  $\lambda$  – інтенсивність надходжень  $> m$  (інтенсивності вихідного потоку), то стаціонарний режим не рентабельний. Аналіз нестаціонарних режимів складний, тому далі будуть розглядатися тільки стаціонарні процеси.

Стаціонарний ординарний потік без післядії називається найпростішим. Математичний опис найпростішого потоку задається системою функцій  $p_0(t), p_1(t), \dots, p_n(t)$ .

За дванадцятигодинний робочий день в магазин приходить близько 100-150 потенційних клієнтів. Враховується тільки ті відвідувачі, які безпосередньо зацікавлені у обслуговуванні і будуть обслужені мінімум одним каналом обслуговування. Оскільки 10 годин – це шістсот хвилин, звідси випливає, що по правилах детермінації, потенційний клієнт прибуває в магазин кожні 4-7 хвилин. Це й буде параметром генерації заявок джерелом. На рисунку 3.1. подано вікно блоку Event-Based random number, де при активній вкладці Uniform вказуємо інтервал поступлення заявок.

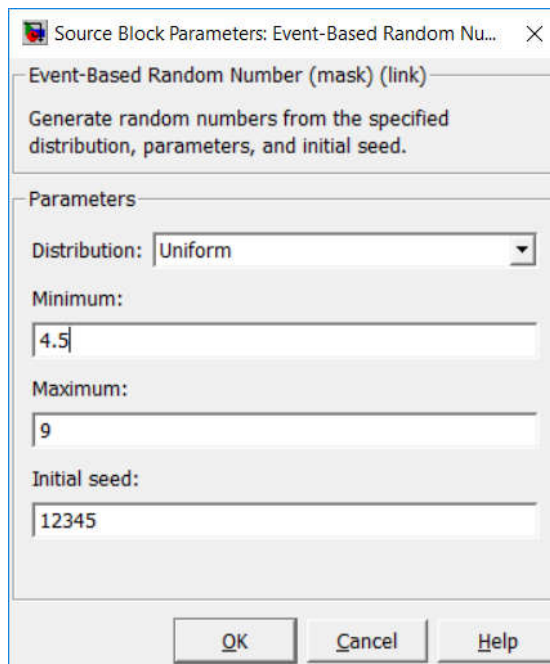


Рисунок 3.1 – Вікно блоку Event-Based random number із вказаними параметрами

Приблизно тридцять відсотків суми постійних клієнтів складаються з клієнтів, орієнтованих на позику. Такі клієнти повинні пройти додатковий канал обслуговування кредитними брокерами.

Клієнти з високим пріоритетом складають не більше 5-10 відсотків від загальної кількості постійних клієнтів.

Детермінований розподіл також містить три параметри: стаціонарність, післядію і ординарність.

Стаціонарна ситуація не передбачає жодних змін характеристик надходження заявок у часі. Тобто подання джерела додатків у цьому випадку залежить лише від заданого інтервалу надходження і не залежить безпосередньо від часу.

Післядія визначає, чи залежать імовірнісні характеристики подання заяв від попередніх подій. У цьому випадку канали обслуговування безпосередньо впливають на потік додатків.

Ординарність показує, чи можна подавати заявки групами.

Детермінізм подачі аплікацій у цій системі стаціонарний, звичайний і без наслідків.

Другий параметр,  $D_1$ , – це індикатор, який вказує закон розподілу запиту служби часу службовим каналом. Параметр  $D_1$  означає, що час, необхідний для обслуговування програми, також генерується детерміновано, тобто з відомим інтервалом обслуговування. Наприклад, обслуговуючий канал продавця. Згідно з реальними дослідженнями, в середньому продавець на обслуговування одного покупця витрачає від 2 до 30 хвилин. Тому в блоці випадкових чисел на основі подій, який генерує випадкове значення часу обслуговування споживача для першого постачальника, відповідні параметри вказані так само, як і параметри подання джерела для службових каналів.

Третій параметр LS означає, що ця система є багатофазною (Link System). Це, в свою чергу, вказує на те, що програми проходять через більш ніж одну стадію обслуговування, що в цьому випадку є не що інше, як обслуговуючі канали. Отже, у цій моделі вимоги, які вказують на потік клієнтів, спочатку повинні обслуговуватися продавцями або брокерами (залежно від типу заявки), потім касиром і безпосередньо в кінці – адміністратором сервісів. Отримуються у результаті від трьох до чотирьох фаз обслуговування, кожна з яких повинна бути пройдена, щоб заявка врахувалась до списку обслужених системою.

Четвертий параметр W вказує на дисципліну запитів обслуговування через обслуговуючі канали. W означає, що в цій системі є черги, тобто вимоги обслуговуються з очікуваннями. У цьому випадку, якщо службовий канал вже зайнятий, то заявки утворюють черги для технічного обслуговування та не залишають систему без нагляду.

П'ятий параметр FF визначає тип черги обслуговування. FF – це ярлик від FIFO, який, у свою чергу, перекладається з англійської мови як First Input First Output. Тобто до обслуговування першою приходиться заявка, яка першою увійшла в чергу.



Четвертий та п'ятий параметри – тип і розмір черги – вказуються в Simulink в одному блоці, який називається чергою, і може мати різні характеристики в залежності від типу черги. Всі черги для цього проекту є типом FIFO, тому в моделі використовуються лише блоки очок FIFO. На малюнку 3.2 наведено приклад налаштування розміру черги у відповідному блоці.

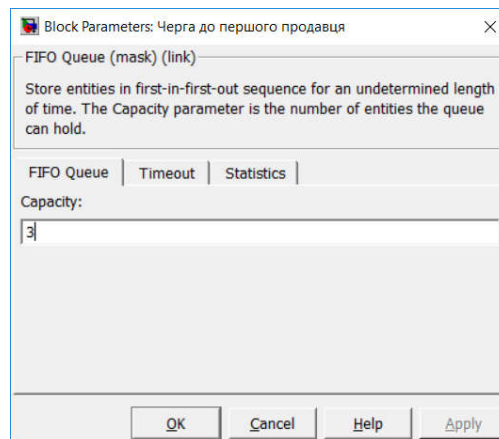


Рисунок 3.2 – Налаштування блоку FIFO Queue

На рисунку 3.2 показано вікно чергування запитів на обслуговування продавцем. За даними досліджень – максимальна кількість покупців в черзі досягає трьох осіб. Якщо черга заповнена, заявка, тобто клієнт, залишає систему.

Шостий параметр  $S$  – це спосіб, у який заявки займають сервісний канал.  $S$  – це найбільш часто використовуваний параметр, який вказує на те, що заявки до сервісного каналу послідовно знижуються.

Наведені параметри найкраще вказують тип моделі та його функціональність. Нижче наводиться детальний опис структури модельної схеми та основних блоків, через які працює система.

### 3.2 Реалізація блоків СМО в Simulink

Часова одиниця симуляції моделі – 1 хвилина. Як наслідок, час симуляції буде дорівнювати 720 одиницям, що означає дванадцятигодинний робочий день.

Модель включає наступні компоненти:

Time-Based Entity Generator – моделює надходження заявок в систему. Закон розподілу часу надходження заявок може бути заданий безпосередньо в блоці (Постійний, Експоненціальний або Рівномірний) або змодельовано з використанням зовнішніх блоків.

Exponential Interarrival Time Distribution – підсистема, яка створює випадкові інтервали часу появи заявок за експоненціальним законом (моделює пуассоновський потік заявок).

FIFO Queue – моделює чергу, яка зберігає (накопичує) надходять заявки для подальшого обслуговування. В даному випадку довжина черги обмежена.

Single Server – моделює обслуговуючий пристрій, час обслуговування якого має експоненціальний розподіл. Закон розподілу часу обслуговування може бути заданий безпосередньо в блоці або з використанням зовнішніх блоків.

Exponential Service Time Distribution – створює випадкові інтервали часу обслуговування заявки за експоненціальним законом.

Waiting Time Evaluation – розраховує теоретичне значення часу очікування заявки в черзі.

Entity Sink – блок, що зберігає обслужені заявки після їхнього виходу із системи.

Scope: Waiting Time Comparison – відображає теоретичне значення моделювання процесу.

Scope: Queue Content – відображає зайнятість черги (довжину) в процесі моделювання.

Display: Queue Waiting Time – відображає середній час перебування заявки в черзі.

Приклад реалізації за допомогою деяких вищезгаданих блоків функціонуючого блоку системи, який відповідає за керування робочим часом працівників зображений на рисунку 3.3.

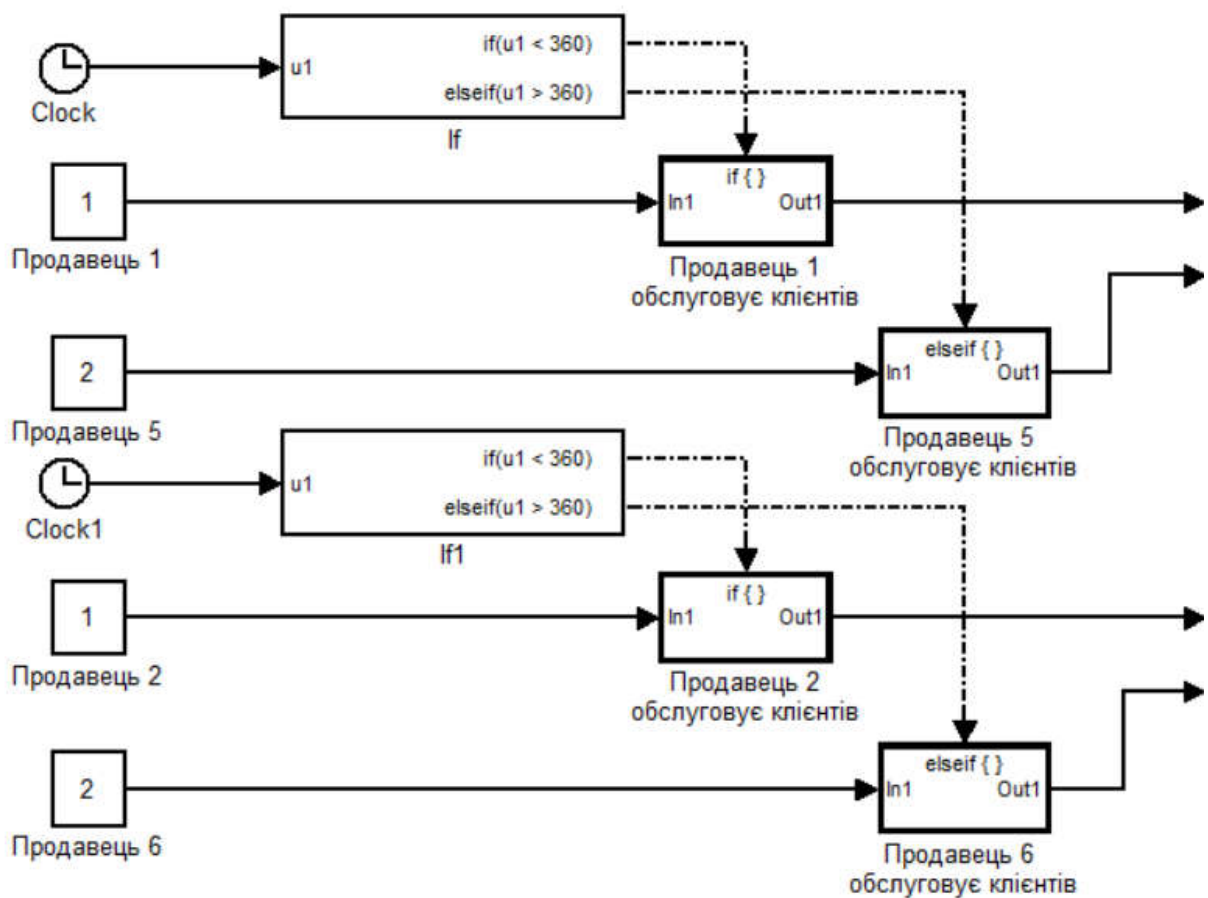


Рисунок 3.3 – Структура блоків регулювання робочого часу працівників

Складається дана структура із генератора заявок Event-Based Entity Generator та генератора часу інтервалу між заявками Time-Based Entity Generator, який тут виконує роль проміжного елемента, що бере значення із порту t, оскільки генератор заявок подає вимоги детерміновано, а отже уже із врахуванням інтервалу часу.

Перший потік заявок, який імітує кругообіг клієнтів в магазині, схематично реалізується способом, представленим на рисунку 3.4.

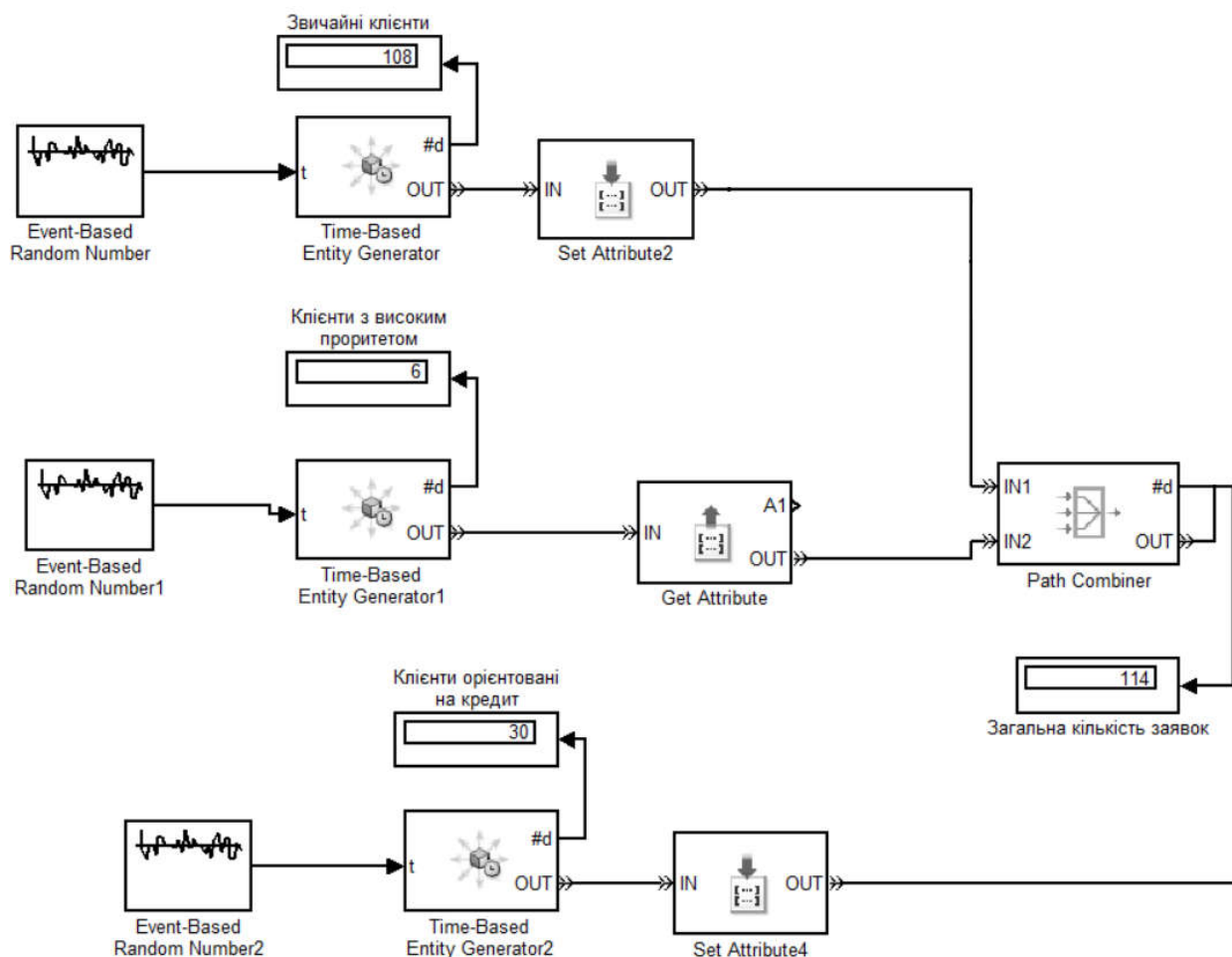


Рисунок 3.4 – Структура блоків, які генерують потоки заявок

Існує також дворівнева ієрархія пріоритетів у системі. Високі пріоритетні заявки організовані як окремий потік вимог і відрізняються від звичайних заявок часом і пріоритетом для обслуговування таких додатків. Заявки з високим пріоритетом обслуговуються по черзі і вдвічі швидше, ніж звичайні вимоги на всіх каналах обслуговування.

Спочатку програми з високим пріоритетом отримують спеціальний атрибут Priority. Цей атрибут здатний прочитати всі сервісні канали і, залежно від наявності, обслуговувати порядок поза чергою чи ні, і скільки

часу. Заявки з високим пріоритетом надходять рідко і складають не більше 10% кількості звичайних клієнтів та клієнтів, орієнтованих на позику.

Вхідний потік найбільш поширений на практиці; це найпростіший потік запитів, що володіють властивостями стаціонарності і без наслідків:

1. властивість стаціонарності, яка виражає незмінність імовірнісного режиму течії з часом. Це означає, що кількість запитів, що надходять в систему рівними інтервалами часу, повинна, в середньому, бути постійною;

2. не існує жодних наслідків для взаємної незалежності одержання тієї чи іншої кількості вимог до послуг у непересічних періодах часу. Це означає, що кількість запитів, що надходять в певний проміжок часу, не залежить від кількості запитів, поданих у попередній період часу;

У найпростішому потоці додатків він дозволяє отримати характеристики QS з параметрів потоку вхідного сигналу, що важко для інших типів потоку програм.

Якщо QS забезпечує бажану продуктивність системи за допомогою найпростішого потоку, то система буде обслуговувати інші потоки запитів з однаковою інтенсивністю.

Потік запитів – це сукупність вимог, які входять в систему і потребують технічного обслуговування. Вивчається вхідний потік вимог з метою встановлення законів цього потоку та подальшого підвищення якості обслуговування.

У більшості випадків вхідний потік некерований і залежить від ряду випадкових факторів. Кількість запитів, що надходять за одиницю часу, є випадковою величиною. Випадкове значення також є інтервалом часу між сусідніми вхідними запитами. Проте, припускається, що середня кількість отриманих запитів за одиницю часу та середній часовий інтервал між сусідніми вхідними вимогами залежні.

Середня кількість клієнтів, що входять до сервісної системи за одиницю часу, називається швидкістю надходження запиту.

На практиці найпростіші умови черги не завжди суворо виконуються. Часто є нестаціонарність процесу (в різні години дня та в різні дні місяця потік вимог може бути різним, він може бути більш інтенсивним вранці або в останні дні місяця). Існує також ефект, коли кількість вимог до випуску товарів наприкінці місяця залежить від їх задоволення на початку місяця.

Наступним етапом імітації є поступання заявок в блок Path Combiner. Дана структура після запрограмованої дії передає заявки далі – у чергу, чия операція реалізується блоком черги FIFO з заданою чергою розміром у 50 заявок. Цей блок у поєднанні з наступним блоком, що представляє перший канал обслуговування, є структурою, яка забезпечує чіткий розподіл максимальної кількості клієнтів, яких може вмістити приміщення магазину, і виключити можливість нескінченного і безперервного потоку заявок. Після цього програма отримує блочний вихідний комутатор, який виводить їх, послідовно перемикаючи їх на наступні канали обслуговування, які блокуються відповідно до значень, взятих у порту p.

Далі починаються наступні етапи обслуговування заявок. Схема, яка реалізує цей процес, зображена на рисунку 3.5.

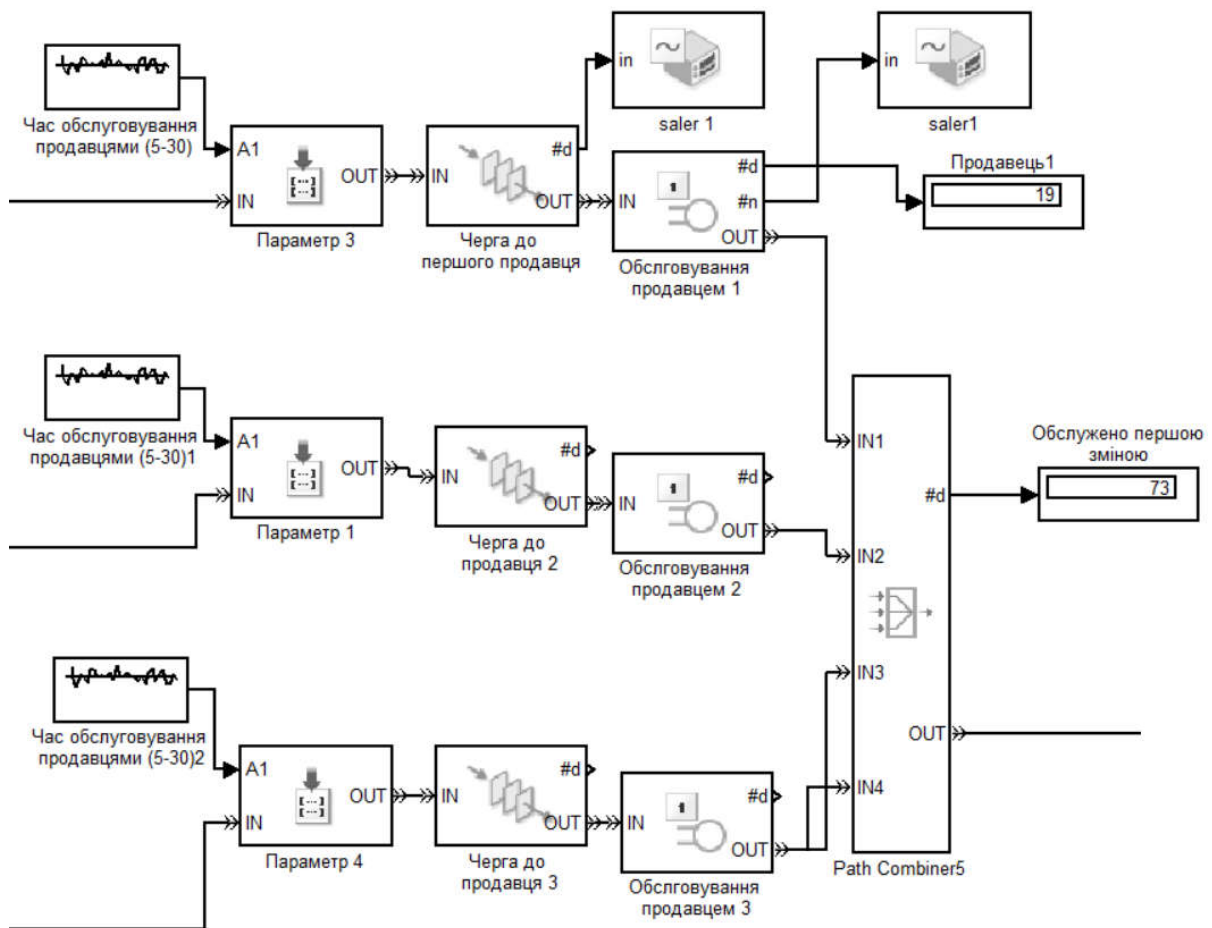


Рисунок 3.5 – Схема, яка реалізує подальше обслуговування потоку заявок продавцями

Самий схематичний процес обробки запитів аналогічний тому, який показано на малюнку 3.5, з тією різницею, що існує набір параметрів, який дозволяє двом вхідним портам приймати різні значення, формувати сигнали та створювати додаткові атрибути для їх роботи у відповідності з завданням. Вивід діаграми процесу обслуговування відповідає блоку Score. Для числового значення, блок Display. Крім того, ще одна особливість полягає в тому, що в цій частині схеми блок випадкових чисел на основі подій генерує лише випадкове значення часу, що обслуговується каналом однієї вимоги, замість самих програм.

Після завершення обслуговування всі канали передають запити про службу в Path Combiner, який генерує один із потоків і посилає його далі на

перемикач, який також отримує значення часових інтервалів роботи продавців. Від цього починається третє обслуговування першого потоку заявок. Його схематичне виконання показано на рисунку 3.6.

Принцип, знову ж таки, однаковий, тільки в цьому випадку існують лише два обслуговуючі канали, тобто два касири.

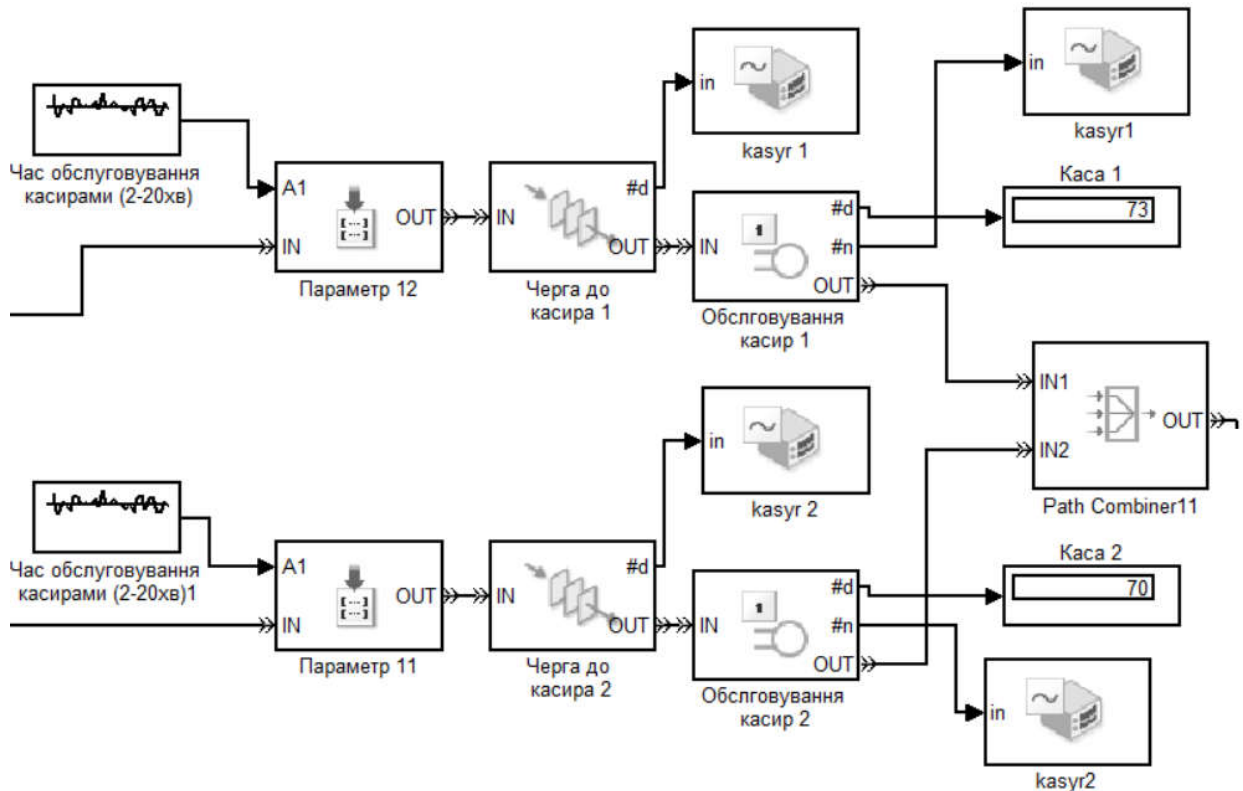


Рисунок 3.6 – Схематична реалізація обслуговування заявок касою

Принцип такий же, як і в попередньому випадку, лише з індикацією за іншими параметрами. Після завершення служби касира, два потоки додатків утворюють один потік знову, використовуючи блок Path combiner, і переходять до наступного етапу обслуговування – технічного обслуговування ІТ-майстрами. Ця структура показана на малюнку 3.7.



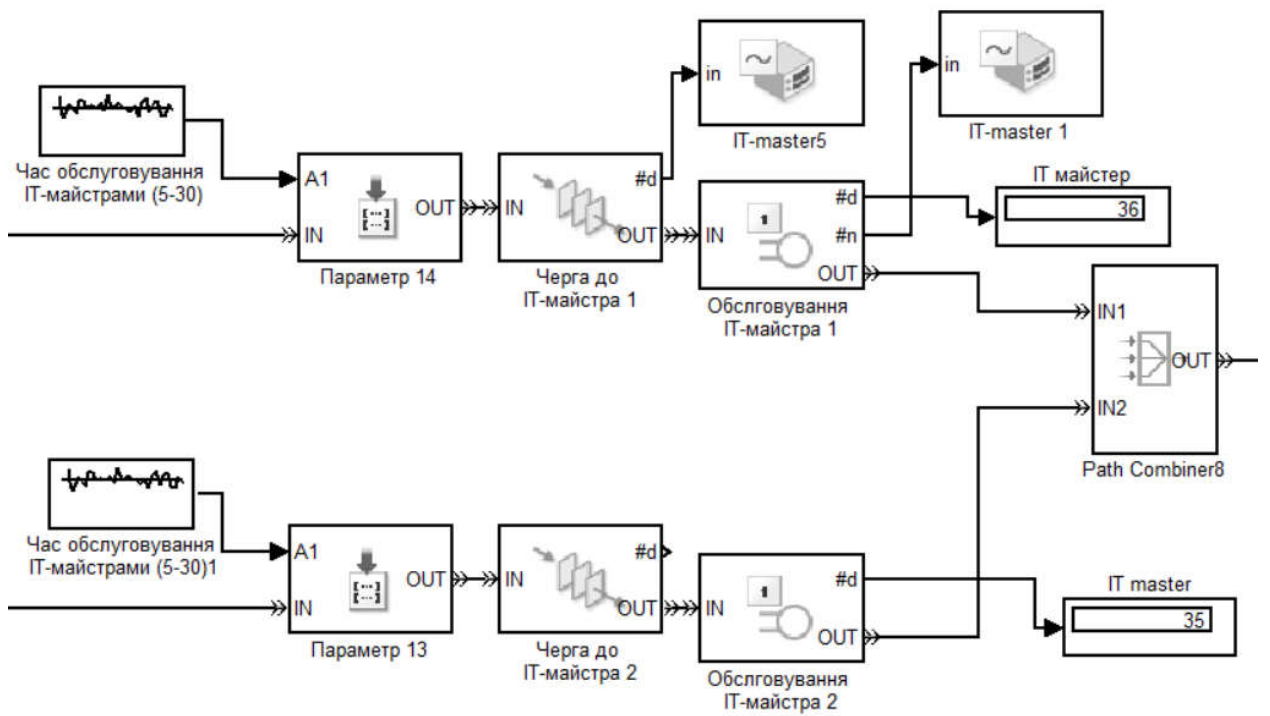


Рисунок 3.7 – Структура блоків, яка імітує роботу майстрів

Другий потік заявок симулює постійне надходження в магазин нового товару. Його обслуговування є простішим, ніж обслуговування першого потоку, оскільки заявки другого потоку минають тільки два канали обслуговування. Його повна схема подана на рисунку 3.8.

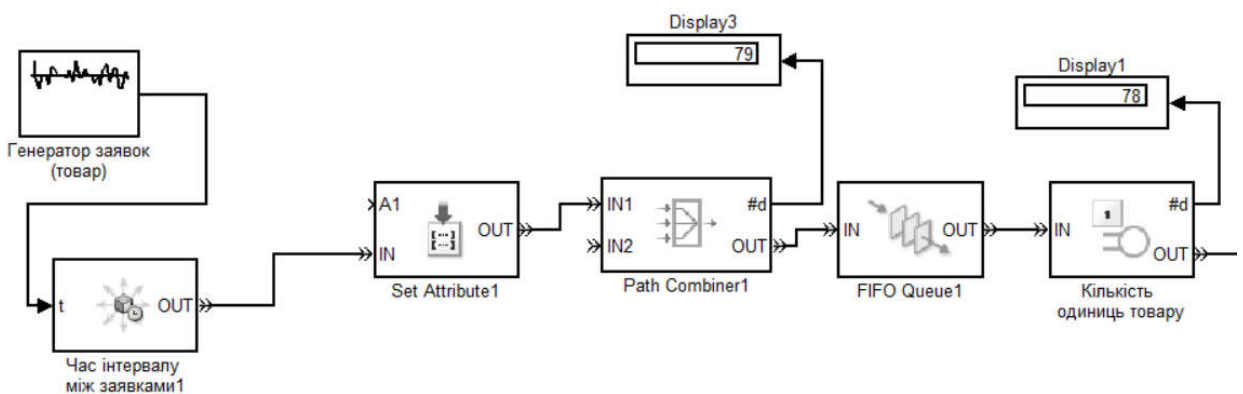


Рисунок 3.8 – Схема генерування обслуговування другого потоку заявок (товару)

Набір поданих та втрачених запитів являє собою вихідний потік QS.

Залежно від структури вихідного потоку виділяється QS без втрат та QS з втратами. Без втрати QS характеризується відсутністю обмежень на кількість місць у черзі та на час, коли додаток знаходиться в системі. З цієї причини вихідний потік складатиметься лише з поданих програм.

У свою чергу, потік загублених запитів може складатися з потоку відхиленних програм, а також від потоку не "терплячих" програм, які залишили систему, оскільки час перебування їх перевищив допустиму величину.

Вихідний потік, як правило, розподіляється на потік обслуговування та потік втрачених претензій.

### 3.3 Верифікація, тестування та захист моделі

Перевірка імітаційної моделі системи масового обслуговування передбачає перевірку її адекватності по відношенню до її реальних аналогів. У цьому випадку треба брати до уваги достовірні факти та ввести відповідні реальні дані. У цьому випадку не складне завдання обчислити, які результати слід видати системою на основі вхідних даних, і, якщо комп'ютер не виробляє ці результати, повідомити про помилку в моделі.

Отже, як зазначено раніше, всі вхідні дані були зібрані в ході досліджень, які включали довготривале спостереження за реальним аналогом імітованої системи, тому немає ніяких сумнівів у їх надійності.

У середньому близько десяти годин роботи в магазині відвідують близько 800 потенційних клієнтів. З них 15% – покупці, заявки в сервісній системі. Цей параметр береться від вимоги перетворити працівників на реальний об'єкт дослідження. Знову ж таки, варто нагадати, що система не враховує відвідувачів, які не зацікавлені в обслуговуванні. Згідно з цими

даними, ми розраховуємо, що потенційний клієнт з'являється в магазині в діапазоні від 4,5 до 9 хвилин. Ці дані вказуються в генераторі детерміністичних додатків.

Що стосується касирів, то вони, відповідно до правил установи, повинні обслуговувати клієнтів від 2 до 20 хвилин залежно від потреб самих покупців. Максимальна кількість очей – 4 особи. На малюнку 3.9 показані відповідні параметри блоку випадкових чисел на основі подій, що відповідає за створення випадкового значення інтервалу обслуговування з заданими межами на прикладі першого касира.

Межі інтервалу обслуговування продавця одного клієнта також регулюються офіційними матеріалами установи.

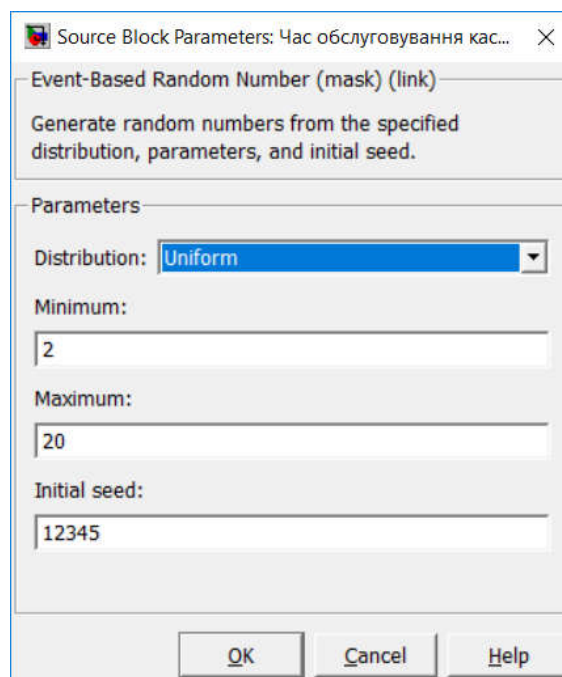


Рисунок 3.9 – Параметри генерування випадкового значення інтервалу часу із заданими межами для касирів

Аналогічно задаються значення часу обслуговування і для продавців. Вікно із параметрами подано на рисунку 3.10.

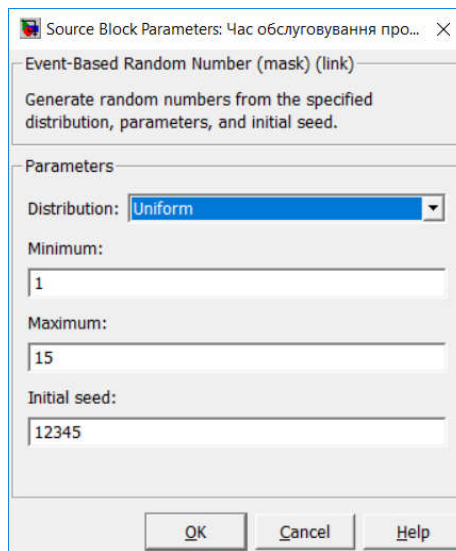


Рисунок 3.10 – Аналогічно встановлюємо параметри часу обслуговування для продавців

Максимальна кількість клієнтів у черзі – 8 осіб. Ця цифра пов'язана з розмірами самого магазину та менталітетом покупців.

Аналогічним чином вказується відповідні вищезазначені параметри для другого джерела та потоку запитів.

Згідно з розрахунками, система буде працювати з результатом, близьким до ідеалу, якщо виконати всі перераховані вище умови. Коли всі вхідні дані та параметри перевіряються, а сама модель перевіряється на наявність помилок, запускається симуляція тривалістю 720 одиниць (для імітації повного робочого дня).

Панель інструментів Simulink, де ми вказуємо тривалість моделювання у спеціальному вікні та де розташована кнопка запуску, показана на малюнку 3.11.

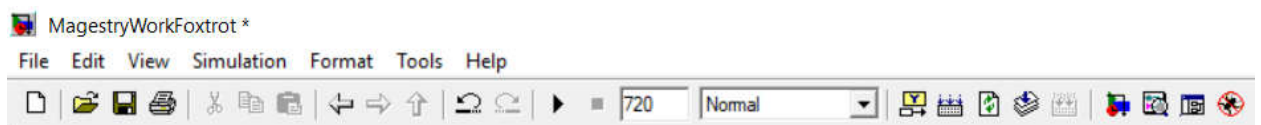


Рисунок 3.11 – Інтерфейс панелі інструментів Simulink

Після завершення симуляції перевіряються та аналізуються вихідні результати.

На рисунку 3.12 зображено числові дані, зібрані першими каналами обслуговування.

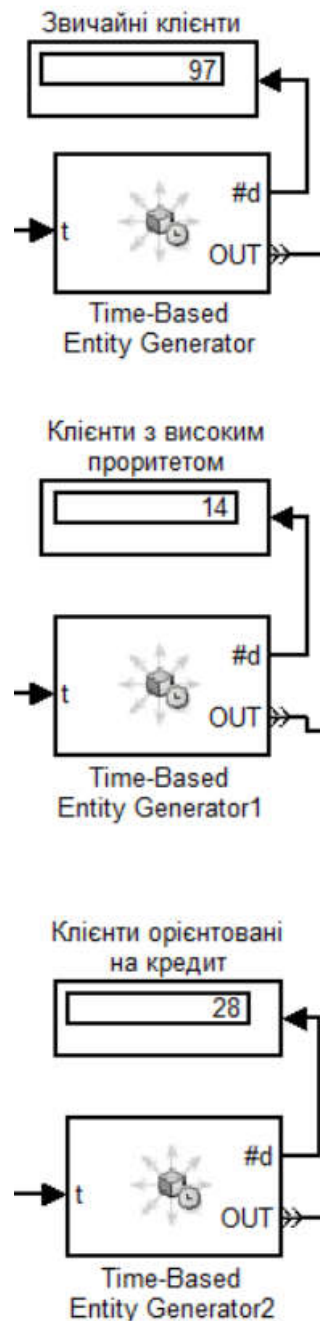


Рисунок 3.12 – Числові дані першого каналу обслуговування

Протягом 720 хвилин роботи моделі магазин відвідали 139 потенційних клієнтів.

Це означає, що нові покупці прийшли з інтервалом в 5,17 хв, що відповідає параметрам інтервалу генератора першого потоку замовлень. Далі, на малюнку 3.13 показана блок-схема продавців, яка демонструє процес збільшення кількості клієнтів, які залишили чергу служби.

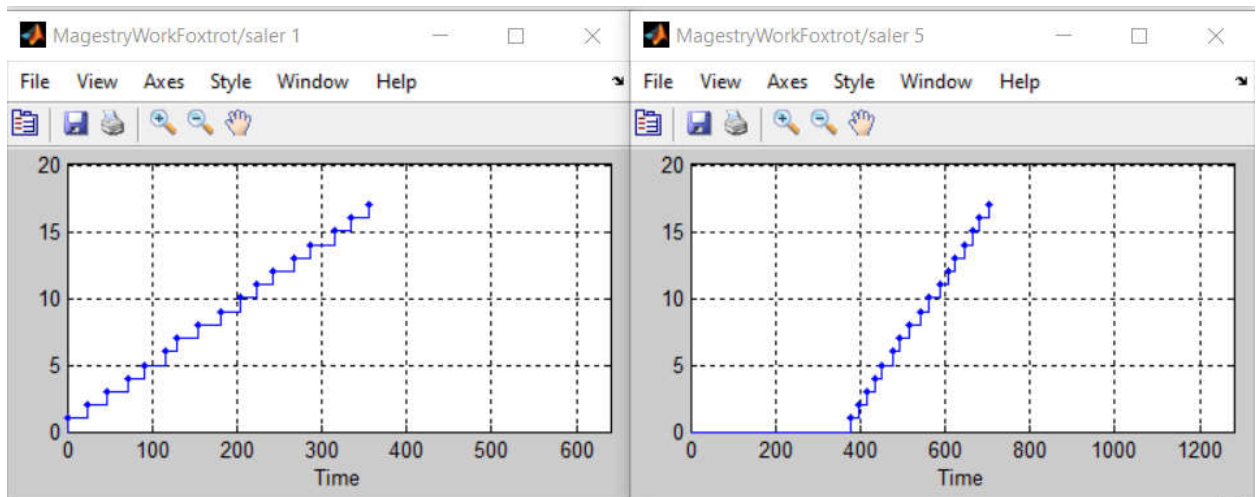


Рисунок 3.13 – Діаграми роботи продавців

Як можна визначити з діаграм, продавці обох змін змогли обслужити всіх відвідувачів магазину. Якщо перевірити числові дані їхньої ефективності послуг, то можна виділити, що перший продавець обслужив 17 запитів в першій половині дня, а п'ятий в другу половину дня також 17.

Графіки роботи першого та другого касирів, зображені на рисунку 3.14.

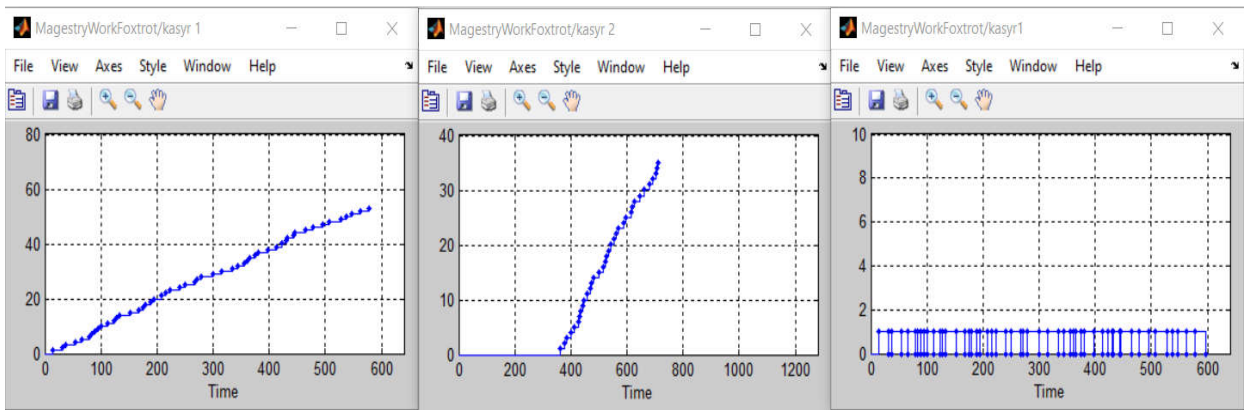


Рисунок 3.14 – Діаграми роботи касирів

З наведених діаграм, можна чітко бачити, що завдяки доволі невеликому середньому часу обслуговування клієнтів касирами, максимальна черга до них складалась всього із двох людей. Касири спільними зусиллями змогли обслужити тих клієнтів, яких обслужили перед ними продавці та брокери.

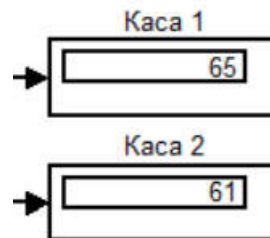


Рисунок 3.16 – Кількість обслужених заявок касирами

Виходячи із цих даних визначаємо, що загалом канали обслуговування минуло 126 заявок. Результати ефективності обслуговування третім етапом – ІТ-майстрами у графічному представленні наведено на рисунку 3.17.

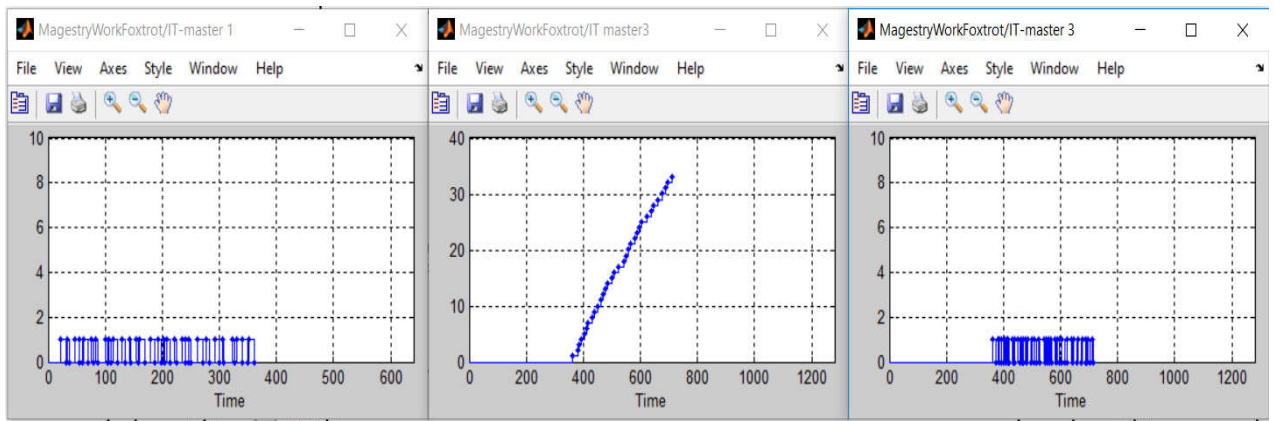


Рисунок 3.17 – Результати обслуговування ІТ-майстрами

Знову ж таки, завдяки регламентованим та ефективним параметрам неважко побачити, що максимальний розмір черги до майстрів складало всього двоє людей. При тому майстри також змогли обслужити всі 126 заявок, які надійшли до них після обслуговування касою.

Отже, на виході є 126 обслужених заявок із 139. А це свідчить про 91,09% відсотків ефективності системи обслуговування першого потоку заявок.

Тепер розглянемо другий потік.

Згідно результатів спостережень, за один день в магазин могло прийти від 40 до 100 ящиків товару. Отже, інтервалом генерування заявок беремо від 6 до 15, як і показано на рисунку 3.18.



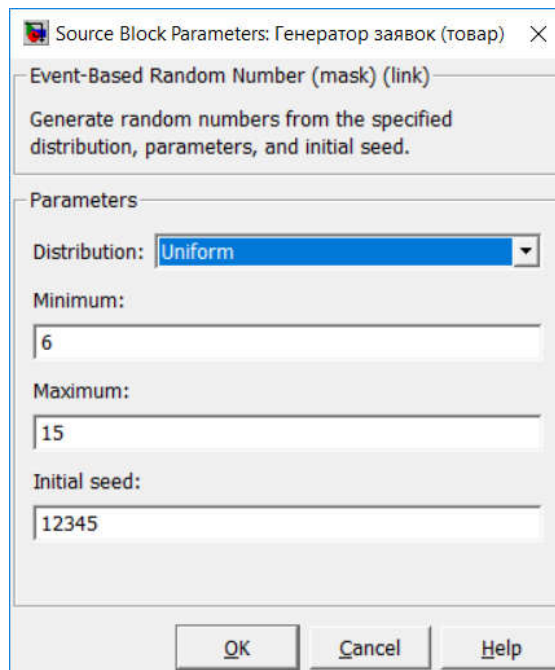


Рисунок 3.18 – Параметри інтервалу генерування заявок другого потоку

Як бачимо з результатів симуляції, система згенерувала 68 заявок, як і показано на рисунку 3.19.

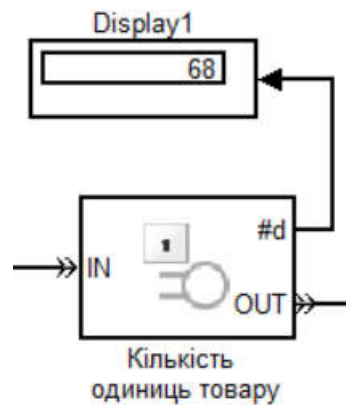


Рисунок 3.19 – Числовий показник каналу обслуговування, який відповідає за приймання товару

Тепер розглянемо діаграми та числові значення обох каналів обслуговування другого потоку (двох вантажників), які подано на рисунку 3.20.

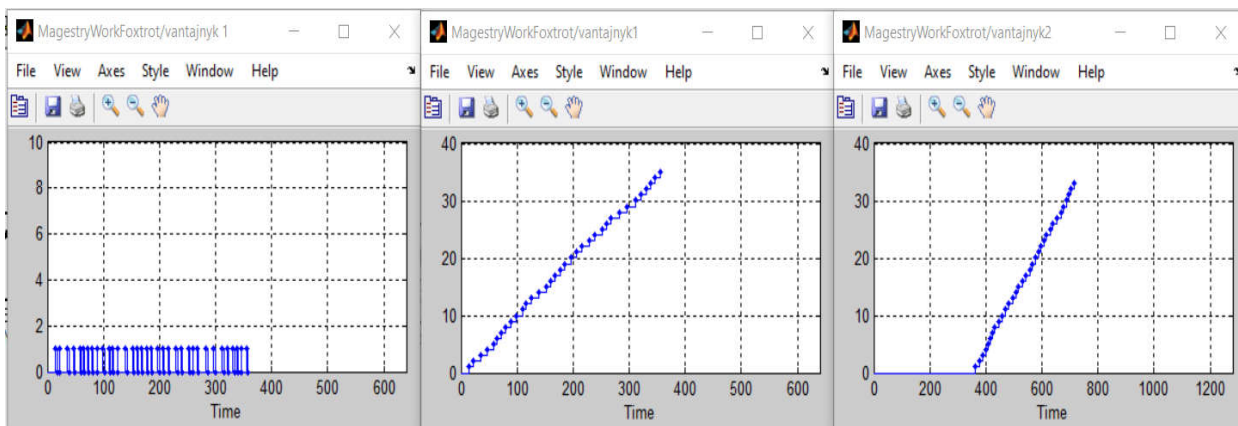


Рисунок 3.20 – Діаграми каналів обслуговування вантажників

Другий та третій графіки показують крок за кроком, як розвивається з часом та функціонує канал обслуговування нових одиниць товару. Перший графік – процес прибуття товару. Час служби вантажником однієї одиниці після прибуття не вказано, оскільки цей показник виходить за межі розробленої моделі, оскільки це стосується об'єктів, які не пов'язані з самою моделлю. Проте вказано часовий інтервал, за який вантажник повинен розпакувати одиницю товару, тобто обробляти одну заявку. У середньому, це від 1 до 10 хвилин. Ці дані показані на рисунку 3.21.

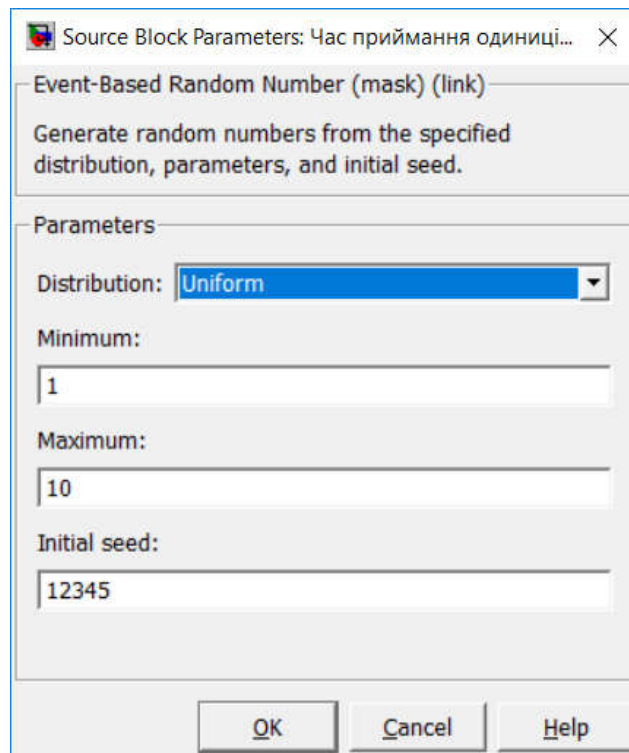


Рисунок 3.21 – Параметри інтервалу часу обслуговування одиниці товару

У результаті обслуговування другого потоку двома каналами отримуємо 68 обслужених заявок із 68, а це свідчить про 100% ефективності обслуговування другого потоку заявок.

Вирахуємо середнє арифметичне показників ефективності обслуговування першого та другого потоку заявок. І в результаті отримаємо середнє значення ефективності розробленої імітаційної моделі системи масового обслуговування.

Отже, середнє значення ефективності розробленої імітаційної моделі системи масового обслуговування – 95,5%. Цей кінцевий показник свідчить про успішність даної розробки та вигідність розроблення її реального прототипу.

Проведені дослідження підтверджують ефективність та працездатність розробленої СМО.

## ВИСНОВКИ

В процесі розробки розроблено імітаційну модель системи масового обслуговування магазину комп'ютерної техніки, яка включає:

1. вісімнадцять сервісних каналів і чотири потоки додатків. Така структура моделі дозволяє застосовувати її до складів із складною структурою, мінімально змінюючи параметри залежно від індивідуальних характеристик установи;

2. крім того, простота структури дозволяє швидко і без ключових змін самої моделі адаптувати її до моделювання сховища будь-яких товарів чи підприємств, діяльність яких подібна;

3. одночасно, параметри розробленої моделі в таких випадках також зазнають мінімальних змін, що дозволяє зберігати тестові показники ефективності та продуктивності;

4. у магазині комп'ютерної техніки з аналогічною структурою, яку описує модель, система може використовуватися без змін параметрів та структури;

5. є можливість модифікації розробленої моделі.

Результати дипломного проектування плануються до використання на підприємстві ТОВ «САВ Дістрибьюшн».

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ORCAD 9.x ORCAD 10.x. Практика моделирования / Ю.И. Болотовский, Г.И. Таназлы. – М.: СОЛОН-Пр., 2012. – 208 с.
2. Автоматизированные информационные технологии в экономике. / Под общ. ред. И.Т.Трубилина. – М.: Финансы и статистика: 2000.
3. Виктимологическое моделирование в криминологии и практике предупреждения преступности: Монография / Т.В. Варчук. – М.: ЮНИТИ-ДАНА, 2012. – 239 с.
4. Закономерности деформирования почв: Математическое моделирование / Д.И. Золотаревская. – М.: ЛИБРОКОМ, 2013. – 144 с.
5. Имитационное моделирование: Учебное пособие / Н.Б. Кобелев, В.А. Половников, В.В. Девятков. – М.: КУРС, НИЦ ИНФРА-М, 2013. – 368 с.
6. Интерьер: дизайн и компьютерное моделирование. / Д.А. Ларченко, А.В. Келле-. – СПб.: Питер, 2011. – 480 с.
7. Информатика. Базовый курс” Под ред. С.В.Симоновича. – СПб., 2000.
8. Исследование операций в экономике / Под ред. Н.Ш. Кремера. – М.: Банки и биржи, ЮНИТИ, 1997.
9. Исследование операций. Задачи, принципы, методология: Учебное пособие для студентов вузов. / Вентцель Е.С. – М.: Высш. шк., 2001. – 206 с.
10. Карта моделирования будущего. Как найти истинный смысл своей судьбы и создать новую реальность / К. Бейрон-. – М.: Эксмо, 2013. – 304 с.
11. Компьютерное моделирование / А.Л. Королев. – М.: БИНОМ. ЛЗ, 2013. – 230 с.

12. Компьютерное моделирование нанотехнологий, наноматериалов и наноструктур: моделирование наносистем методами молекулярной динамики: Курс лекций / С.Ю. Юрчук. – М.: МИСиС, 2013. – 47 с.
13. Компьютерное моделирование физических систем: Учебное пособие / Л.А. Булавин, Н.В. Выгорницкий, Н.И. Лебовка. – Долгопрудн: Интеллект, 2011. – 352 с.
14. Компьютерные технологии обработки информации / Под. ред. С.В.Назарова. – М.: Финансы и статистика, 1995.
15. Лычкина, Н.Н. Имитационное моделирование экономических процессов: Учебное пособие / Н.Н. Лычкина. – М.: НИЦ ИНФРА-М, 2012. – 254 с.
16. Математика в экономике: математические методы и модели: Учебник для бакалавров / М.С. Красс, Б.П. Чупрынов; Под ред. М.С. Красс. – М.: Юрайт, 2013. – 541 с.
17. Математические методы и модели в экономике: Учебник / К.В. Балдин, В.Н. Башлыков, А.В. Рукосуев. – М.: Флинта, МПСИ, 2012. – 328 с.
18. Математические методы и модели исследования операций: Учебник / А.С. Шапкин, В.А. Шапкин. – М.: Дашков и К, 2013. – 400 с.
19. Математические методы построения стохастических моделей обслуживания. / Калашников В.В. – М.: Наука, 1988. – 310 с.
20. Математическое и компьютерное моделирование. Вводный курс: Учебное пособие / Ю.Ю. Тарасевич. – М.: ЛИБРОКОМ, 2013. – 152 с.
21. Математическое моделирование многоагентных систем конкуренции и кооперации (Теория игр для всех): Учебное пособие / В.Н. Колокольников, О.А. Малафеев. – СПб.: Лань, 2012. – 624 с.
22. Математическое моделирование производственного и финансового менеджмента: Учебное пособие / В.Н. Савиных. – М.: КноРус, 2013. – 192 с.

23. Математическое моделирование экономических процессов и систем: Учебное пособие / О.А. Волгина, Н.Ю. Голодная, Н.Н. Одияко. – М.: КноРус, 2012. – 200 с.
24. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «Магістр» підготовки 6.050102 комп'ютерна інженерія / І.Р. Паздрій – Тернопіль: ТНЕУ, 2014. – 37 с.
25. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Магістр” напряму підготовки 6.050102 «Комп'ютерна інженерія» фахового спрямування «Комп'ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Р.Б. Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. – Тернопіль: ТНЕУ, 2014. – 65с.
26. Методы и модели оптимизации управленческих решений: Учебное пособие / А.Р. Урубков, И.В. Федотов. – М.: ИД Дело РАНХиГС, 2012. – 240 с.
27. Моделирование бизнес-процессов. / Маклаков С.В. – М: Диалог МИФИ, 2002.
28. Моделирование микроэкономических процессов и систем: Учебник / Л.Н. Васильева, Е.А. Деева. – М.: КноРус, 2012. – 392 с.
29. Моделирование систем: Учебно-практическое пособие / В.В. Афонин. – М.: БИНОМ. ЛЗ, ИНТУИТ, 2012. – 231 с.
30. Моделирование экономических систем и процессов: Учебное пособие / М.П. Власов, П.Д. Шимко. – М.: НИЦ ИНФРА-М, 2013. – 336 с.
31. Основы теории математического моделирования: Учебное пособие / А.И. Барботько, А.О. Гладышкин. – Ст. Оскол: ТНТ, 2013. – 212 с.
32. Системное проектирование сложных программных систем. / Липаев В.В. – М: СИНТЕГ, 1998.
33. Теоретические основы разработки и моделирования систем автоматизации: Учебное пособие / А.М. Афонин, Ю.Н. Царегородцев,

А.М. Петрова, Ю.Е. Ефремова. – М.: Форум, 2011. – 192 с.

34. Теория массового обслуживания в экономической сфере: Учебное пособие для вузов. / Лабскер Л.Г. – М.: Банки и биржи: ЮНИТИ, 1998. – 319 с.

35. Теория массового обслуживания. / Ивановский В. Б., Чернов В. П. – М.: ИНФРА-М, 2000.

36. Теория очередей и управление запасами. / Рыжиков Ю. И. – СПб: 2001.