

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Імітаційна модель виробничої системи у програмному середовищі FlexSim»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студентка групи ІТ-72 Ніколаєнко Кароліна Олександрівна

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**
2021 р.

_____ «____» _____

Науковий керівник

(підпис)

к.т.н., доц., Чибіряк Я. І.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2021

Сумський державний університет
 Факультет електроніки та інформаційних технологій
 Кафедра комп'ютерних наук
 Секція інформаційних технологій проектування
 Спеціальність 122 «Комп'ютерні науки»
 Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик

«__» _____ 2021 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Ніколаєнко Кароліна Олександрівна

1 Тема роботи Імітаційна модель виробничої системи у програмному середовищі FlexSim

керівник роботи Чибіряк Яна Іванівна, к.т.н., доцент,

затверджені наказом по університету від « 14 » квітня 2021 р. №0181-VI

2 Строк подання студентом роботи «7» червня 2021 р.

3 Вхідні дані до роботи _____
 технічне завдання на розробку імітаційної моделі виробничої системи

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

аналіз предметної області, моделювання та проектування імітаційної моделі, програмна реалізація імітаційної моделі

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
актуальність роботи, постановка задачі, порівняння програмних середовищ моделювання, вимоги до моделі, проектування імітаційної моделі, програмна реалізація імітаційної моделі, дослідження виробничої системи, висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.10.2020

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	01.02.2021 – 20.02.2021	
2	Написання технічного завдання	22.02.2021 – 01.04.2021	
3	Визначення методів і засобів реалізації	02.04.2021 – 07.04.2021	
4	Планування робіт	08.04.2021 – 21.04.2021	
5	Моделювання та проектування	19.04.2021 – 28.04.2021	
6	Програмна реалізація	29.04.2021 – 18.05.2021	
7	Перевірка гіпотез	19.05.2021 – 24.05.2021	
8	Написання документації	25.05.2021 – 07.06.2021	

Студент

(підпис)

Ніколаєнко К.О.

Керівник роботи

(підпис)

к.т.н., доц. Чибіряк Я.І.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Імітаційна модель виробничої системи у програмному середовищі FlexSim».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 15 найменувань, додатків. Загальний обсяг роботи – 83 сторінки, у тому числі 60 сторінок основного тексту, 2 сторінки списку використаних джерел, 19 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці імітаційної моделі виробничої системи у середовищі FlexSim. В роботі проведено аналіз предметної області, планування робіт, написано технічне завдання. У роботі виконано моделювання, проектування та розробка імітаційної моделі, дослідження виробничої системи, перевірка гіпотез. Результатом проведеної роботи є імітаційна модель виробничої системи та інтерфейс користувача. Практичне значення роботи полягає у наданні практичних рекомендацій щодо покращення роботи виробничої системи.

У першому розділі наведено аналіз предметної області виробничої промисловості та моделювання технічних систем, проведено порівняння програмних продуктів-аналогів як засобів реалізації, деталізовано мету та задачі проекту.

У другому розділі описано етап моделювання та проектування імітаційної моделі, побудовано структурно-функціональні моделі процесів побудови та використання моделі у нотації IDEF0, концептуальна модель системи OFD, проведено моделювання бази даних.

В третьому розділі описано процес програмної реалізації проекту, розробки бази даних, імітаційної моделі та інтерфейсу користувача.

Ключові слова: імітаційна модель, виробнича система, гіпотеза, інтерфейс користувача.

ЗМІСТ

ВСТУП.....	7
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Аналіз методів моделювання систем	9
1.2. Аналіз програмних продуктів-аналогів	11
1.3. Аналіз підходів імітаційного моделювання	12
1.4. Постановка задачі	14
2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІМІТАЦІЙНОЇ МОДЕЛІ	16
2.1. Проектування імітаційної моделі	16
2.2. Побудова концептуальної моделі системи.....	21
2.3. Проектування моделі бази даних	23
3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІМІТАЦІЙНОЇ МОДЕЛІ	25
3.1. Структуризація даних.....	25
3.2. Програмна реалізація імітаційної моделі	32
3.3. Дослідження роботи виробничої системи.....	58
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61
Додаток А.....	63
Додаток Б.....	74

Перелік умовних позначень

БД – база даних

ВС – виробнича система

ІМ – імітаційна модель

ТЗ – технічне завдання

GUI – графічний інтерфейс користувача

ВСТУП

Економіка є одним з найважливіших аспектів розвитку будь-якої країни. За рівнем та обсягом промислового виробництва визначається її конкурентоспроможність на світовому ринку. Потреба в ефективній обробній промисловості зростає, оскільки витрати на матеріали, транспортування і робочу силу збільшуються з кожним роком. Успішним компаніям необхідно враховувати й оптимізувати витрати на виробництво. Розвиток суспільства у сфері інформаційних технологій зумовив появу такого поняття, як четверта промислова революція, для якої все більше використовується термін “Індустрія 4.0”. Німеччина, як світовий лідер у секторі виробничого обладнання та одна з країн з найрозвиненішою галуззю виробництва, визначила Індустрію 4.0 як новий тип індустріалізації та можливість зміцнити свою позицію як постачальника обладнання та ІТ бізнес рішень [1].

Однією з основних тенденцій Індустрії 4.0 є цифровий двійник. Цифровий двійник – цифрова модель об’єкту або процесу, побудована на основі отриманої інформації з датчиків, встановлених на фізичному об’єкті, що дозволяє за допомогою симуляції у реальному часі оптимізувати ефективність системи. Концептуальна модель цифрового двійника включає три компоненти, одним з яких є модель системи у віртуальному просторі [2], яка може бути реалізована у вигляді імітаційної моделі.

Моделювання виробництва є недорогим і безпечним способом тестування і прогнозування поведінки системи будь-якої складності при невеликих змінах чи повній модернізації для досягнення цілей виробництва з мінімальними витратами [10], [16].

На сьогодні галузь виробництва в Україні недостатньо розвинена і не може конкурувати з продуктами аналогами інших країн [3], [4]. Відсутність новітніх технологій, технічного переоснащення, висококваліфікованого персоналу стало причиною низького рівня розвитку виробничої промисловості в Україні. Тому наразі

є гостра потреба удосконалювати виробничі системи, що завжди супроводжується невизначеністю та значними ризиками. Високорозвинені країни для вирішення такої складної проблеми використовують імітаційне моделювання, що найбільше підходить для створення складних систем з багаторівневою організацією, великою кількістю елементів, взаємозв'язків між ними, відтворення процесу у часі з урахуванням випадкових подій та дискретних величин.

Мета – розробити імітаційну модель виробничої системи та виконати перевірку гіпотез щодо підвищення ефективності її роботи.

Об'єкт дослідження – імітаційне моделювання виробничої ділянки.

Предмет дослідження – вплив параметрів виробничої системи на показники ефективності її роботи.

Побудована модель може бути використана для дослідження роботи виробничих систем [8], [9], визначення стратегій оптимізації виробництва та надання практичних рекомендацій щодо його покращення.

Для досягнення поставленої мети були поставлені такі задачі:

- провести аналіз предметної області;
- обґрунтувати вибір методів дослідження систем ;
- обрати програмні засоби реалізації системи
- побудувати імітаційну модель виробничої системи;
- виконати перевірку гіпотез щодо покращення функціонування системи;
- надати практичні рекомендації по підвищенню ефективності роботи системи.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз методів моделювання систем

Фізичний експеримент – один з перших методів моделювання, але активно використовується по цей день. Фізична модель – це відтворення досліджуваної системи зі збереженням її фізичних властивостей.

Важливим етапом створення авіаційної техніки, комплексів, систем є проведення випробувань моделі.

Переваги:

- Мають велику точність відносно реального об'єкту та надають найбільш повну, точну інформацію про систему.

Недоліки фізичного моделювання:

- Проведення експериментів з реальним об'єктом є ризикованим.
- Створення прототипу може бути затратним.
- Великі витрати часу на моделювання та проведення експериментів над великими та складними об'єктами.

Аналітичне моделювання полягає у математичному описі об'єкта. Модель подається у вигляді функціональних залежностей початкових та кінцевих параметрів. Рішення найчастіше представлене у вигляді перетворення Лапласа або системи диференціальних рівнянь. Як правило аналітична модель є статичною.

Переваги:

- Простота моделі.
- Метод є найкращим для знаходження оптимального рішення.

Недоліки аналітичного моделювання:

- Дозволяють аналізувати малу кількість факторів.

- Значні спрощення, що можуть призвести до неточності результатів.
- Системи високої складності важко аналізувати.
- Неможливо описати паралельні процеси.
- Складно визначити вплив мінливості параметрів моделі.

Імітаційне моделювання дозволяє відтворювати процеси у часі, що дозволяє по вхідним даним визначити стан системи у майбутньому. Нині цей метод є найбільш ефективним, іноді єдиним для дослідження систем на етапі їх проектування [11], [12], [13].

Метод імітаційного моделювання надає можливість описати різні елементи системи та взаємодію між ними. Система може перебувати у різних станах, які залежать від певних параметрів, зміна яких імітує перехід з одного стану в інший [12]. Таким чином, імітаційна модель – динамічне зображення змін станів системи з плином часу.

В основному для моделювання використовуються підхід випадкових величин та багаторазове відтворення процесу. У результаті отримуються статистичні дані, що характеризують функціонування системи [14].

Переваги:

- Витрати на побудову імітаційної моделі обмежені вартістю програмного забезпечення та деяких послуг.
- У порівнянні з фізичною моделлю, імітаційна здатна надати результати за коротший час.
- Варіативність значень параметрів системи дає можливість проводити велику кількість експериментів.
- Імітаційне моделювання дозволяє зобразити систему, у никаючи математичних формул і залежностей.
- Графічне зображення структури та процесів роботи системи більш наочне і просте для розуміння.
- Універсальний метод моделювання будь-яких сфер життя: виробництво, охорона здоров'я, фінансування та ін.

Недоліки:

- Імітаційна модель не завжди допомагає вирішити поставлені питання.
- Моделювання великих і складних систем може зайняти багато часу.
- Модель не є абсолютно точною копією реальної системи.
- Імітаційне моделювання не дає такої точності, як математичне.

1.2. Аналіз програмних продуктів-аналогів

Для аналізу програмних продуктів, за допомогою яких можна побудувати імітаційну модель виробничої системи, було створено порівняльну таблицю середовищ моделювання, які спрямовані на моделювання бізнес процесів (табл 1.1).

Таблиця 1.1 – Порівняльна таблиця продуктів-аналогів

Програма Функція	AnyLogic	FlexSim	AweSim	Arena	ExtendSim	Enterprise Dynamics	Simul8
Візуалізація у 3D вигляді	+	+	+	+	+	+	+
Відображення статистичних даних у графічному вигляді	+	+	+	+	+	+	+
Інтеграція з базами даних	+	+	+(Dbase, Access, FoxPro, Excel)	+(FoxPro, Excel, Oracle, Informix, Access та ін.)	+(внутрішня БД, ADO)	+(Industry Specific Database)	+(Excel, SQL)
Використання мов програмування	+(Java, Python)	+(C++)	+(C, Visual Basic)	+(Visual Basic)	+(ModL)	+(4DScript)	+(Visual Logic)

Продовження таблиці 1.1

Програма	AnyLogic	FlexSim	AweSim	Arena	ExtendSim	Enterprise Dynamics	Simul8
Функція							
Запуск моделі на WEB-сервері	+	+ (локально)	-	-	+	+	+
Створення інтерфейсу користувача GUI	-	+	-	-	-	-	-

Середовище моделювання FlexSim зручне у використанні для моделювання та аналізу. Процес побудови моделей відбувається у 3D просторі, що візуально зручніше і значно полегшує процес верифікації [10]. Інтерфейс програми та інструмент Process Flow дозволяють задати логіку складної моделі.

Середовище FlexSim має інструмент створення інтерфейсу користувача, що дає змогу створити окремий модуль зміни параметрів та вхідних даних моделі.

Програмний модуль ExpertFit, вбудований у FlexSim, призначений для визначення закону розподілу на основі вхідних даних у вигляді вибірки чисел.

Тому для моделювання системи виробництва було обране середовище FlexSim.

1.3. Аналіз підходів імітаційного моделювання

Існує чотири основних підходи імітаційного моделювання: динамічне моделювання, системна динаміка, дискретно-подійне й агентне моделювання, метод статистичних випробувань Монте-Карло.

Динамічні системи (механічні або фізичні процеси) описуються алгебраїчними, диференційними рівняннями і блок-схемами.

Системно-динамічний підхід для дослідження динамічних процесів спрямований на вивчення складних систем зі зворотнім зв'язком. Процеси представляються у вигляді діаграми, що складається з петель позитивного та негативного зворотного зв'язку. Найбільша увага приділяється моделювання зворотних зв'язків.

У дискретно-подійному підході модельний час рухається від події до події, або через дискретні проміжки часу. Цей підхід застосовується у тому випадку, якщо змінні системи змінюються миттєво у визначені проміжки часу. При дискретно-подійному моделюванні розглядаються тільки основні події модельованої системи, наприклад очікування, обробка, рух з вантажем, розвантаження та інші. Дискретно-подієве моделювання застосовується у логістиці, системах масового обслуговування, транспортних і виробничих системах [8-14].

В агентному моделюванні досліджується поведінка агента та його вплив на функціонування системи [5]. Ключовими елементами агентного моделювання є карти станів. Цей підхід застосовується для імітації інтелектуальних, розподілених та децентралізованих систем, щоб отримати відомості про вплив функціонування і взаємодії елементів на систему.

Метод Монте-Карло – числовий метод розв'язання математичних задач за допомогою моделювання випадкових величин. Метод статистичних випробувань дозволяє моделювати будь-який процес, на протікання якого впливають випадкові фактори. Даний метод полягає у використанні генератора рівномірно розподілених випадкових чисел у зв'язку з функцією розподілення ймовірностей досліджуваного процесу [6]. Після визначення функцій на основі теоретичного, емпіричного або іншого розподілу, що відповідає характеру досліджуваного процесу, створюються випадкові вибірки – значення вхідних даних. За допомогою прогонів імітаційної моделі багато разів отримуємо відповідну множину значень вхідних і вихідних параметрів. За допомогою статистичного аналізу вихідних даних приймається рішення.

Для побудови імітаційної моделі виробничої ділянки був обраний дискретно-подійний метод.

Після вибору підходу розробляється імітаційна модель. Процес побудови складається з таких етапів [7]:

1. Точне формулювання мети дослідження.
2. Збір інформації і даних.
3. Розробка концептуальної моделі.
4. Перевірка концептуальної моделі на адекватність поставленій задачі.
5. Побудова моделі у програмному середовищі по концептуальній моделі.
6. Верифікація запрограмованої моделі.
7. Розробка, проведення та аналіз експериментів.
8. Документування та представлення отриманих результатів.

Для того, щоб досягти найбільшої точності моделі, при її розробці необхідно взаємодіяти з предметними експертами та замовниками проекту. Для більш простої верифікації та валідації використовується анімація моделі.

1.4. Постановка задачі

Метою роботи є розробка імітаційної моделі, дослідження показників ефективності роботи виробничої системи за такими параметрами, як завантаженість обладнання, кількість деталей, які оброблялись довше допустимого часу перебування у системі, прибуток від реалізації. Використовуючи побудовану модель перевірити гіпотези щодо покращення роботи виробничої системи. За результатами перевірки гіпотез надати практичні рекомендації, щодо підвищення ефективності роботи виробничої системи.

Для виконання проекту потрібно виконати такі задачі:

- провести аналіз предметної області сфери виробничої промисловості та моделювання систем, та на основі отриманої інформації визначити актуальність роботи;
- проаналізувати існуючі методи і засоби реалізації проекту, визначити ті, що будуть використовуватися для виконання проекту та обґрунтувати їх вибір;
- розробити 3D моделі верстатів, максимально наближені до реальних за розмірами та формою;
- розробити базу даних для зберігання вхідних даних до моделі;
- побудувати імітаційну модель виробничої системи відповідно до поставленої задачі, задати логіку, створити панель збору статистики та інтерфейс користувача, який даватиме змогу змінювати параметри моделі;
- виконати перевірку запропонованих гіпотез пріоритетом щодо покращення виробничого процесу: зменшення часу обробки на верстатах на 20%, додавання однотипних верстатів до системи, встановлення послідовності обробки за динамічним;
- провести аналіз отриманих результатів перевірки гіпотез, на основі яких надати практичні рекомендації по підвищенню ефективності роботи системи.

У результаті виконання проекту потрібно отримати імітаційну модель виробничої системи, що відповідає вимогам, детально описаним у технічному завданні (додаток А).

2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІМІТАЦІЙНОЇ МОДЕЛІ

2.1. Проектування імітаційної моделі

Першим етапом реалізації проекту є проектування процесів розробки імітаційної моделі (ІМ) у вигляді контекстної діаграми.

Контекстна діаграма процесу розробки імітаційної моделі описана за допомогою методології IDEF0 (рис. 2.1).

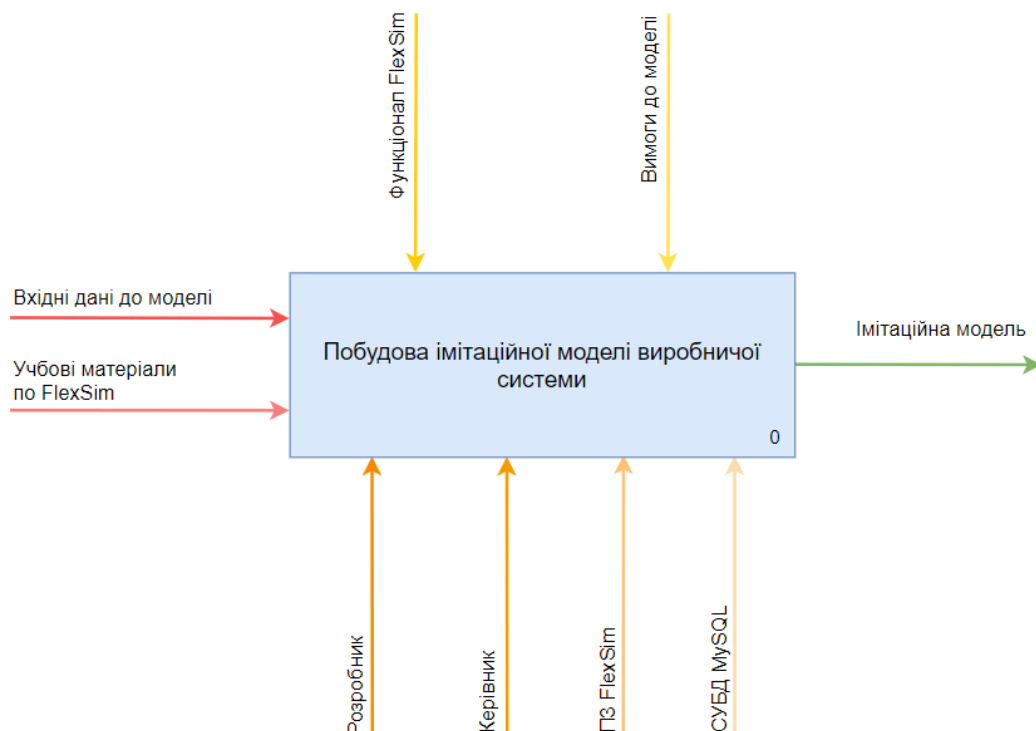


Рисунок 2.1 – Контекстна діаграма процесу розробки ІМ

Елементи діаграми процесу розробки ІМ:

- Процес: побудова імітаційної моделі ВС;

- Вхід: вхідні дані до моделі, визначені у технічному завданні (ТЗ), учбові матеріали по використанню ПЗ FlexSim;
- Управління: інтерфейс та функціонал ПЗ FlexSim, у якому реалізується ІМ, вимоги до моделі, зазначені у ТЗ;
- Механізми: розробник даного проекту, керівник проекту, ПЗ FlexSim, СУБД MySQL для розробки та управління базою даних (БД);
- Вихід: імітаційна модель, готова для використання, тобто аналізу ВС та перевірки гіпотез.

Деталізуємо основний процес концептуальної моделі розробки ІМ. Діаграма декомпозиції першого рівня у нотації IDEF представлена на рис.2.2. Процес програмної реалізації ІМ деталізований у моделі декомпозиції другого рівня (рис. 2.3).

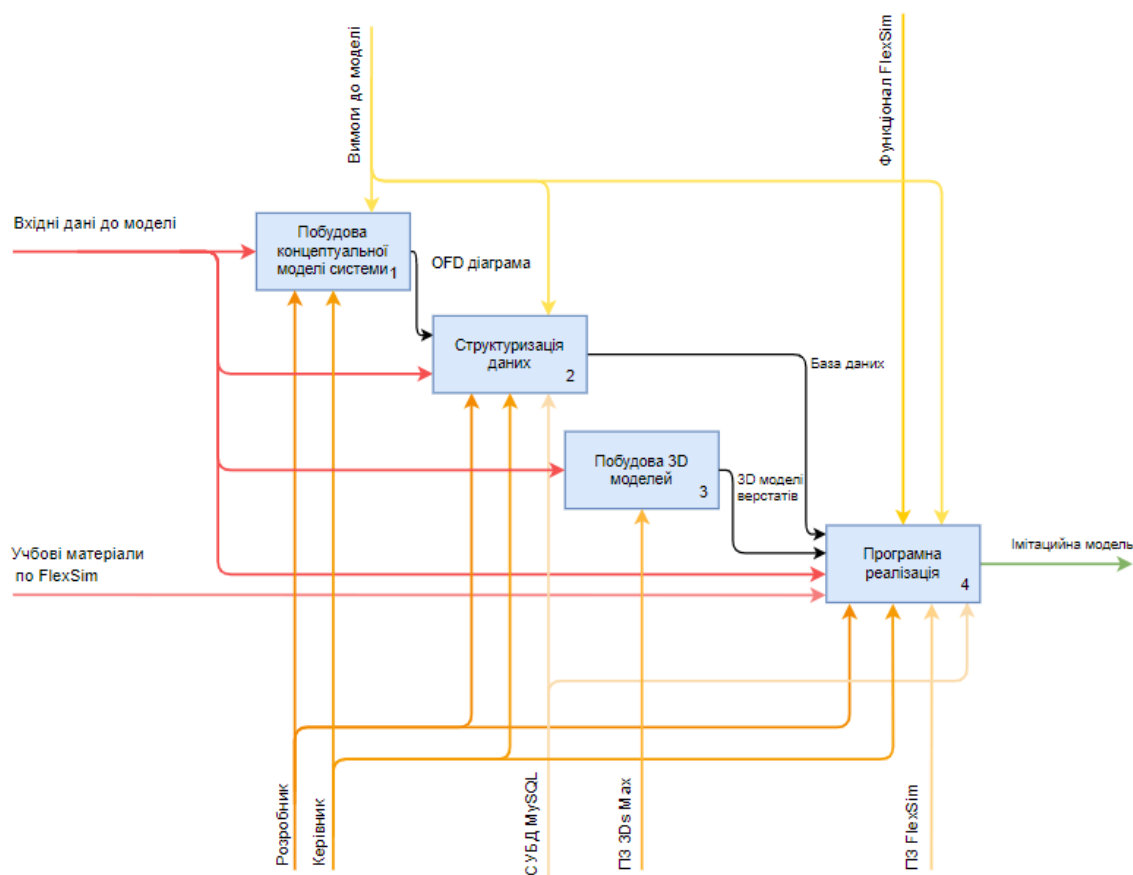


Рисунок 2.2 – Модель декомпозиції першого рівня розробки ІМ у нотації IDEF

Процеси, виділені у діаграмі декомпозиції розробки ІМ:

- Побудова концептуальної моделі системи – OFD діаграма;
- Структуризація даних – розробка БД для зберігання вхідних даних до моделі;
- Побудова 3D моделей у програмі 3Ds Max;
- Програмна реалізація – побудова імітаційної моделі відповідно вимогам.

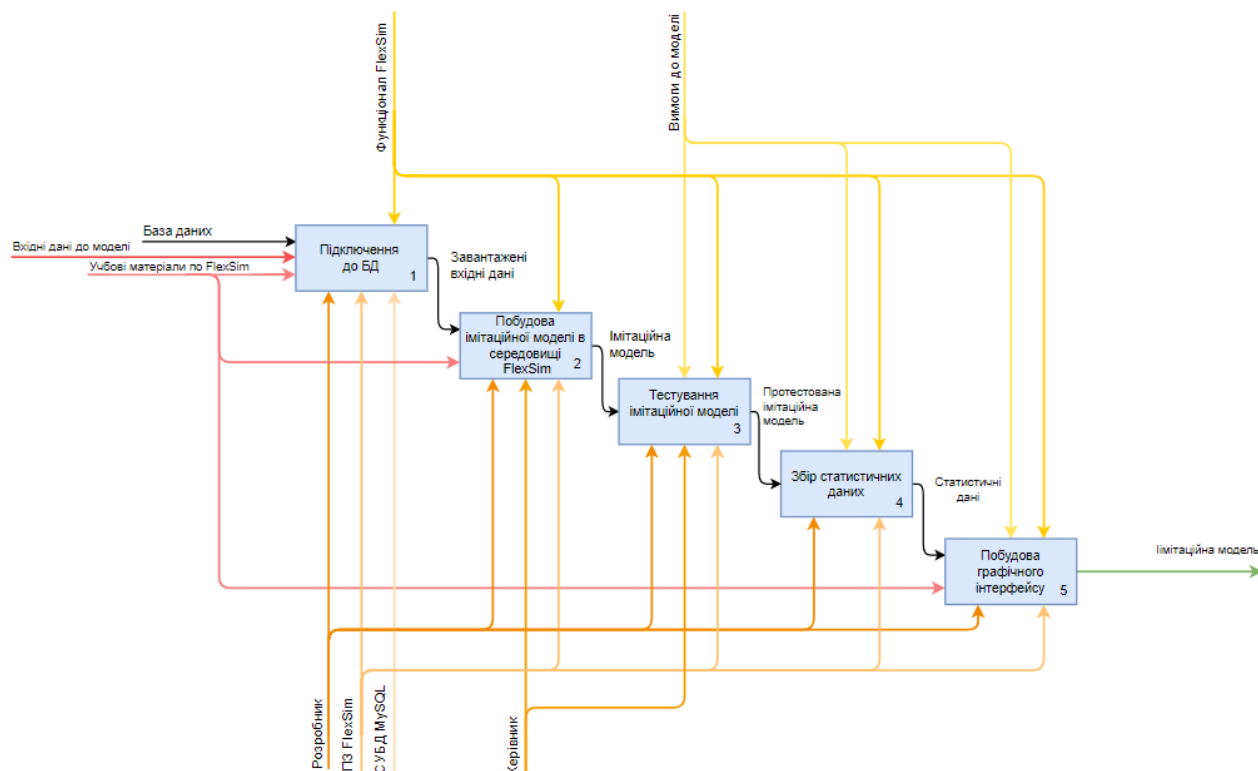


Рисунок 2.3 – Модель декомпозиції другого рівня процесу програмної реалізації ІМ

Програмна реалізація імітаційної моделі поділена такі етапи:

- Підключення до БД засобами FlexSim;
- Побудова імітаційної моделі у середовищі FlexSim – додавання відповідних елементів та налаштування логіки моделі;
- Тестування імітаційної моделі – визначення відповідності моделі вимогам та виправлення помилок;
- Збір статистичних даних – побудова панелі збору статистики засобами FlexSim;

- Побудова графічного інтерфейсу для зміни вхідних даних та параметрів моделі.

За допомогою IDEF0 опишемо функціональний аспект імітаційної моделі.

Структурно-функціональна модель процесу, який реалізується у розроблюваній моделі зображена у вигляді концептуальної моделі на рис. 2.4. Для досягнення вищого рівня деталізації основна функція ділиться на під функції, які описані у моделі декомпозиції першого рівня (рис. 2.5).

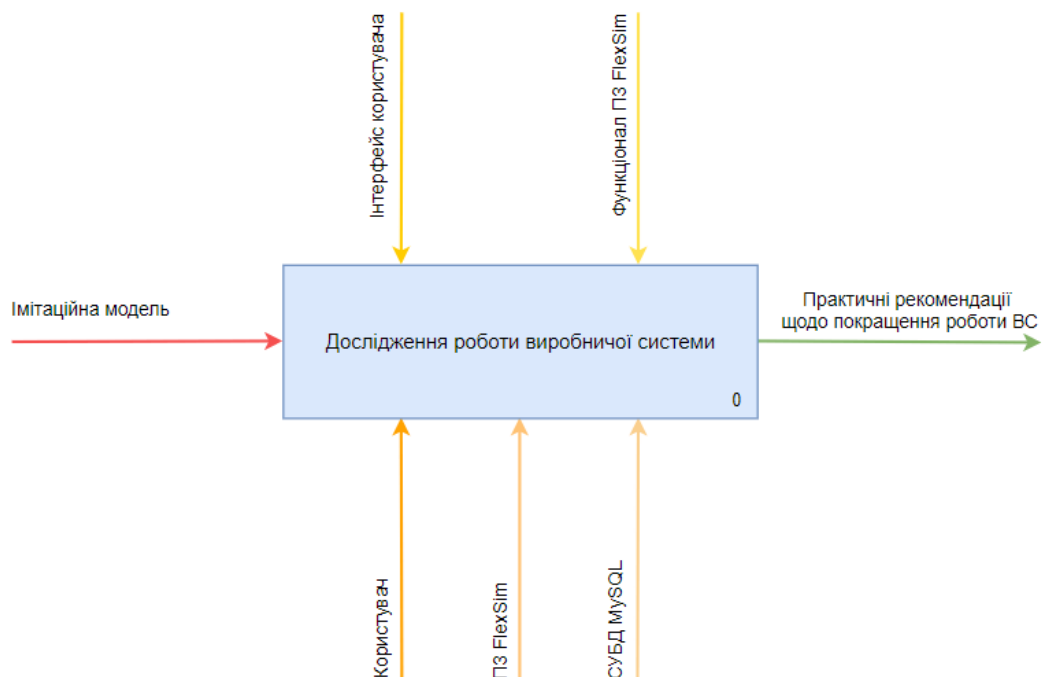


Рисунок 2.4 – Контекстна діаграма процесів, що реалізуються в ІМ

Елементи контекстної діаграми функціоналу імітаційної моделі:

- Процес: дослідження роботи виробничої системи, визначення її ефективності при заданих початкових даних;
- Вхід: імітаційна модель, готова до використання;
- Управління: функціонал ПЗ FlexSim, інтерфейс користувача, який дає змогу змінювати вхідні дані та задавати параметри;
- Механізми: користувач, ПЗ FlexSim, СУБД MySQL;

- Вихід: практичні рекомендації щодо покращення роботи ВС, надані на основі результатів перевірки гіпотез.

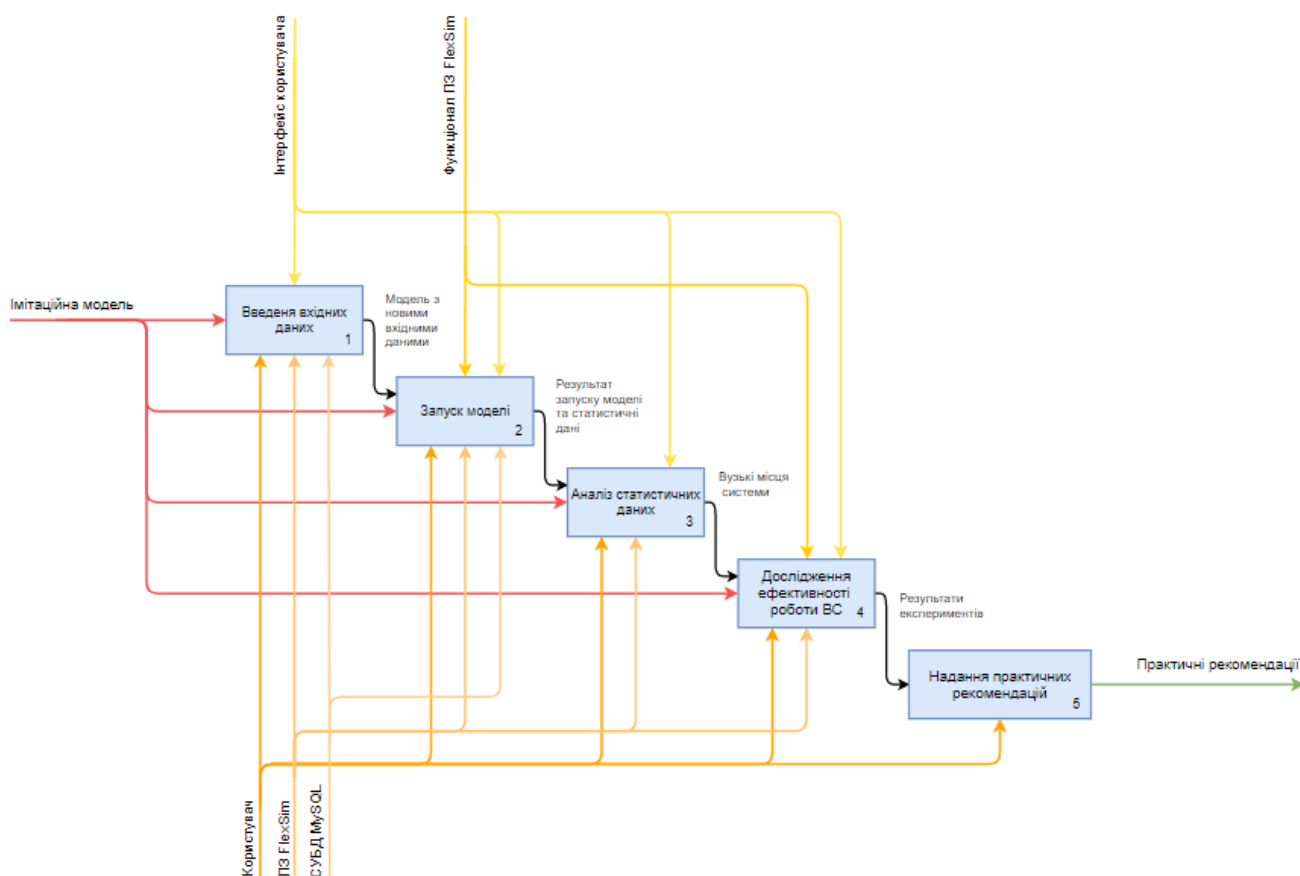


Рисунок 2.5 – Модель декомпозиції першого рівня процесів, що реалізуються в ІМ

Етапи дослідження ВС за допомогою імітаційної моделі:

- Введення вхідних даних користувачем, для дослідження роботи системи з іншими параметрами;
- Запуск моделі – проведення процесу імітації;
- Аналіз статистичних даних – проведення аналізу даних, що відображається на панелі збору статистики після запуску моделі;
- Дослідження ефективності роботи ВС – висновок про ефективність системи на основі проаналізованих статистичних даних;

- Надання практичних рекомендацій щодо покращення роботи ВС за результатами дослідження.

2.2. Побудова концептуальної моделі системи

Схематичне зображення процесу імітації показано на OFD діаграмі (рис. 2.6).

У систему надходять 4 типи заготовок деталей, які очікують на обробку у черзі BUF raw. По відділах дільниці заготовки розвозяться транспортом Т і складаються у черги BUF wait. Оператор відносить заготовку на верстат і після обробки переносить її у чергу BUF done. Час обробки кожної заготовки залежить від її типу та етапу обробки. Якщо заготовка повністю пройшла обробку, вона потрапляє у місце зберігання BUF finished. Якщо пройдені не усі етапи обробки, вона повертається по конвеєру до черги BUF raw.

На діаграмі також вказано частоту виникнення та час усунення несправності верстатів.

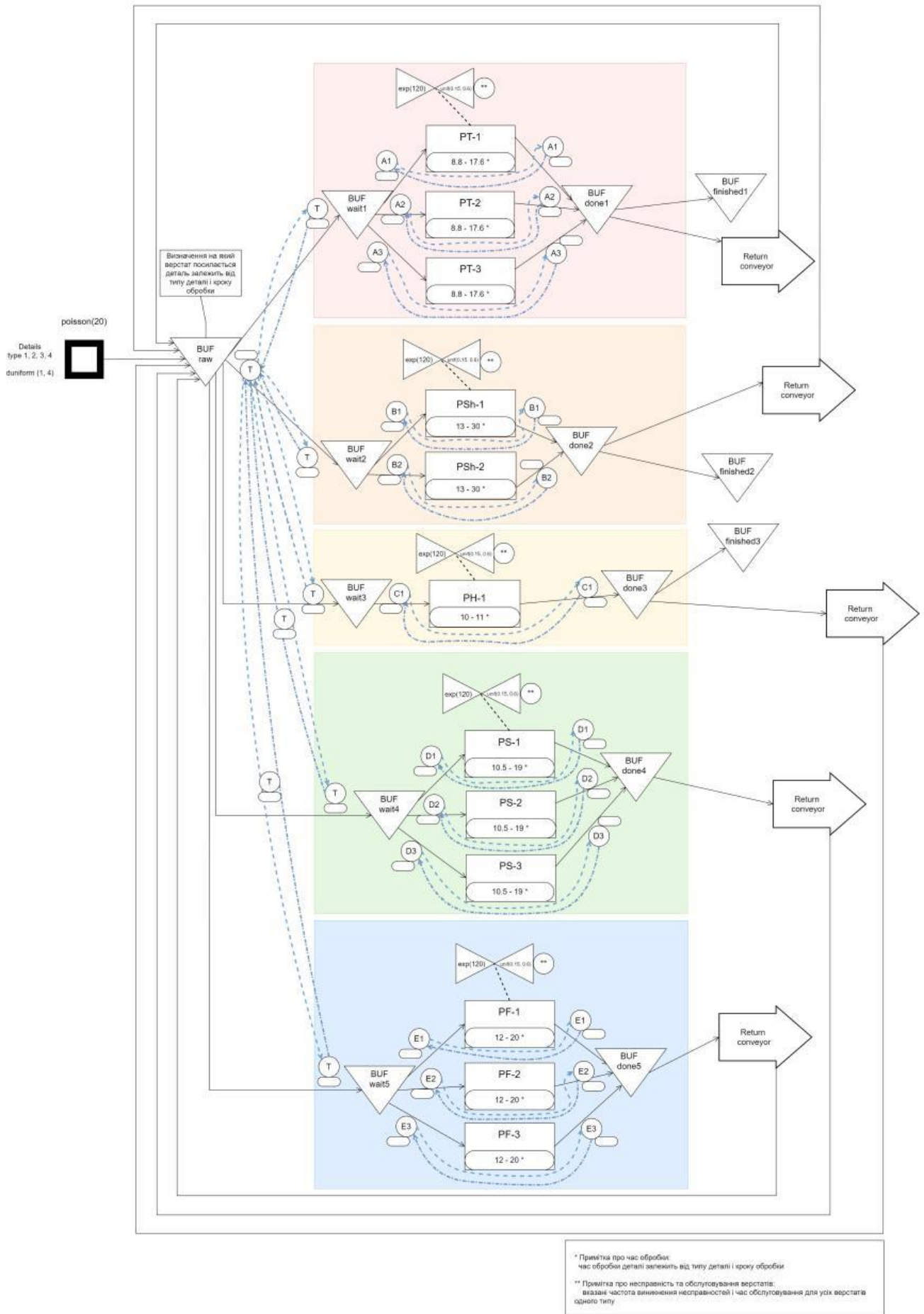


Рисунок 2.6 – OFD діаграма

2.3. Проектування моделі бази даних

Для збереження даних, що використовуються моделлю, створимо базу даних, що міститиме такі таблиці:

Way_map – маршрутна карта руху деталей по ділянці;

Time_income – загальний час виготовлення деталей та прибуток;

Process_time – час обробки деталей;

Types – типи деталей;

Побудуємо концептуальну ER-діаграму (Entity-Relationship diagram), щоб відобразити основні сутності та зв'язки між ними (рис. 2.7).

Більш детальний опис даних представлений у вигляді логічної моделі (рис. 2.8) і включає визначення первинних та зовнішніх ключів.

Фізична модель (рис. 2.9) надає уявлення про реальний проект реляційної БД, про структуру і зв'язки даних у конкретній СУБД.

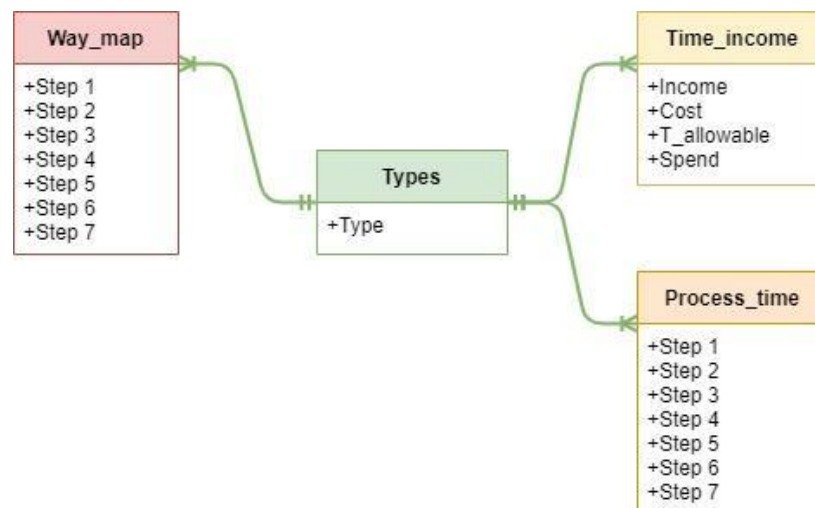


Рисунок 2.7 – Концептуальна модель бази даних

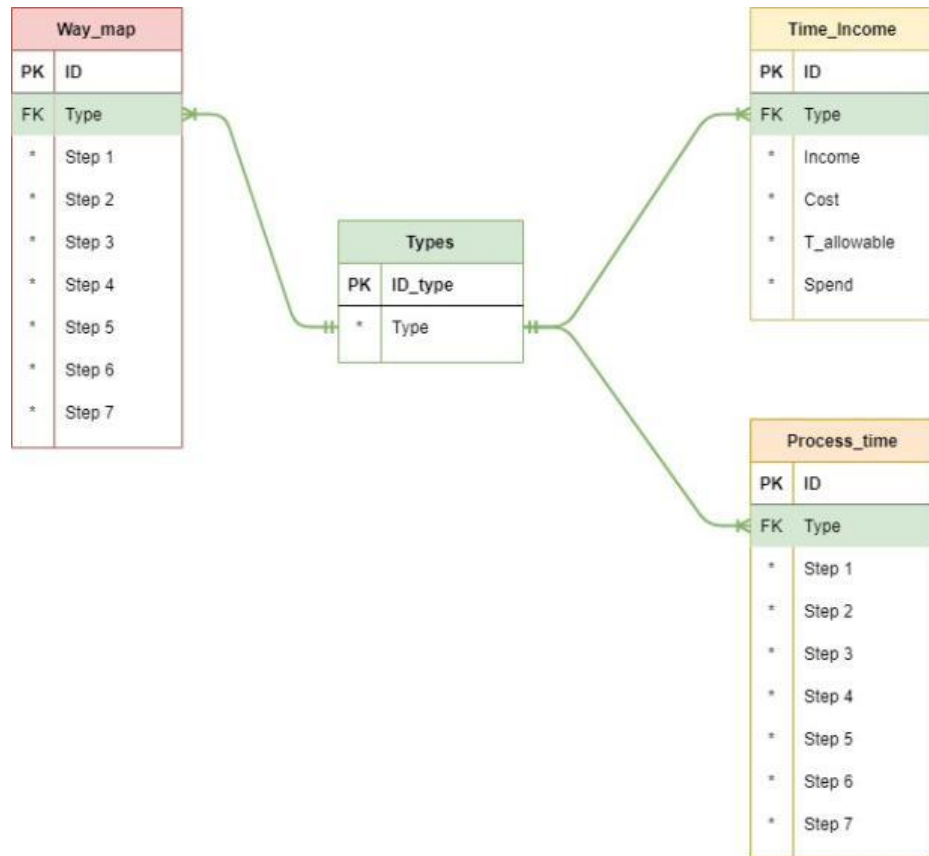


Рисунок 2.8 – Логічна модель бази даних

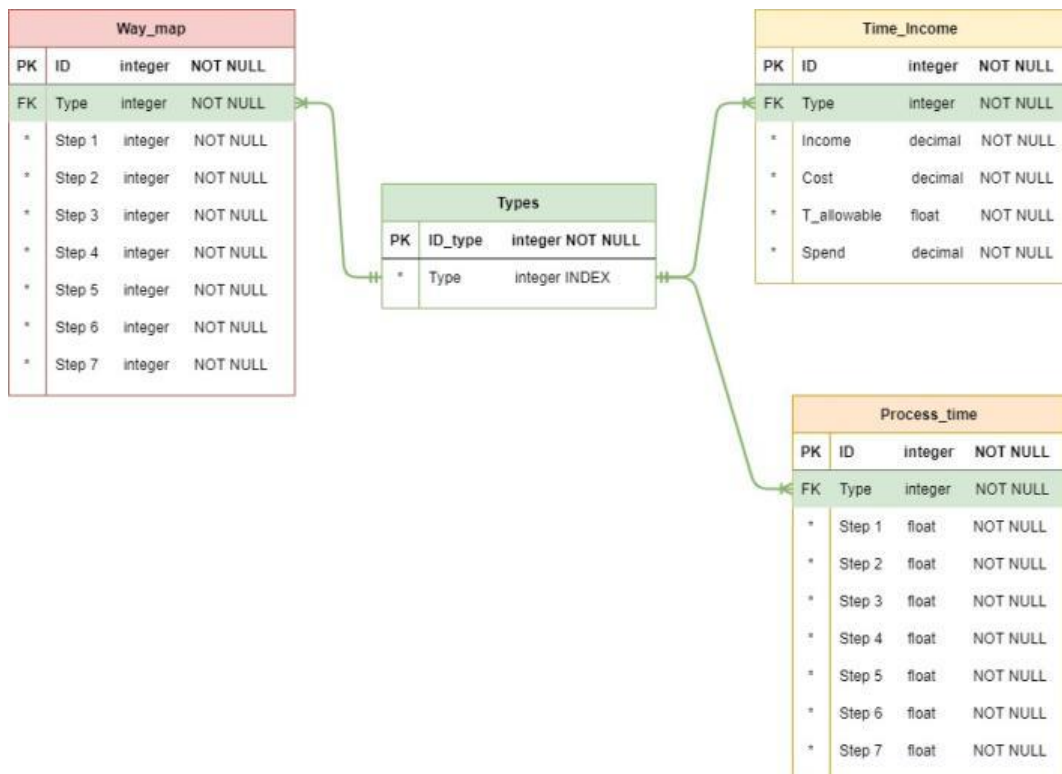


Рисунок 2.9 – Фізична модель бази даних

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІМІТАЦІЙНОЇ МОДЕЛІ

3.1. Структуризація даних

Вхідні дані до моделі зберігатимуться у вигляді таблиць у базі даних. Для створення БД використовується веб-додаток PhpMyAdmin та СУБД MySQL.

Створимо БД під назвою manufacture і додамо таблиці, зв'язавши їх відповідно до побудованої фізичної моделі бази даних.

Таблиця 3.1 – Опис полів таблиць бази даних manufacture

Поле таблиці	Опис
type_of_blank	Тип заготовки
income	Прибуток від реалізації деталі
cost	Собівартість деталі
max_time	Максимально допустимий час перебування деталі у системі
spend	Штраф при перевищенні максимально допустимого часу max_time
step 1-7	Відповідний крок обробки деталі

```
CREATE TABLE type (
    id_type integer NOT NULL AUTO_INCREMENT,
    type_of_blank integer NOT NULL,

    INDEX (type_of_blank),
    PRIMARY KEY (id_type)
);
```

Server: 127.0.0.1:3306 » Database: manufacture » Table: type

Structure SQL Search Insert Export Import Privileges Operations Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_type	int(11)			No	None		AUTO_INCREMENT	[Edit] [Delete] [Dropdown]
2	type_of_blank	int(11)			No	None			[Edit] [Delete] [Dropdown]

Рисунок 3.1 – Структура таблиці Type у базі даних manufacture

Server: 127.0.0.1:3306 » Database: manufacture » Table: type

Structure SQL Search Insert Export

Showing rows 0 - 3 (4 total, Query took 0.0001 seconds.)

SELECT * FROM `type`

Show all | Number of rows: 50 | Filter rows: Search this table

+ Options

		id_type	type_of_blank
[Edit]	[Delete]	1	1
[Edit]	[Delete]	2	2
[Edit]	[Delete]	3	3
[Edit]	[Delete]	4	4

Рисунок 3.2 – Заповнена таблиця Type

```

CREATE TABLE time_income (
    id integer NOT NULL AUTO_INCREMENT,
    type_of_blank integer NOT NULL,
    income decimal NOT NULL,
    cost decimal NOT NULL,
    max_time float NOT NULL,
    spend decimal NOT NULL,

    PRIMARY KEY (id),
    FOREIGN KEY (type_of_blank) REFERENCES type(type_of_blank)
);

```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	
2	type_of_blank	int(11)			No	None			
3	income	decimal(10,0)			No	None			
4	cost	decimal(10,0)			No	None			
5	max_time	float			No	None			
6	spend	decimal(10,0)			No	None			

Рисунок 3.3 – Структура таблиці Time_income у БД manufacture

Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

```
SELECT * FROM `time_income`
```

Show all | Number of rows: 50 | Filter rows: Search this table

	id	type_of_blank	income	cost	max_time	spend
1	1	1	1550	350	190	80
2	2	2	1850	420	160	120
3	3	3	1350	280	230	160
4	4	4	1450	315	140	100

Рисунок 3.4 – Заповнена таблиця Time_income

```
CREATE TABLE way_map (
  id integer NOT NULL AUTO_INCREMENT,
  type_of_blank integer NOT NULL,
  step1 integer NOT NULL,
  step2 integer NOT NULL,
  step3 integer NOT NULL,
  step4 integer NOT NULL,
  step5 integer NOT NULL,
  step6 integer NOT NULL,
  step7 integer NOT NULL,
  PRIMARY KEY (id),
```

FOREIGN KEY (type_of_blank) REFERENCES type(type_of_blank)
);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	
2	type_of_blank	int(11)			No	None			
3	step1	int(11)			No	None			
4	step2	int(11)			No	None			
5	step3	int(11)			No	None			
6	step4	int(11)			No	None			
7	step5	int(11)			No	None			
8	step6	int(11)			No	None			
9	step7	int(11)			No	None			

Рисунок 3.5 – Структура таблиці Way_map у БД manufacture

id	type_of_blank	step1	step2	step3	step4	step5	step6	step7
1	1	1	5	4	2	4	3	1
2	2	5	2	5	4	3	0	0
3	3	1	4	5	1	2	0	0
4	4	4	1	5	2	0	0	0

Рисунок 3.6 – Заповнена таблиця Way_map

```
CREATE TABLE process_time (
    id integer NOT NULL AUTO_INCREMENT,
    type_of_blank integer NOT NULL,
    step1 float NOT NULL,
    step2 float NOT NULL,
    step3 float NOT NULL,
    step4 float NOT NULL,
    step5 float NOT NULL,
```

step6 float NOT NULL,

step7 float NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (type_of_blank) REFERENCES type(type_of_blank)

);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	
2	type_of_blank	int(11)			No	None			
3	step1	float			No	None			
4	step2	float			No	None			
5	step3	float			No	None			
6	step4	float			No	None			
7	step5	float			No	None			
8	step6	float			No	None			
9	step7	float			No	None			

Рисунок 3.7 – Структура таблиці Process_time БД manufacture

Showing rows 0 - 3 (4 total, Query took 0.0001 seconds.)

```
SELECT * FROM `process_time`
```

Show all | Number of rows: 50 | Filter rows: Search this table | Sort by key: None

	id	type_of_blank	step1	step2	step3	step4	step5	step6	step7
1	1	1	8.8	12	12	13	10.5	11	11.5
2	2	2	20	14	14.5	16	10	0	0
3	3	3	17.6	19	14	11.6	30	0	0
4	4	4	19	16.8	13	19	0	0	0

Рисунок 3.8 – Заповнена таблиця Process_time

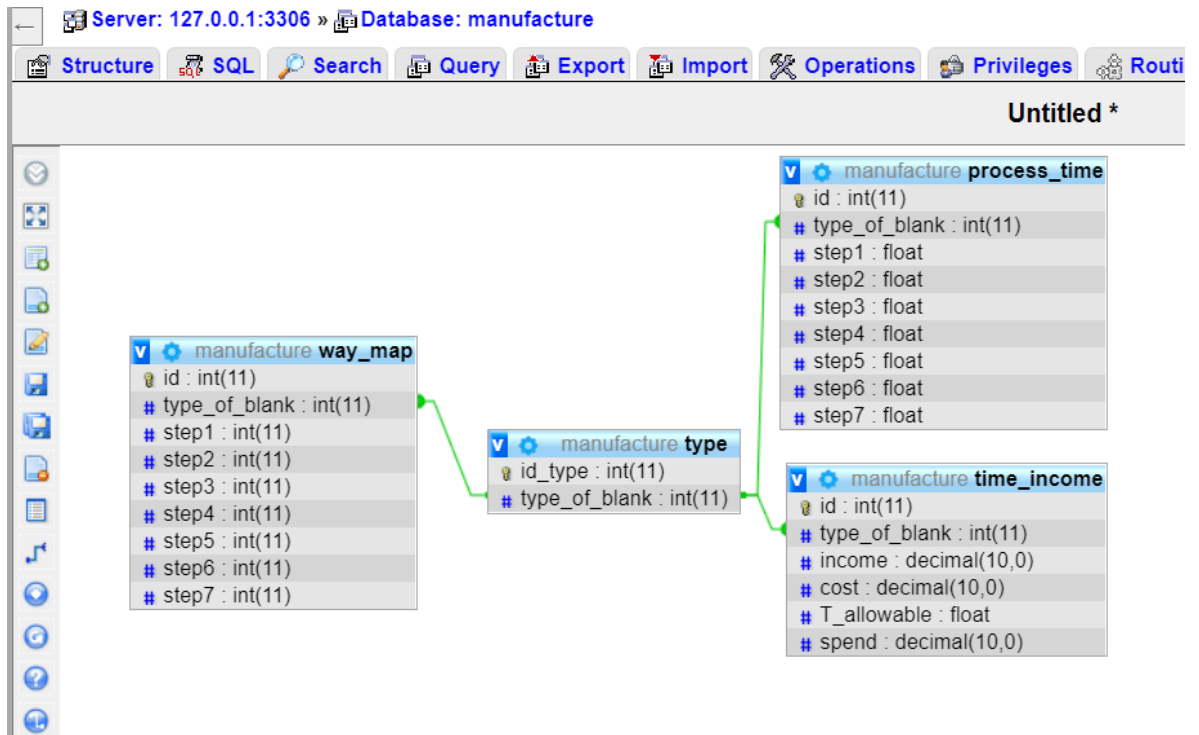


Рисунок 3.9 – Структура БД manufacture

Створену базу даних підключимо до моделі за допомогою інструмента Database Connector (рис. 3.10).

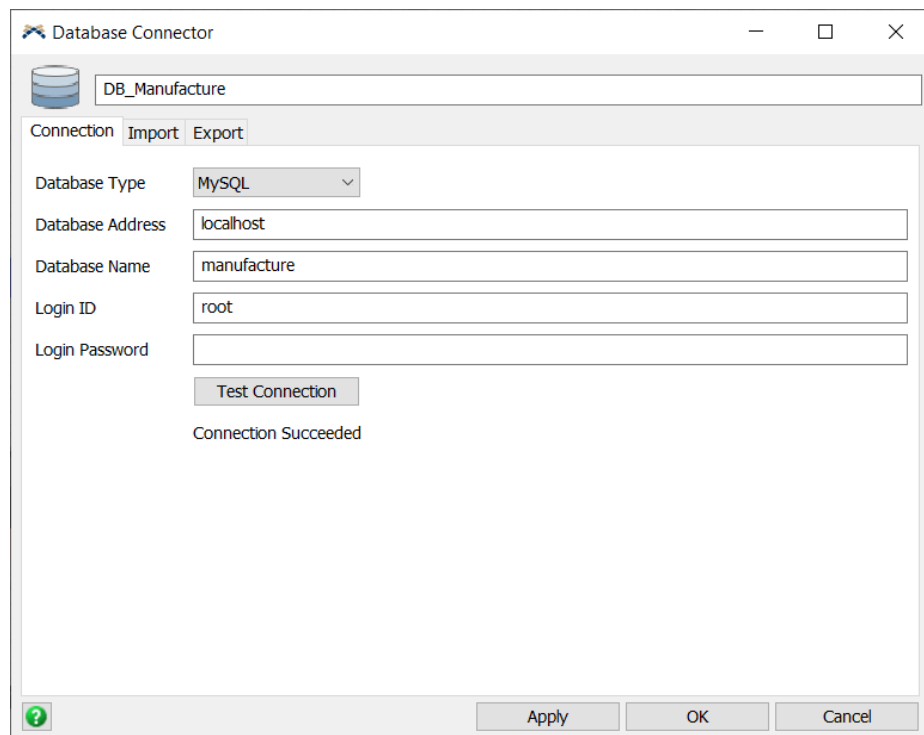


Рисунок 3.10 – Підключення до БД за допомогою Database Connector

Для імпорту та використання вхідних даних моделлю потрібно створити глобальні таблиці та заповнити їх за допомогою SQL запитів (рис. 3.11-3.13).

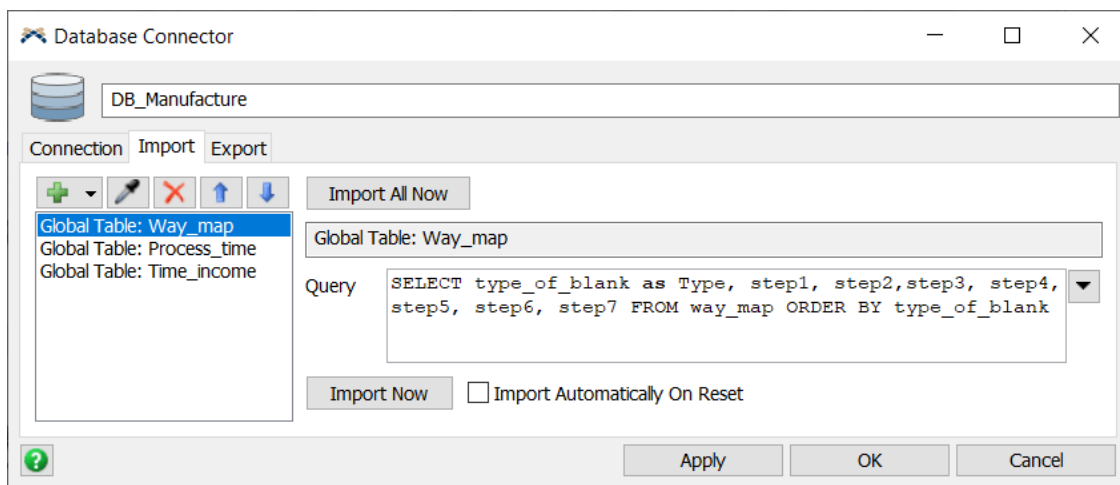


Рисунок 3.11 – Імпорт даних з БД у таблицю Way_map

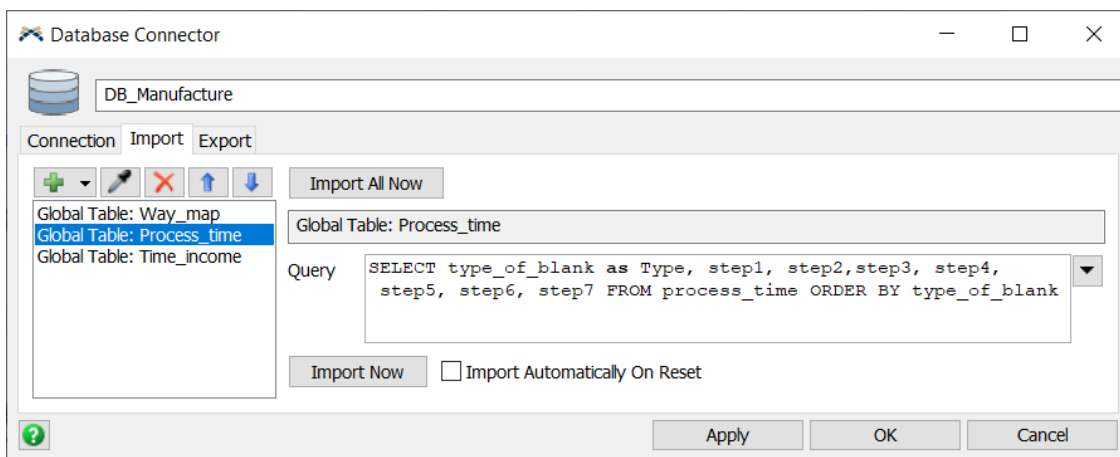


Рисунок 3.12 – Імпорт даних з БД у таблицю Process_time

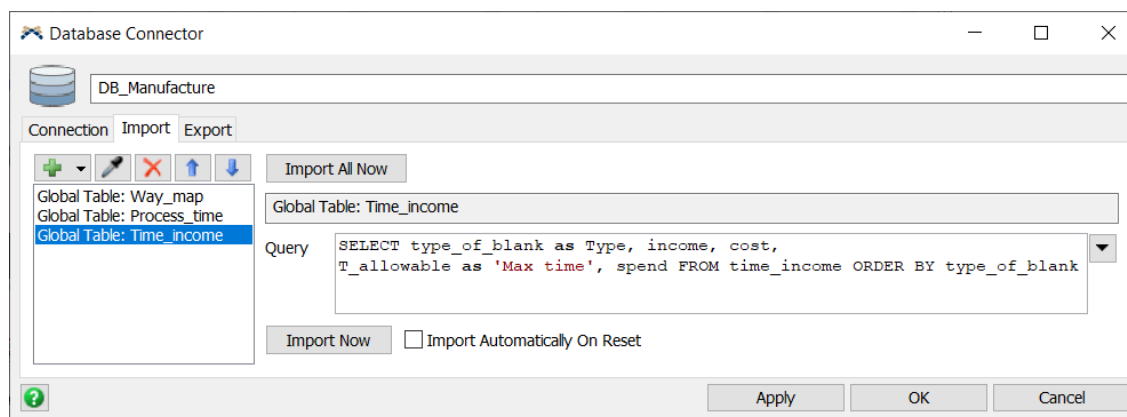


Рисунок 3.13 – Імпорт даних з БД у таблицю Time_income

3.2. Програмна реалізація імітаційної моделі

Додамо потрібні елементи системи у робочу область FlexSim та дамо їм відповідні назви.

Source – джерело надходження заготовок деталей у систему.

BUF raw – це черга, що зберігає заготовки деталей, що надійшли на виробничу ділянку.

BUF wait1, BUF wait2, BUF wait3, BUF wait4, BUF wait5 – черги, що зберігають деталі, які очікують на обробку у певному відділі ділянки.

PT_1, PT_2, PT_3, PSh_1, PSh_2, PH_1, PS_1, PS_2, PS_3, PF_1, PF_2, PF_3 – процесори, що обробляють деталі; токарні, шліфувальні, хонінгувальні, свердлильні, фрезерні верстати відповідно.

BUF done1, BUF done2, BUF done3, BUF done4, BUF done5 – черги, що зберігають деталі, що пройшли обробку на певному кроці, які потім відправляються на наступний крок або виходять із системи.

BUF finished1, BUF finished2, BUF finished3 – черги, що зберігають повністю оброблені та реалізовані деталі.

A1, A2, A3, B1, B2, C1, D1, D2, D3, E1, E2, E3, T – оператори та транспортні засоби, що транспортують деталі у системі.

Dispatcher1, Dispatcher2, Dispatcher3, Dispatcher4, Dispatcher5 – диспетчери, які використовуються для управління групою транспортерів або операторів.

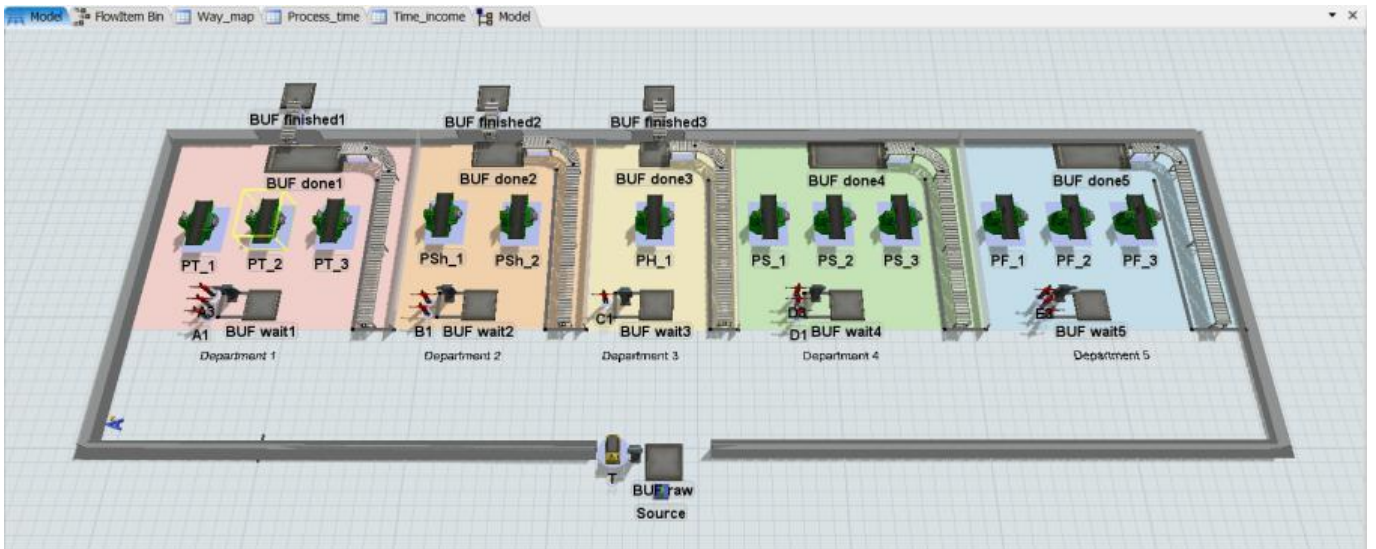


Рисунок 3.14 – Основні елементи моделі

Для того, щоб обмежити рух операторів і не давати проходити крізь об'єкти, використаємо навігатор A* (A Star) (рис. 3.15). A* – це алгоритм, який використовується для пошуку шляху між точками. Алгоритм визначає, який напрямок є найшвидшим, включаючи переміщення по діагоналі між вузлами. Шлях може бути обмежений бар'єрами.

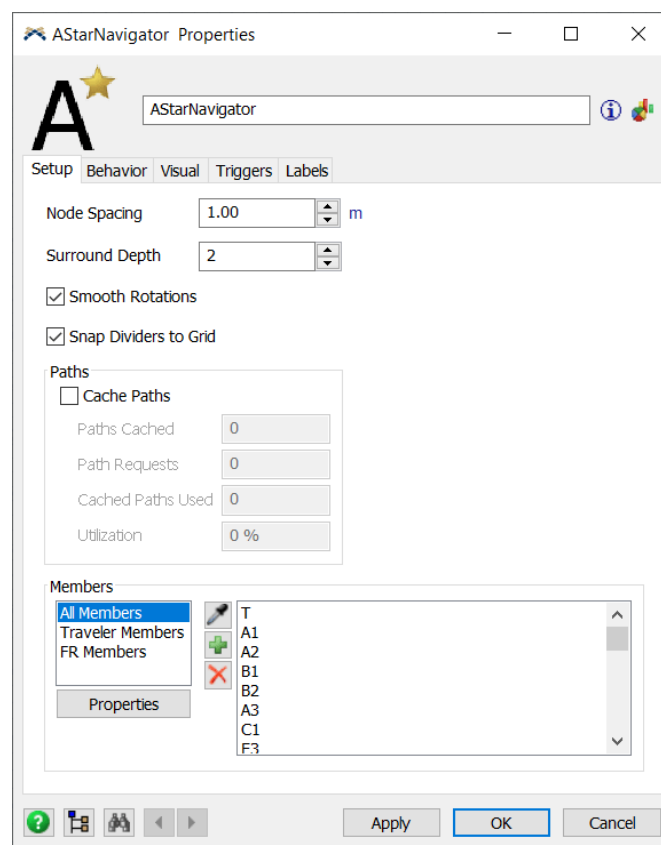


Рисунок 3.15 – Налаштування навігатора A*

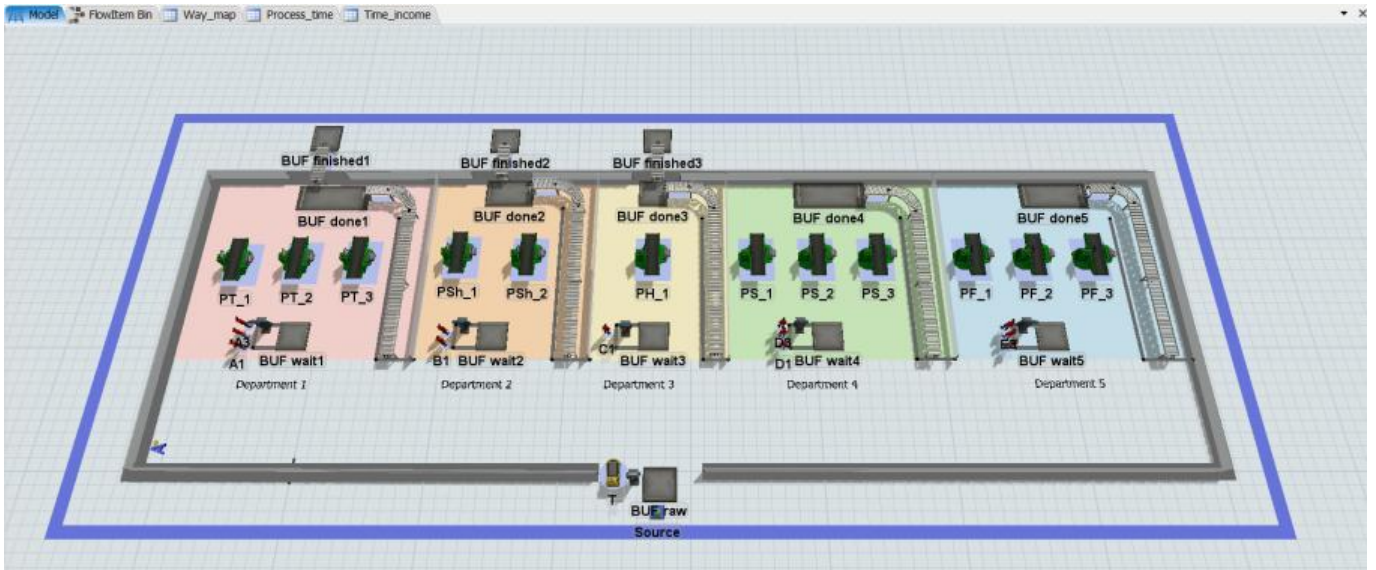


Рисунок 3.16 – Область дії навігатора А*

На наступному етапі потрібно з'єднати тривимірні об'єкти, щоб елементи потоку переходили від одного фіксованого ресурсу до іншого у відповідному порядку. Один з найпростіших методів з'єднання об'єктів – через порти.

У FlexSim порт – це точка на об'єкті, через яку елементи потоку можуть передаватися від одного об'єкта до іншого.

З'єднаємо порти елементів системи за допомогою інструменту Connect Object.

Для того, щоб залучити операторів до процесу підключимо їх до диспетчерів, а диспетчерів до центральних портів процесорів за допомогою Connect Center Ports.

Центральні порти зазвичай використовуються для підключення виконавців завдань (Task Executor) до фіксованих ресурсів. Вони дозволяють об'єктам обмінюватися даними або взаємодіяти складним чином: транспортувати елементів потоку, налаштовувати обробку та обробляти елементи потоку та ін.

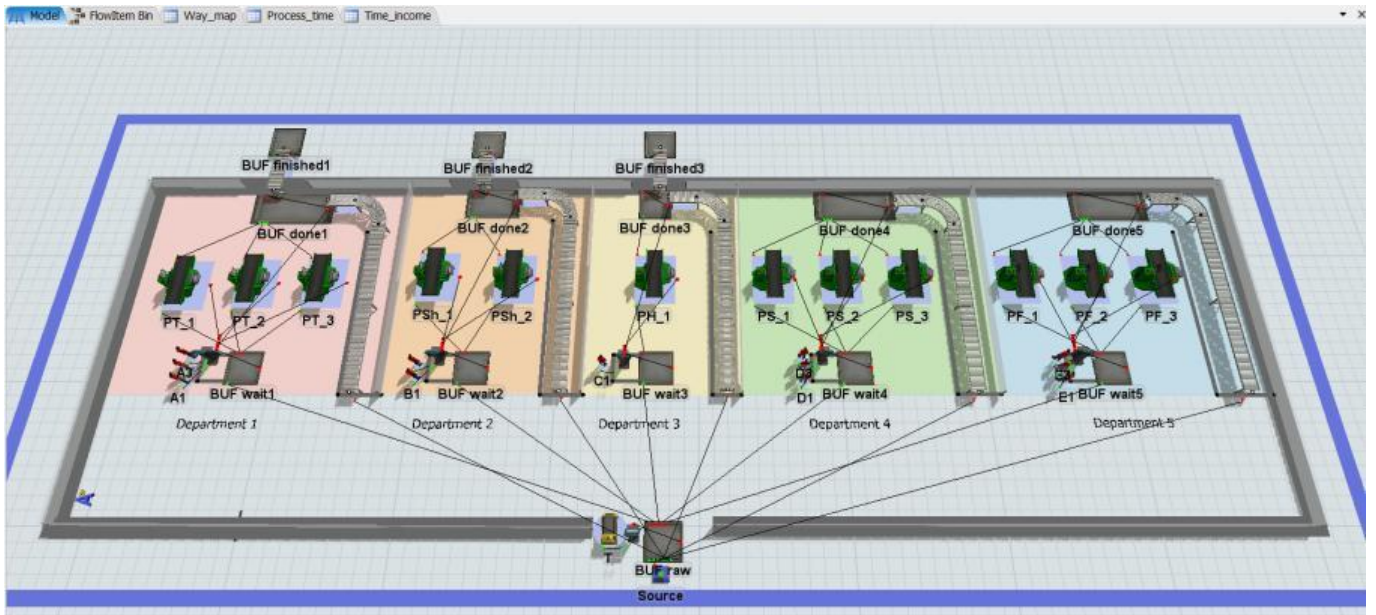


Рисунок 3.17 – З'єднання об'єктів моделі

Мітки – важливий компонент побудови логіки моделі. Мітка – це спосіб зберігання інформації про елемент потоку, токен або 3D-об'єкт. Вони можуть відстежувати важливу інформацію або динамічно змінювати процеси, що відбуваються під час симуляції, в залежності від різних умов.

Елементу потоку Vox, який у імітаційній моделі представляє деталі, присвоєно мітки, що зберігатимуть інформацію про крок обробки, час перебування у системі та ціну деталі з урахуванням собівартості та штрафів.

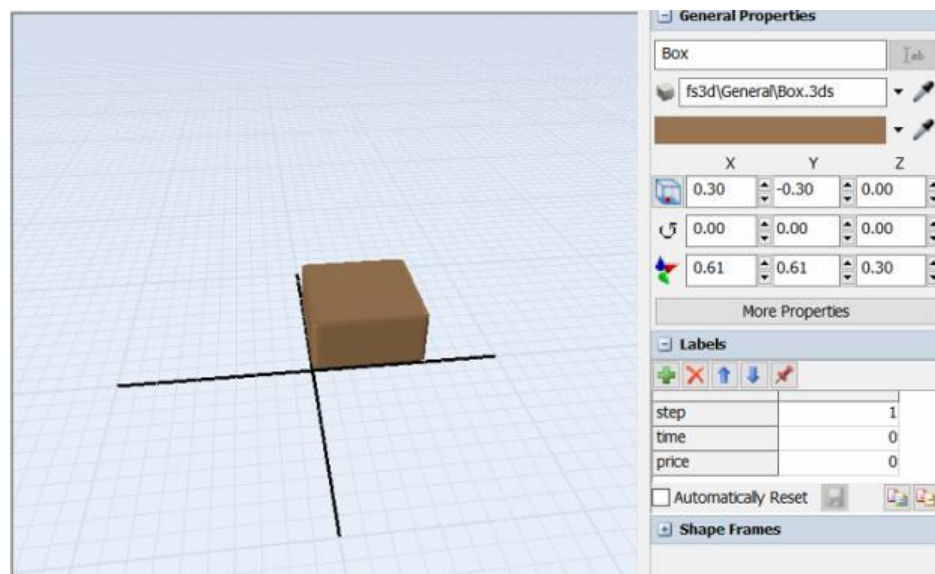


Рисунок 3.18 – Призначення міток елементам потоку

Часові інтервали надходження елементів у систему задано таблицею у вигляді вибірки даних (табл. А.2). Визначимо закон розподілу для даної вибірки.

Створимо новий проект, тип аналізу зазначимо Fit distribution to data, тобто визначення закону розподілу, що відповідає даним.

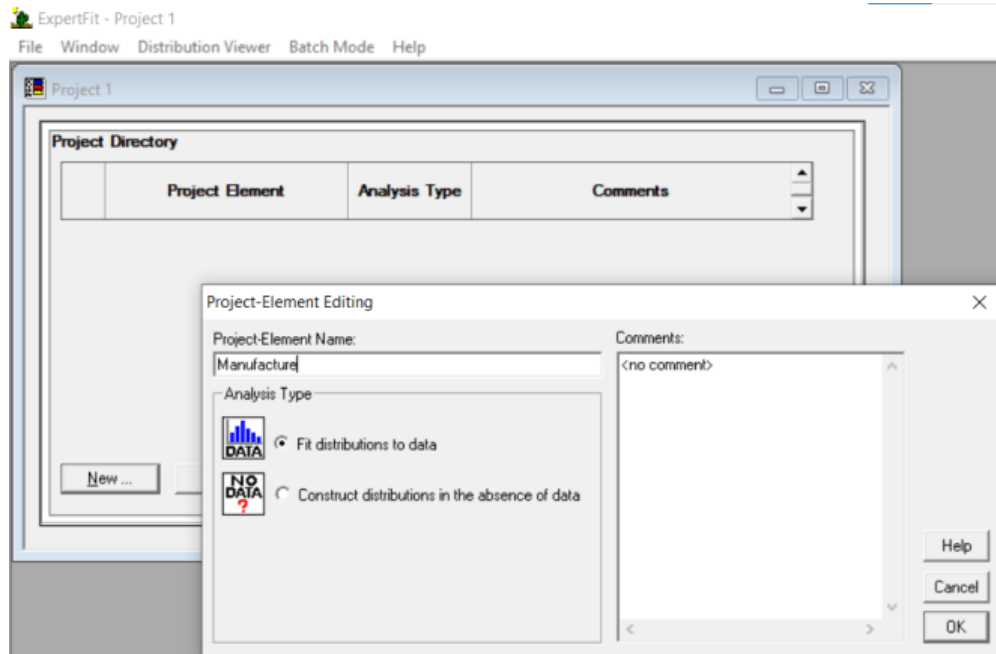


Рисунок 3.19 – Створення нового програмного проекту

На вкладці Data викликаємо вікно Data Editor та вводимо вхідні дані, перед цим скопійовані у буфер обміну, кнопкою Paste to End from Clipboard.

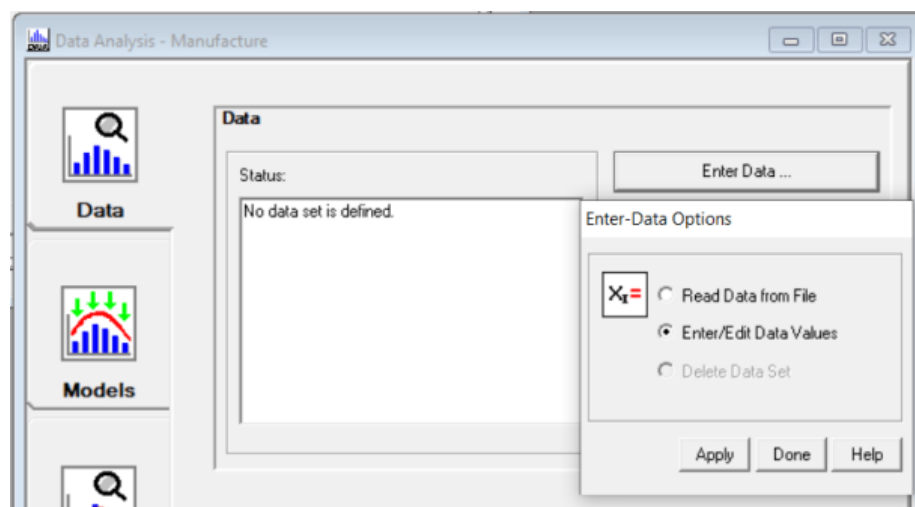


Рисунок 3.20 – Задання параметрів введення вхідних даних

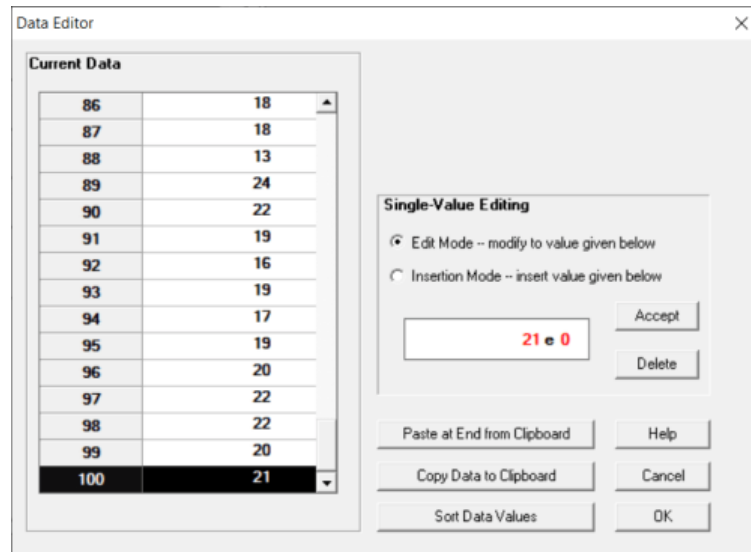


Рисунок 3.21 – Введення числової вибірки інтервалів надходження елементів

У результаті отримуємо таблицю з характеристикою введеної вибірки.

Data Characteristic	Value
Source file	<edited>
Observation type	Integer valued
Number of observations	100
Minimum observation	5
Maximum observation	29
Mean	18.11000
Median	18.00000
Variance	18.17970
Lexis ratio (var./mean)	1.00385
Skewness	-0.03844

Рисунок 3.22 – Характеристика введених даних

Для визначення закону розподілу скористаємося автоматичним підбором на вкладці Models – Automated Fitting.

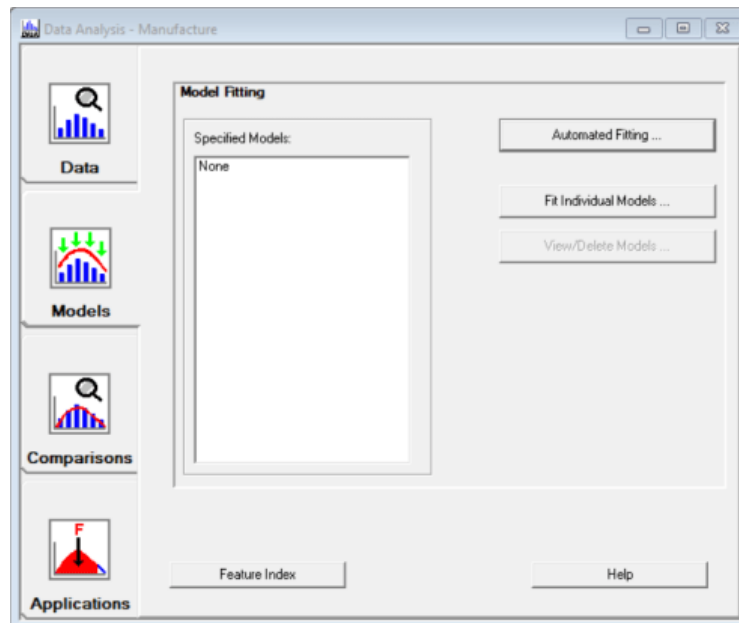


Рисунок 3.23 – Автоматичне визначення закону розподілу на вкладці Models

Automated-Fitting Results

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters
1 - Poisson	100.00	Lambda 18.11000
2 - Discrete Uniform	66.67	Lower endpoint 5 Upper endpoint 29
3 - Geometric	33.33	Probability 0.05233

4 models are defined with scores between 0.00 and 100.00

Absolute Evaluation of Model 1 - Poisson

Evaluation: Indeterminate
 Suggestion: Additional evaluations using Comparisons Tab are strongly recommended.
 See Help for more information.

Additional Information about Model 1 - Poisson

"Error" in the model mean relative to the sample mean 0

Copy
Print
Help
Done

Рисунок 3.24 – Результат автоматичного визначення закону розподілу

За отриманими результатами можна зробити висновок, що заданій вибірці найбільше відповідає закон Пуассона з лямбдою 18.

У параметрах об'єкта Source задаємо час надходження відповідно до визначеного закону розподілу.

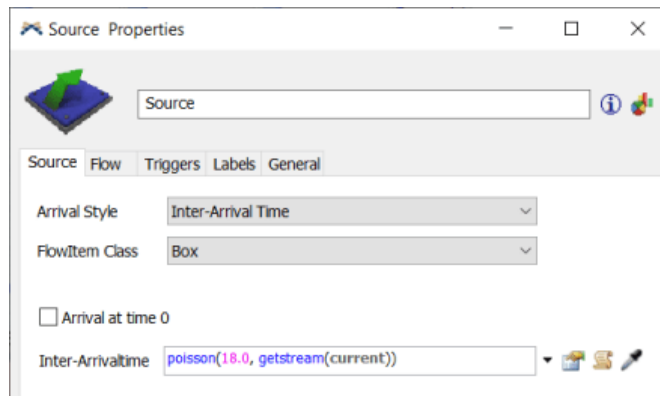


Рисунок 3.25 – Налаштування часу надходження елементів у систему

Відповідно до умови задачі, представленій у технічному завданні (додаток А), на виробничу ділянку надходять 4 типи деталей, що розподілені рівномірно. Реалізуємо дану логіку за допомогою тригера On Creation у налаштуваннях Source.

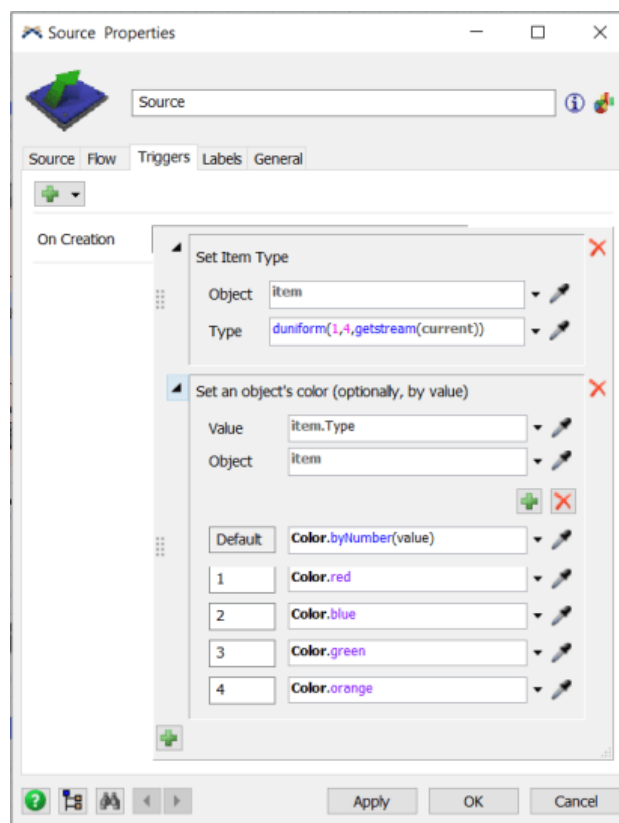


Рисунок 3.26 – Налаштування тригера On Creation об'єкта Source

Розподіл елементів, що надійшли у систему, по відділам відбувається у черзі BUF raw за допомогою таблиці Way_map. Елементи з черги перевозяться транспортом, тому потрібно відмітити пункт Use Transport.

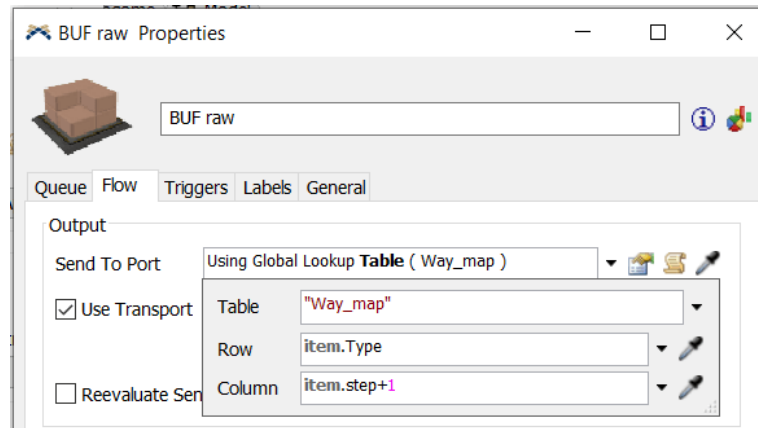


Рисунок 3.27 – Налаштування вихідного потоку черги BUF raw

Таким же чином налаштовуємо час обробки на кожному процесорі відповідно до таблиці Process_time.

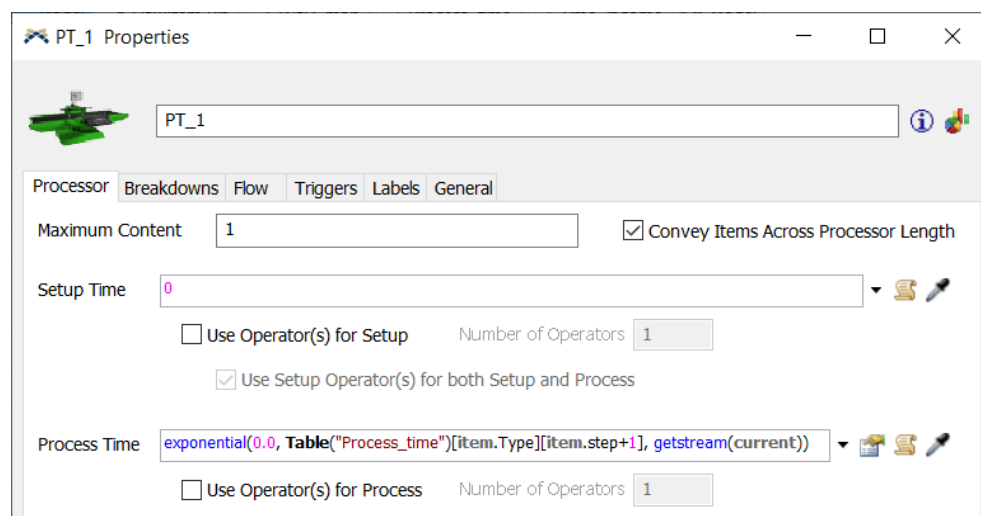


Рисунок 3.28 – Задання часу обробки на процесорах

Додатково реалізуємо подію виникнення поломок обробляючих об'єктів за допомогою MTBF/MTTR.

Об'єкти MTBF/MTTR використовуються для установки випадкового часу відмови і відновлення роботи для груп об'єктів в моделі. Вони дають можливість вказувати, в який стан переходитимуть об'єкти, коли вони вийдуть з ладу, та їх поведінку.

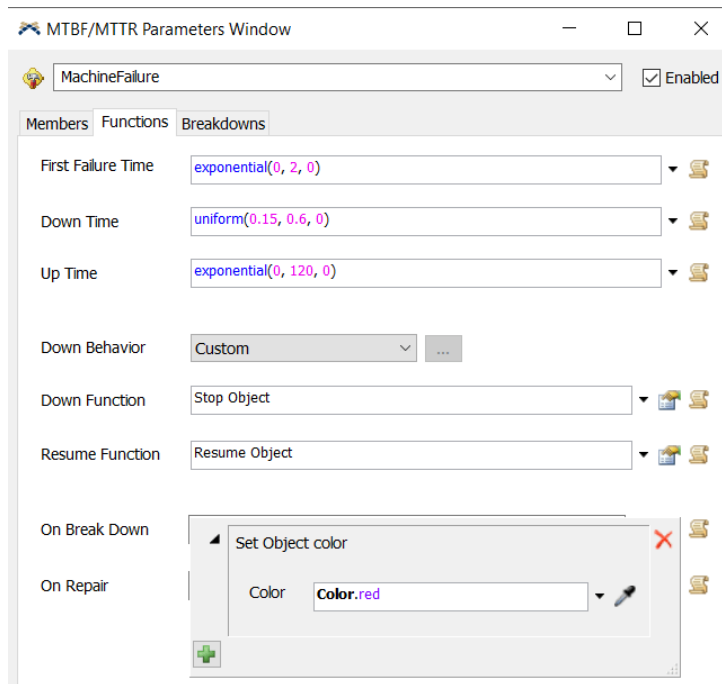


Рисунок 3.29 – Параметри MTBF/MTTR

Застосуємо створений MTBF/MTTR до усіх процесорів у системі.

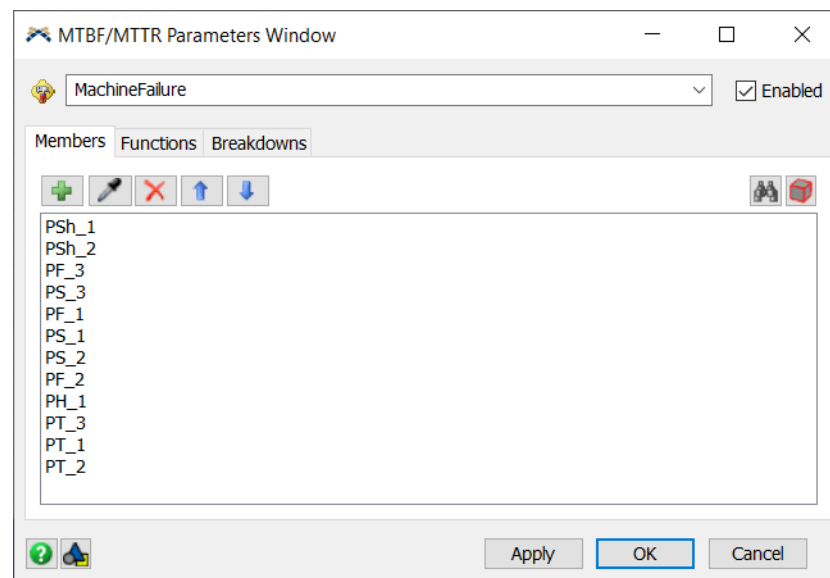


Рисунок 3.30 – Додавання процесорів у список впливу MTBF/MTTR

На вкладці Breakdowns параметрів MTBF/MTTR налаштуємо виникнення поломок процесора тільки коли він у стані обробки.

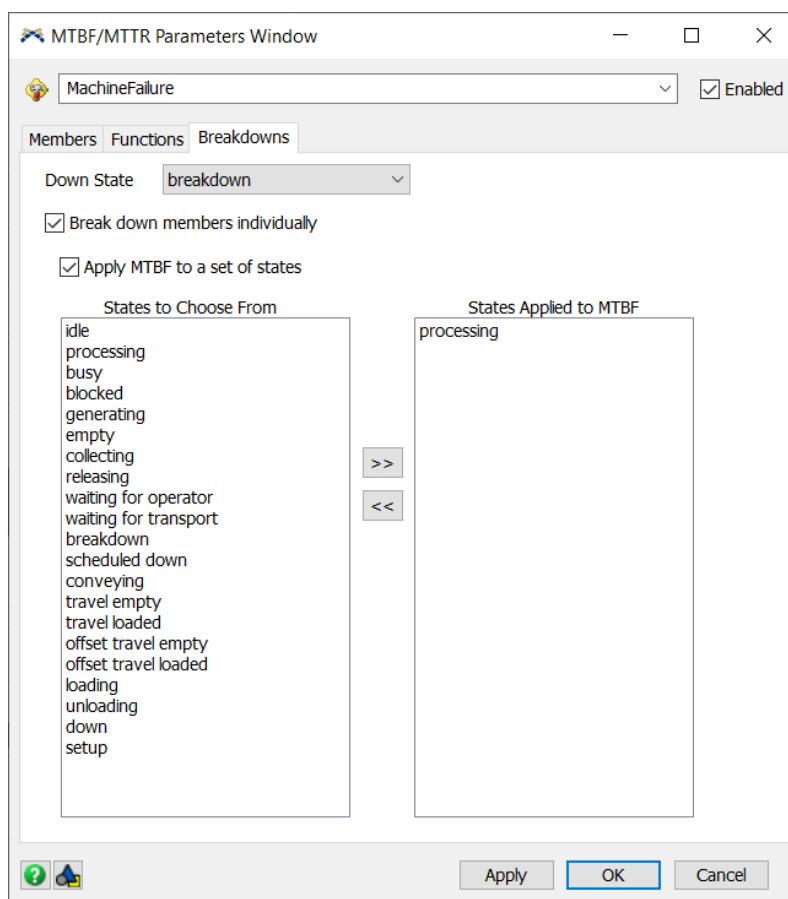


Рисунок 3.31 – Налаштування станів виникнення поломок

Визначення шляху елемента потоку, який зберігається у черзі BUF done після обробки на верстаті, задамо кодом.

Таблиця 3.2 – Таблиця змінних коду вихідного потоку об'єкту BUF done

Ім'я змінної	Опис
item	Поточний елемент
item.step	Крок обробки поточного елемента
item.Type	Тип поточного елемента
value	Номер порту, по якому буде відправлено поточний елемент на подальшу обробку

Спочатку перевіримо, чи не є крок обробки поточного елемента останнім відносно максимальної кількості кроків, заданих у таблиці Way_map. Якщо ця умова

виконується, елемент відправляється у чергу BUF finished по конвеєру, що підключений до другого порту черги BUF done.

Для елементів, які мають менше кроків обробки, усі наступні кроки після останнього позначені 0. Тому також потрібно перевірити рівність $value == 0$. При виконанні рівності елемент відправляється у чергу BUF finished, інакше повертається у BUF raw по конвеєру, що підключений до першого порту BUF done.

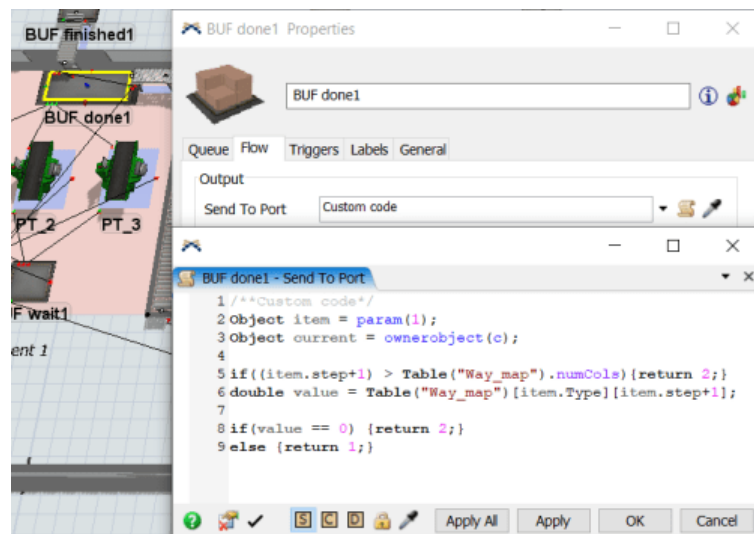


Рисунок 3.32 – Налаштування логіки вихідного потоку черги BUF done

Після обробки на процесорі елемент переходить на наступний крок. За допомогою тригера On Entry та функції Increment збільшимо значення кроку на 1.

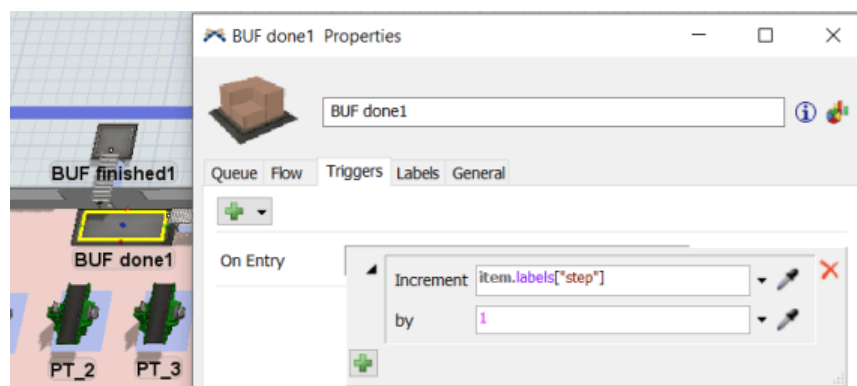


Рисунок 3.33 – Збільшення значення кроку поточного елемента у черзі BUF done

Потрапляючи на конвеєр до черги BUF finished, елемент виходить із системи. На цьому етапі визначимо час перебування елемента у системі, визначимо чи перевищив він допустимий час, та розрахуємо прибуток від реалізації.

Таблиця 3.3 – Таблиця змінних коду розрахунку ціни деталі

Ім'я змінної	Опис
item.time	Час перебування поточного елемента у системі
item.price	Мітка поточного елемента, що зберігає прибуток від реалізації деталі з урахуванням штрафів
Amount_of_delayed	Глобальна змінна, що зберігає кількість елементів, що перевищили допустимий час перебування у системі

Прибуток item.price розраховується як різниця ціни реалізації та собівартості деталі, які також зазначені у таблиці Time_income.

Час item.time визначимо різницею time(), поточного системного часу, та getcreationtime(), час, коли елемент з'явився у системі. Отримане значення порівнюємо із значенням max_time таблиці Time_income, і якщо воно більше, встановлюється чорний колір елемента, від item.price віднімається значення штрафу та кількість затриманих елементів Amount_of_delayed збільшується на 1.

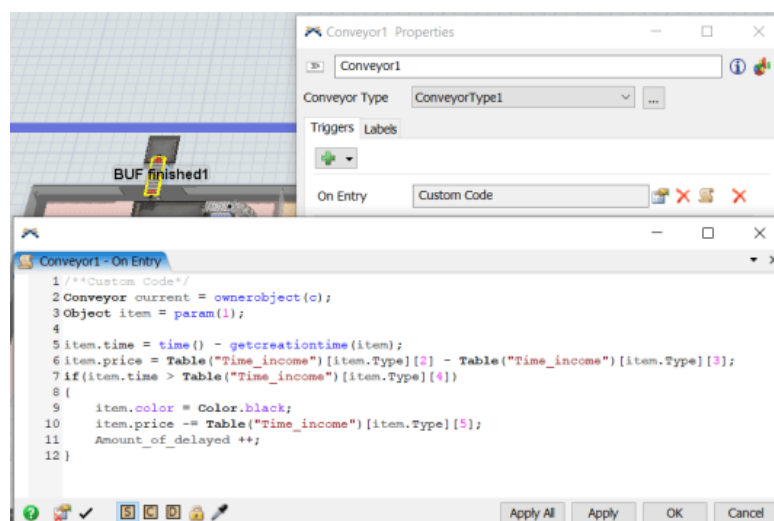


Рисунок 3.34 – Розрахунок прибутку та кількості затриманих елементів

Для перевірки гіпотези С попередньо налаштуємо стратегію pull для потоку елементів на усіх процесорах.

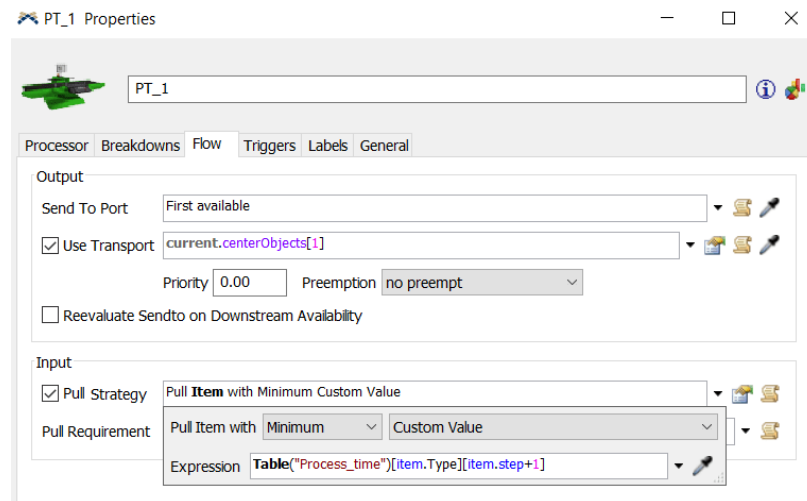


Рисунок 3.35 – Налаштування стратегії pull

Інтерфейс користувача передбачає зміну часу надходження. FlexSim дає можливість досліджувати та копіювати елементи існуючих інтерфейсів. Тому використаємо елемент Inter-Arrivaltime параметрів об'єкту Source у створюваному інтерфейсі.

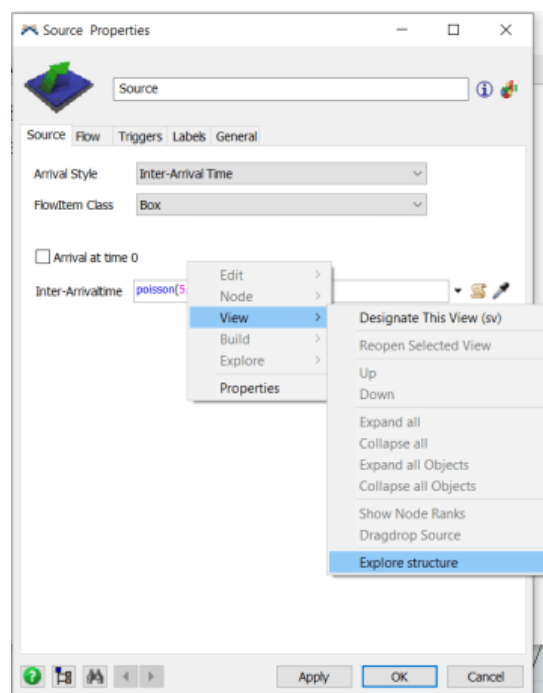


Рисунок 3.36 – Копіювання елемента Intre-Arrivaltime із структури інтерфейсу налаштувань Source

Скопійований елемент інтерфейсу вставляємо у структуру нового інтерфейсу користувача.

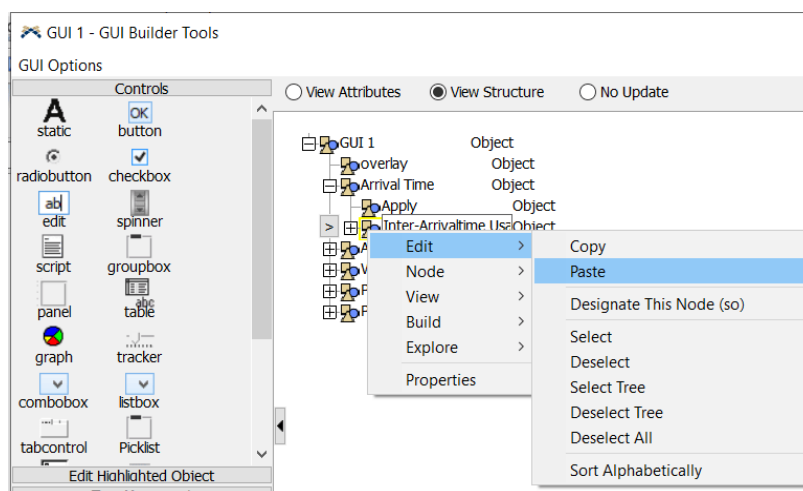


Рисунок 3.37 – Вставка елемента Inter-Arrivaltime в створюваний інтерфейс користувача

Для того, щоб зміна часу через інтерфейс впливала на модель, потрібно задати правильні шляхи в атрибуту objectfocus у дереві GUI.

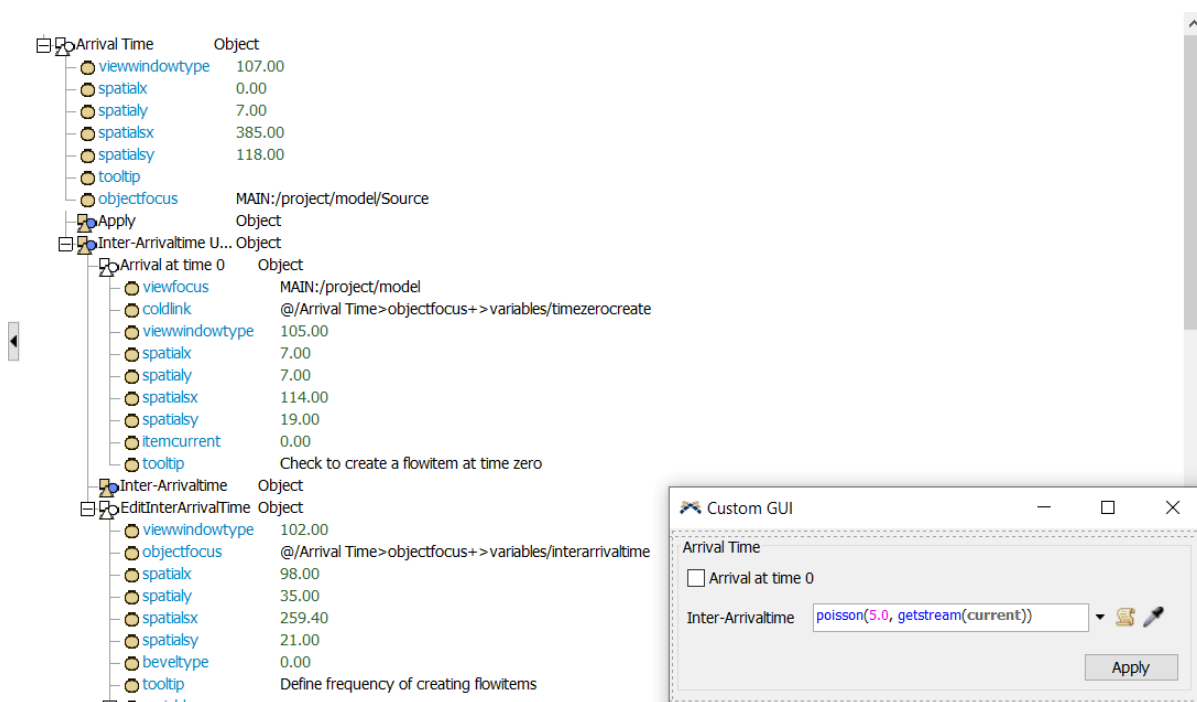


Рисунок 3.38 – Налаштування зв'язку між елементом інтерфейсу Inter-Arrivaltime та моделлю

Для зручної зміни кількості процесорів, залучених у процес, додамо combobox-и для 1, 2, 4 та 5-го відділів виробничої системи. У ході аналізу вхідних даних та структури ВС було зроблено висновок, що для третього відділу достатньо одного верстата, тому у моделі виділено один процесор PH_1.

У випадку, якщо GUI викликається через кнопку Model Control GUI, створюється його копія. Через це об'єкти моделі не матимуть доступу до значень атрибутів елементів інтерфейсу, таких як поточне значення combobox – itemcurrent. Між доданими combobox та моделлю також потрібно налагоджувати зв'язок.

Для події OnSelect пропишемо код, у якому обране значення в інтерфейсі користувача присвоюється тому ж елементу у дереві моделі.

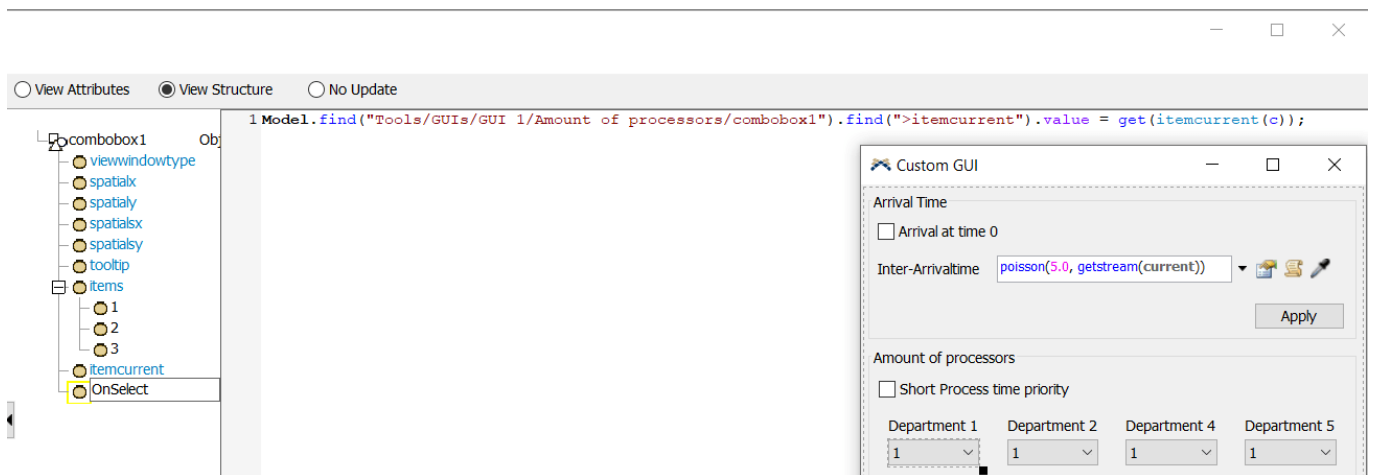


Рисунок 3.39 – Налагодження зв'язку між елементами combobox та моделлю

Далі використаємо тригер OnReset та відслідкуємо значення кількості працюючих процесорів val з інтерфейсу користувача. Залежно від порядкового номера процесора перевіряємо умову: якщо $val > 1$ (порядковий номер 2) або $val > 2$ (порядковий номер 3), то відкрити вхідний порт, інакше закрити.

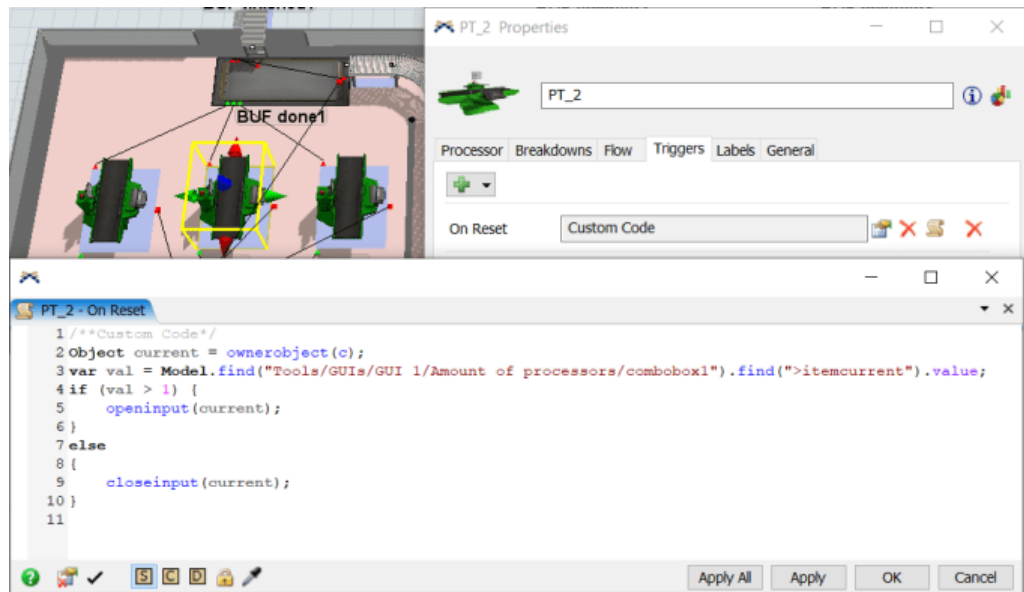


Рисунок 3.40 – Налаштування підключення та виключення процесорів із системи

Попередньо налаштувавши на усіх процесорах стратегію pull, за якою першими будуть оброблятися елементи з найменшим часом обробки, додамо до інтерфейсу елемент checkbox, який вмикатиме і вимикатиме дану функцію.

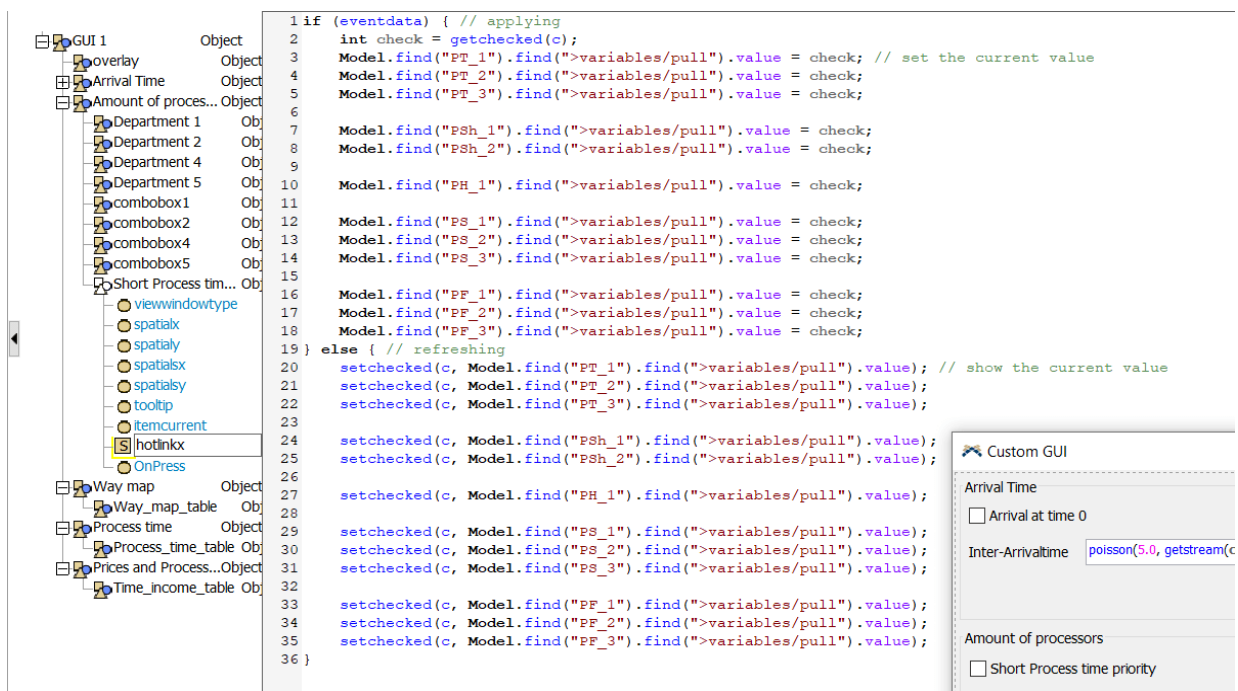


Рисунок 3.41 – Налаштування елементу управління підключенням стратегії pull

Для події OnPress використаємо функцію виконання коду, прописаного в атрибутах link.

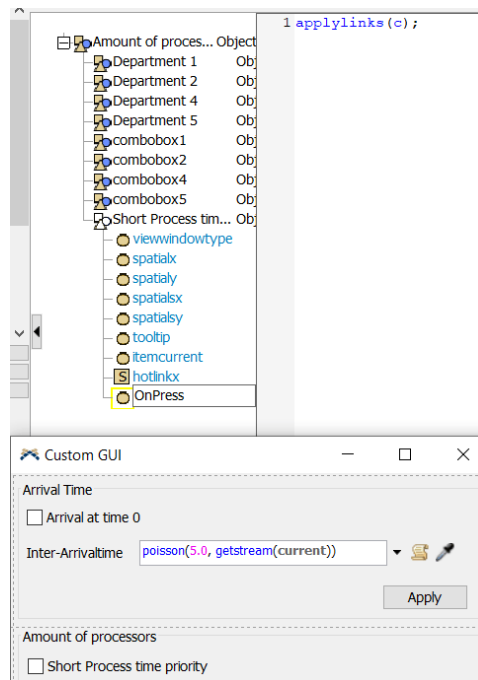


Рисунок 3.42 – Виконання коду коли відбувається подія OnPress

Зміну вхідних даних, що зазначені у таблицях, організуємо за допомогою елемента table view, який відображає вміст таблиці та дає змогу її редагувати.

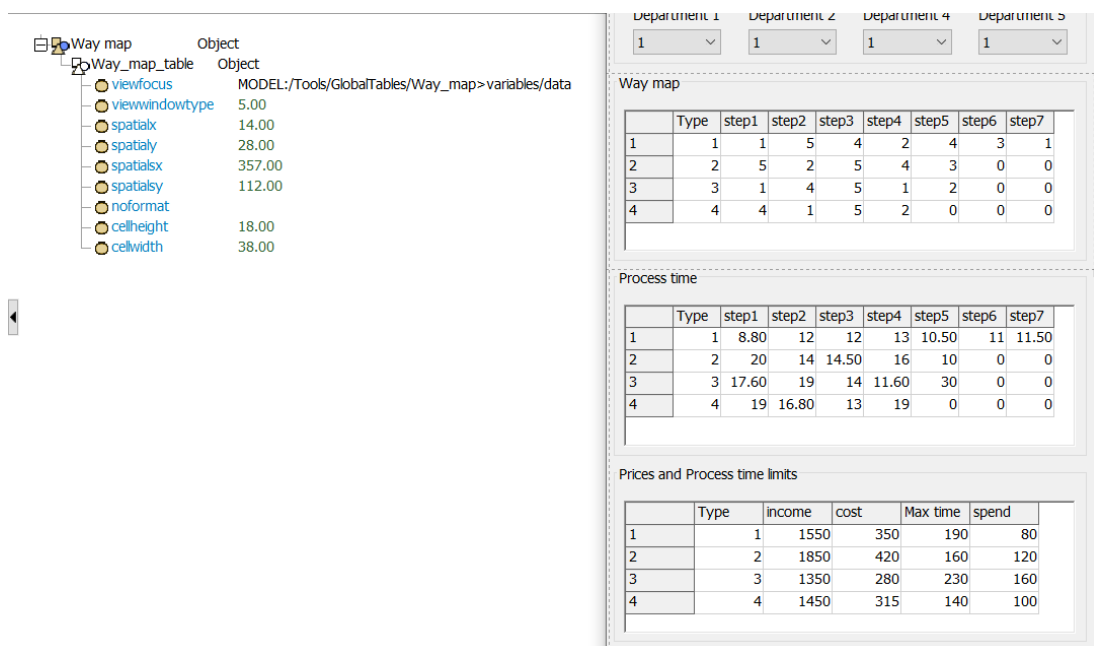


Рисунок 3.43 – Додавання таблиць в інтерфейс користувача

Проведемо запуск моделі з базовими вхідними даними та по одному працюючому процесору у кожному відділі. Час симуляції визначимо як 1 робочий день з 8:00 до 17:00.

У результаті імітації можна зробити висновок, що у системі є вузькі місця у відділах 2, 4 і 5. Також на панелі статистики Processors (рис. 3.44) видно, що процесори цих відділів, мають високу завантаженість. На панелі Efficiency (рис. 3.45) показано загальний прибуток, кількість затриманих деталей у системі, які в наслідку були оштрафовані, та кількість деталей, які надійшли на ВС та були повністю оброблені та реалізовані. Всього оброблених деталей 16 із 30, серед яких 8 оштрафовані.

По отриманим статистичним даним видно, що система має низьку пропускну здатність.

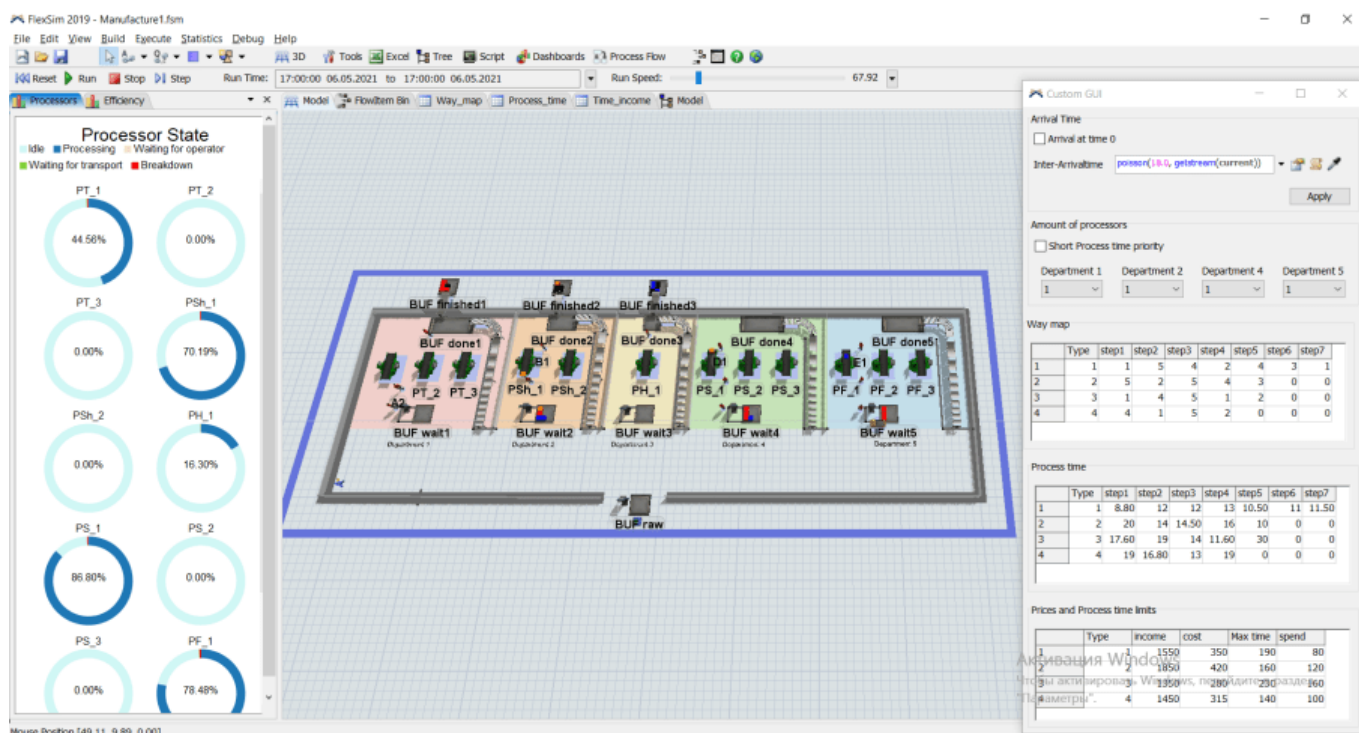


Рисунок 3.44 – Панель статистики Processors та результат запуску моделі

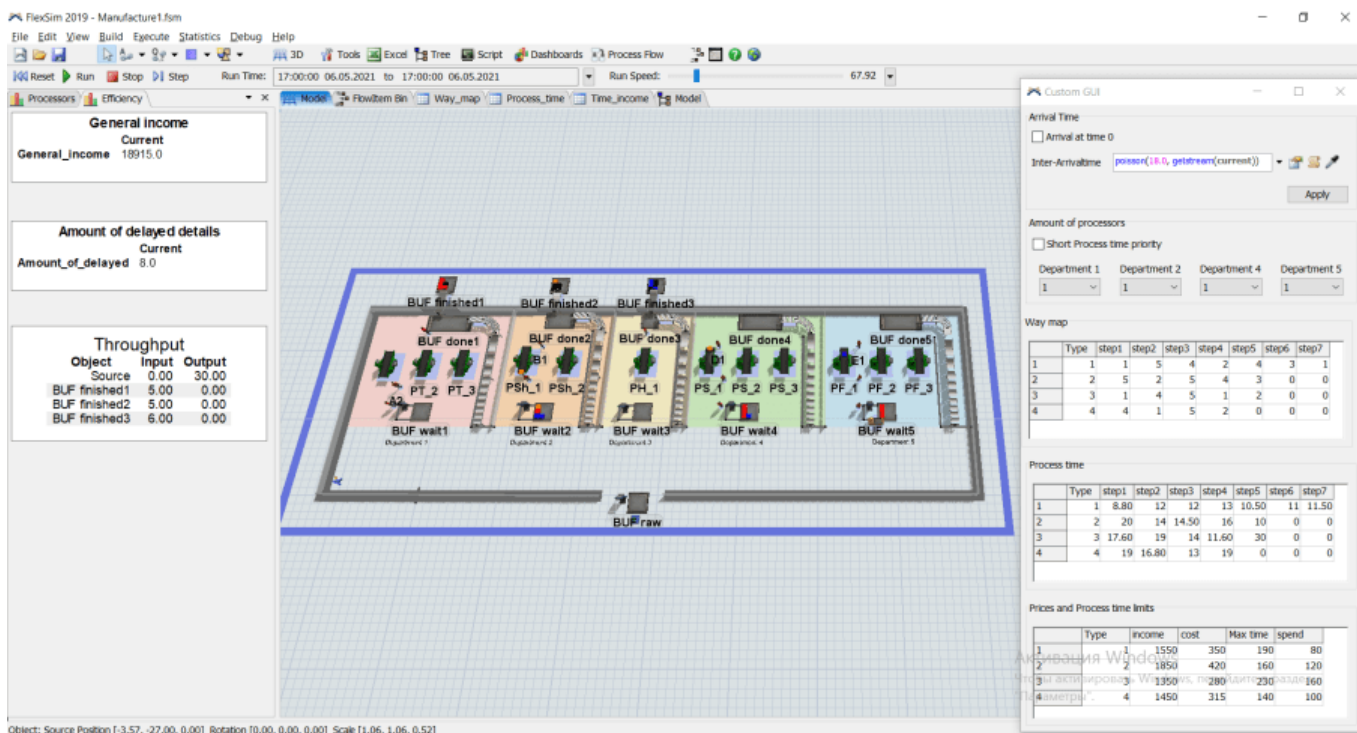


Рисунок 3.45 – Панель статистики Efficiency та результат запуску моделі

Для покращення ефективності ВС було запропоновано перевірити 3 гіпотези, які зазначені у технічному завданні (додаток А).

Гіпотеза А.

Зменшимо час обробки на верстатах відділу 2 на 20% та зменшимо ціну на відповідний відсоток (табл. А.4). Нові значення введемо у таблицю Process time та Prices and Process time limits (Time_income).

Запустимо модель з новими вхідними даними. На панелі статистики Efficiency (рис. 3.46) можна побачити, що кількість оброблених деталей збільшилась разом з кількістю затриманих у системі, а загальний прибуток збільшився всього приблизно на 9%.

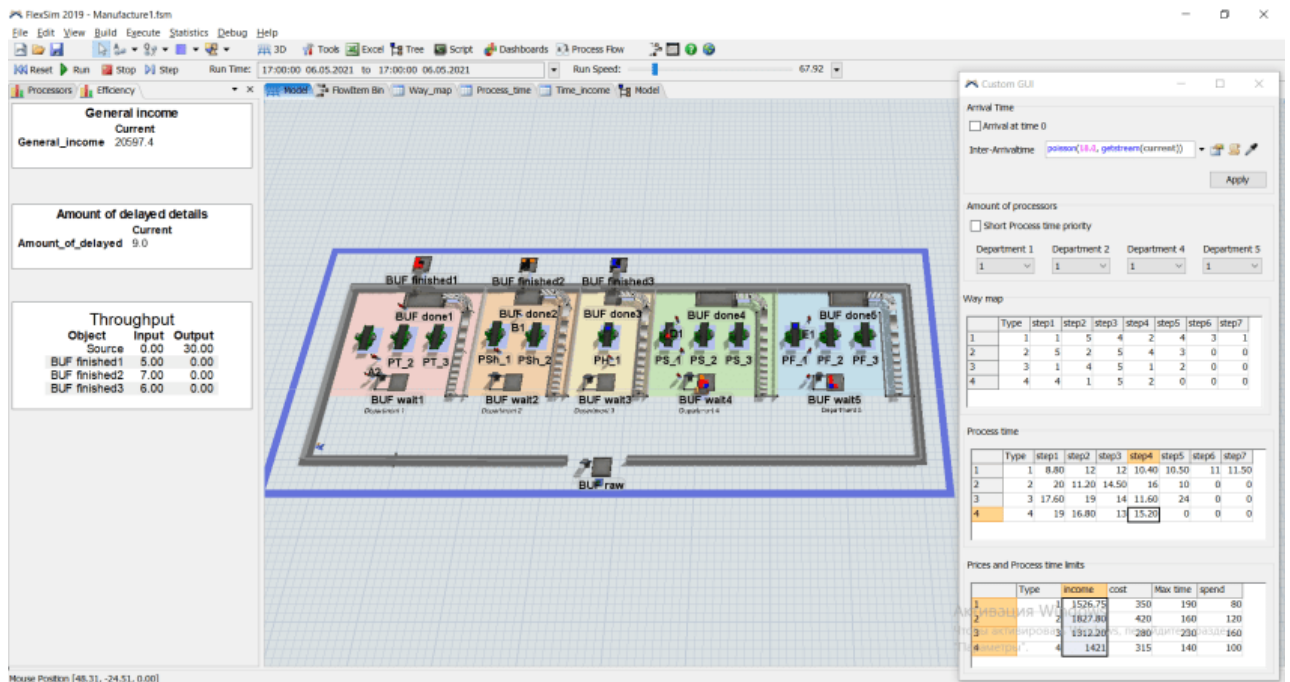


Рисунок 3.46 – Перевірка гіпотези А зі змінами в одному відділі

Зменшено час обробки на верстатах відділу 2 та 4 та встановимо відповідну ціну деталі. З отриманих статистичних даних Effісієнсу видно, що загальний прибуток та кількість оброблених деталей збільшилися відносно початкового стану системи (рис. 3.47).

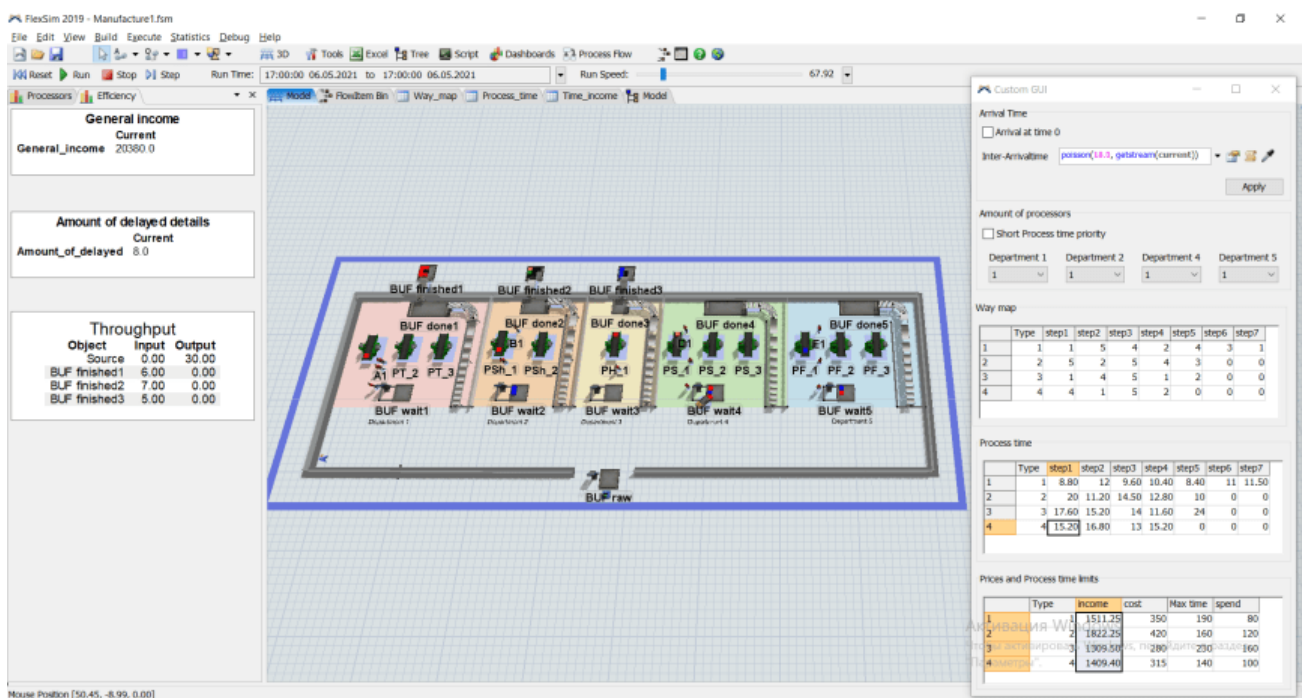


Рисунок 3.47 – Перевірка гіпотези А зі змінами в двох відділах

Результати запуску імітації зменшеним часом обробки у відділах 2, 4 та 5 та у всіх чотирьох відділах, окрім третього, представлені на рис. 3.48 та рис. 3.49.

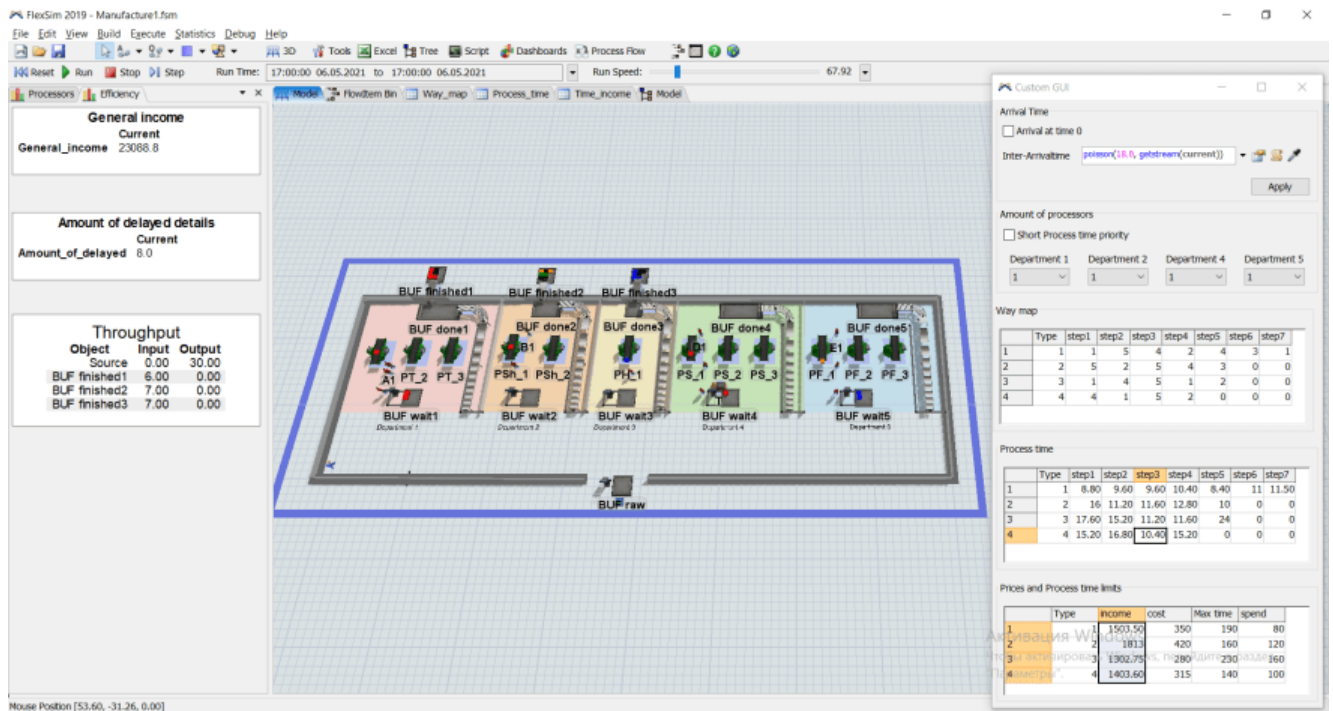


Рисунок 3.48 – Перевірка гіпотези А зі змінами в трьох відділах

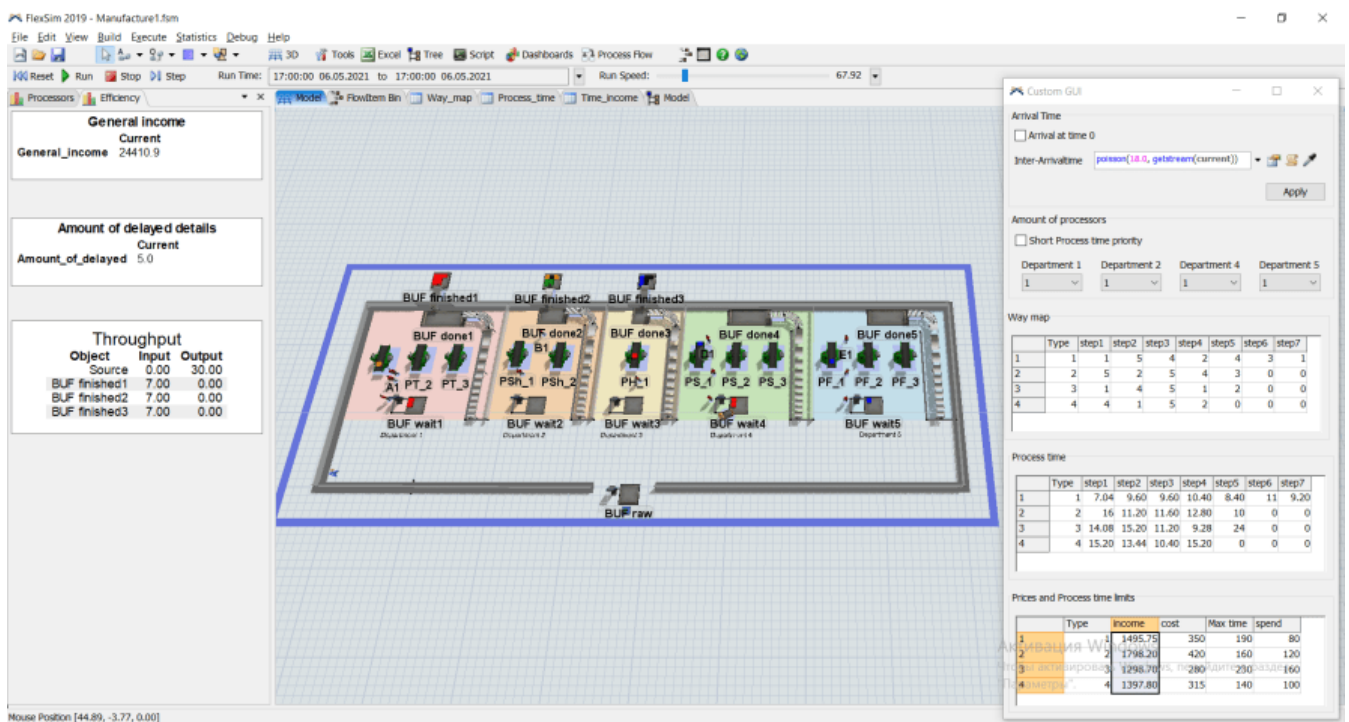


Рисунок 3.49 – Перевірка гіпотези А зі змінами в чотирьох відділах

Отже, при зменшенні часу обробки на верстатах чотирьох найбільш завантажених відділах та зменшенні ціни деталі загальний прибуток та ефективність системи збільшилися.

Гіпотеза В.

Для перевірки гіпотези В збільшимо кількість верстатів на 1 у відділах 2, 4 і 5 та собівартість деталі відповідно до табл. А.5.

Додавання нових робочих машин знизило завантаженість процесорів даних відділів (рис. 3.50), а прибуток та кількість оброблених деталей збільшилися приблизно на 50% (рис. 3.51) відносно результатів початкового запуску системи. При цьому немає оштрафованих за затримку у системі деталей. Але натомість виникла черга у першому відділі.

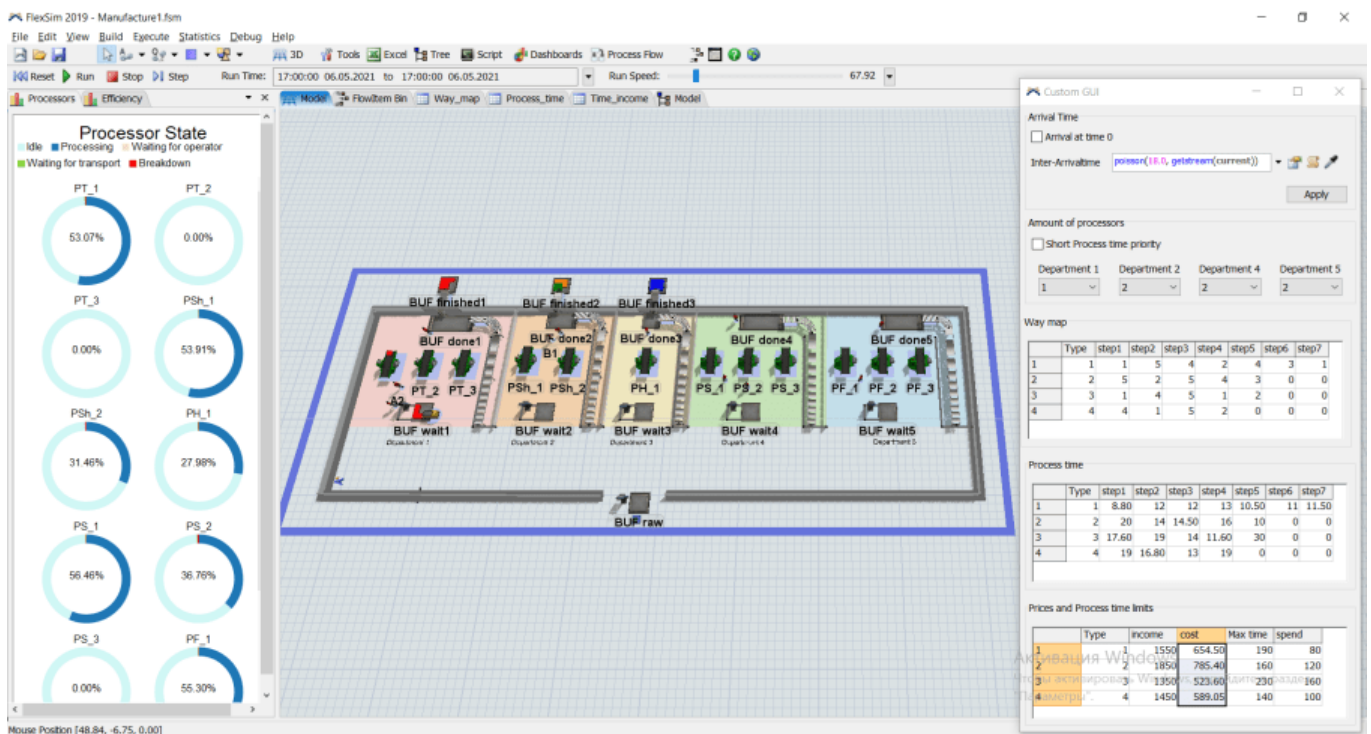


Рисунок 3.50 – Статистика Processors при перевірці гіпотези В і збільшення кількості процесорів на 1 у трьох відділах

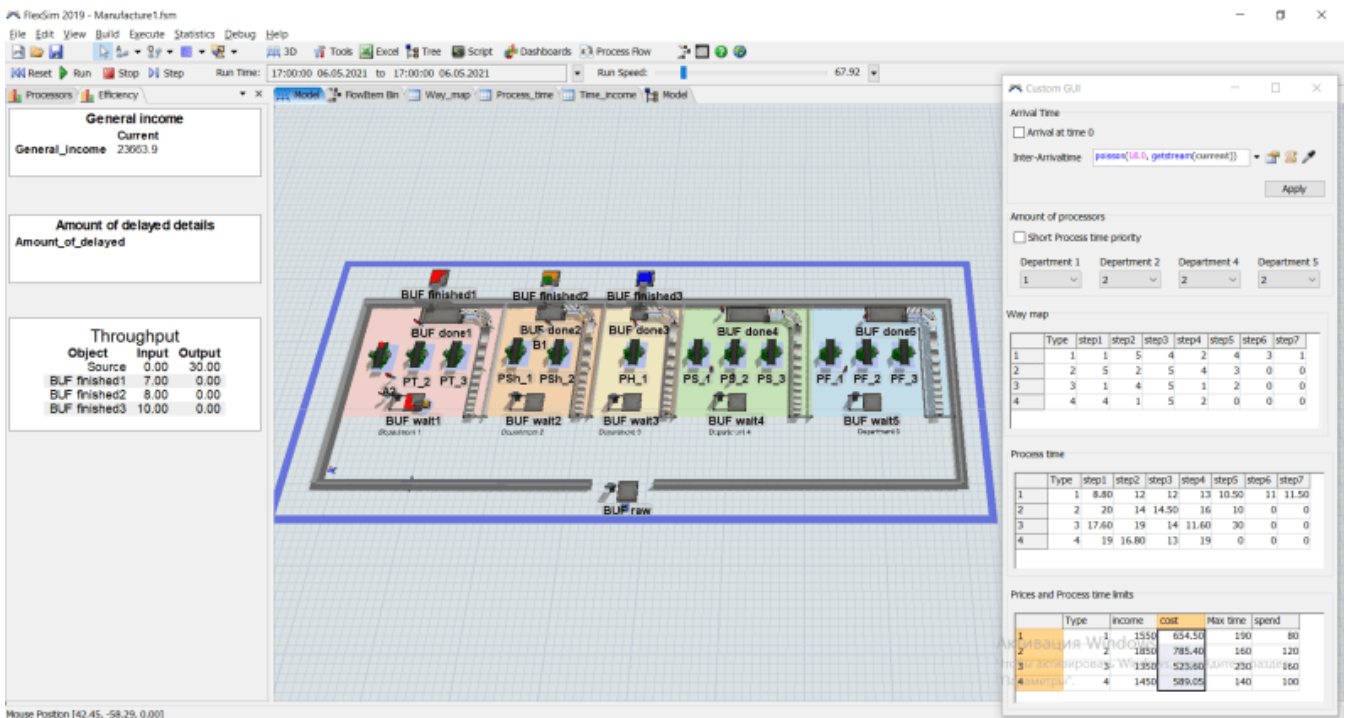


Рисунок 3.51 – Статистика Efficiency при перевірці гіпотези В і збільшення кількості процесорів на 1 у трьох відділах

Додамо один процесор у відділ 1.

Як показали зібрані після запуску моделі статистичні дані (рис. 3.52-3.53), загальний прибуток та обсяг оброблених збільшилися більше, ніж на 50% від початкового значення.

При цьому завантаженість процесорів на нормальному рівні, а кількість деталей, що оброблялись довше допустимого часу, дорівнює 0.

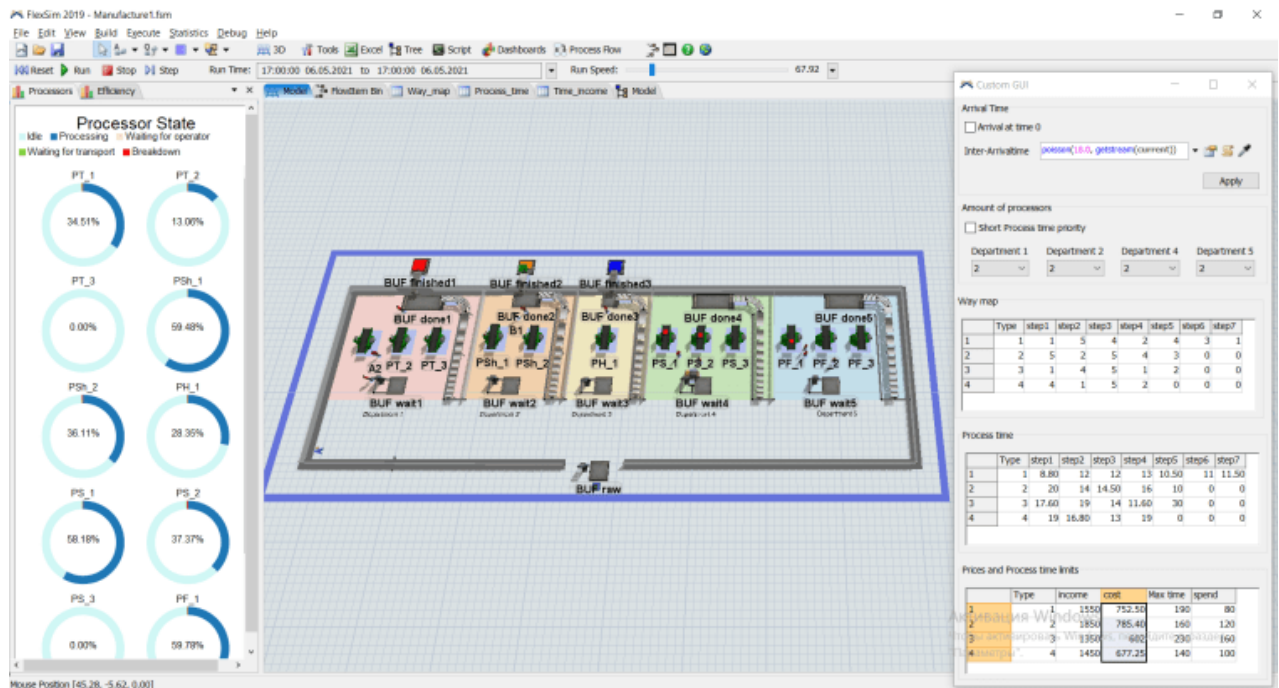


Рисунок 3.52 – Статистика Processors при перевірці гіпотези В і додавання процесора у чотирьох відділах

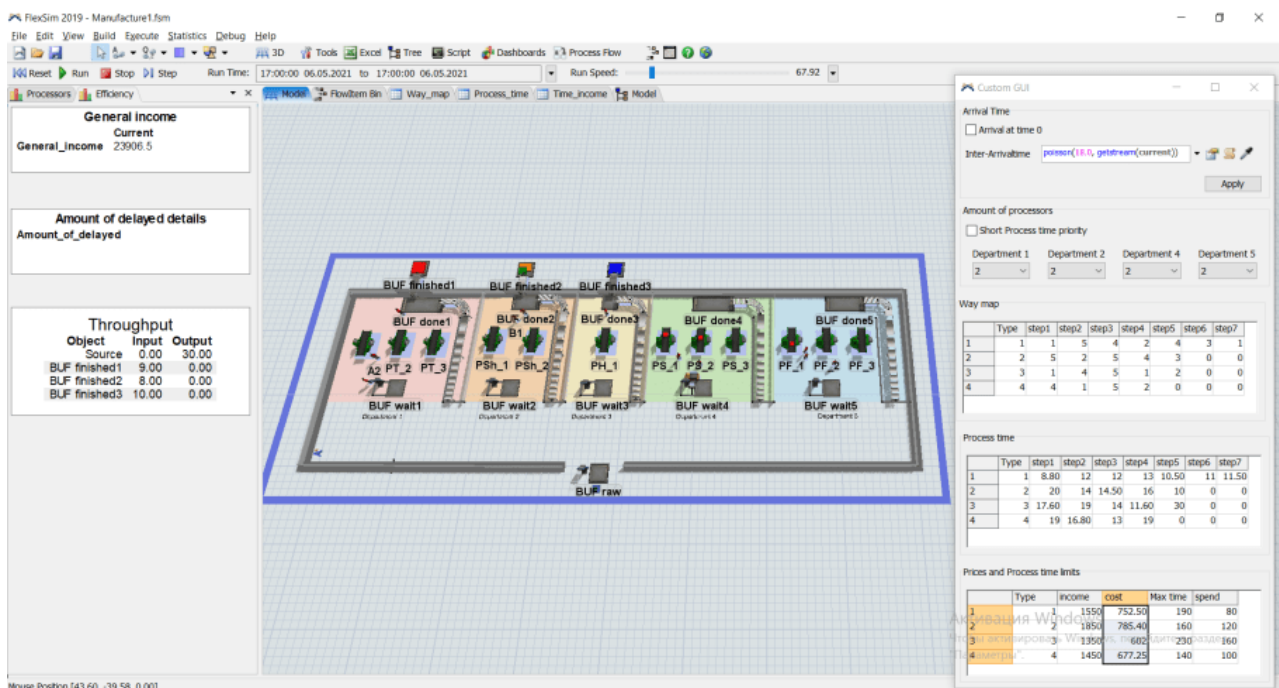


Рисунок 3.53 – Статистика Efficiency при перевірці гіпотези В і додавання процесора у чотирьох відділах

Гіпотеза С.

Для перевірки гіпотези С в інтерфейсі користувача позначимо використання стратегії Short Process time priority і запусимо імітацію.

У випадку з базовими вхідними даними така стратегія підвищення пропускної здатності не є ефективною, що показують результати на рис. 3.54, за якими кількість оброблених деталей за загальний прибуток суттєво не відрізняється від показників початкової системи.

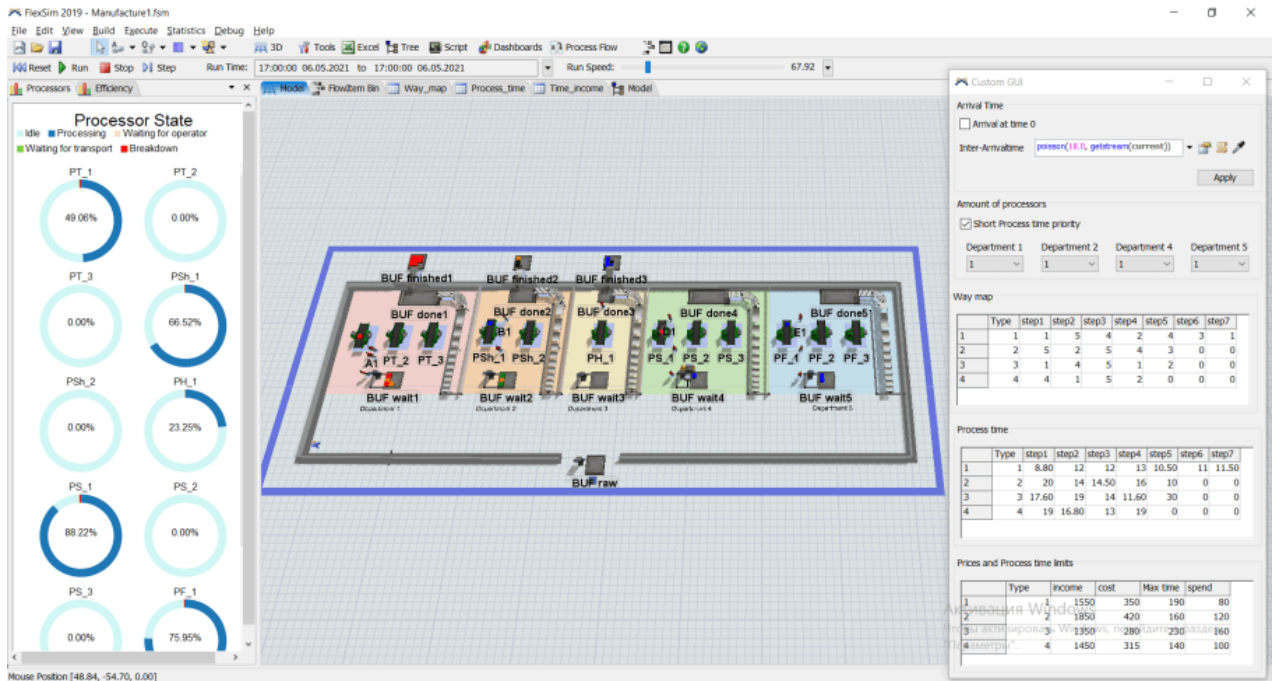


Рисунок 3.54 - Статистика Processors при перевірці гіпотези В

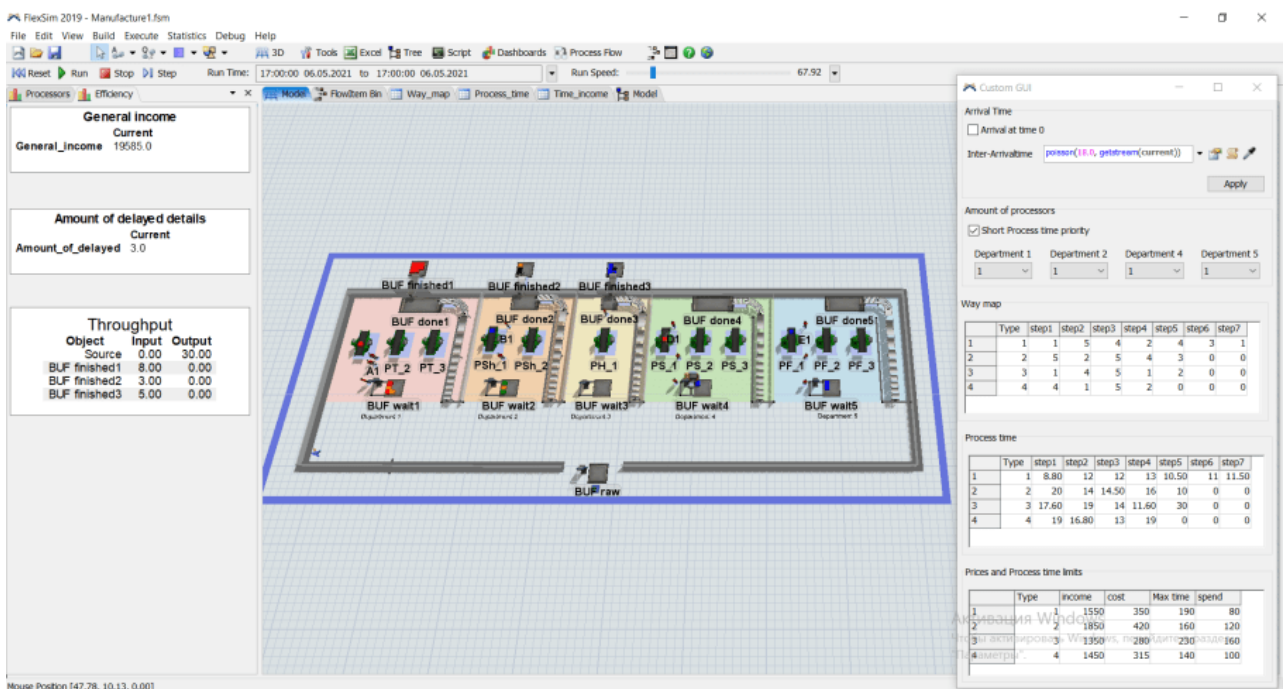


Рисунок 3.55 – Статистика Efficiency при перевірці гіпотези С

3.3. Дослідження роботи виробничої системи

Отримані результати перевірки гіпотез щодо покращення ефективності виробничої системи, що зібрані у таюл. 3.4, показують, що найкращим рішенням серед запропонованих є перехід на новий режим роботи обладнання виробничої дільниці (гіпотеза А), при чому зменшувати час обробки слід на верстатах відділів 1, 2, 4 і 5.

Таблиця 3.4 – Результати перевірки гіпотез

Опис показника статистики	Базові значення	Гіпотеза А				Гіпотеза В		Гіпотеза С
		1	2	3	4	1	2	
Загальний прибуток (General income)	18915	20597,4	20380	23088,8	24410,9	23663,9	23906,5	19585
Кількість затриманих у системі деталей (Amount of delayed details)	8	9	8	8	5	0	0	3
Кількість деталей, що надійшли у дільницю (Source – Output)	30	30	30	30	30	30	30	30
Кількість оброблених та реалізованих деталей (BUF finished 1, 2, 3 – Input)	16	18	18	20	21	25	27	16
Завантаженість процесорів	44,5% - 86,8%	44,5% - 86,8%	45,9% - 78,1%	47,2% - 78,8%	38,7% - 80,3%	28% - 56,5%	13% - 59,8%	49% - 88,2%

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра був проведений аналіз предметної області та останніх досліджень, визначено актуальність роботи.

Було визначено та деталізовано мету та задачі проекту, написано технічне завдання та проведено планування робіт.

У технічному завданні визначено основні вимоги до проекту та імітаційної моделі, описана структура виробничої системи, склад і зміст робіт із розробки імітаційної моделі.

У плануванні робіт деталізовано мету методом SMART, побудовано діаграми WBS для відображення ієрархічної структури робіт та OBS, яка показує організаційну структуру проекту, складено календарний план у вигляді діаграми Ганта, проведено аналіз ризиків проекту.

Проаналізовано методи і засоби реалізації, складено порівняльну характеристику продуктів-аналогів, що могли бути використані для вирішення поставлених задач. Був обраний метод імітаційного моделювання і дискретно-подійний підхід та середовище моделювання FlexSim.

Під час проектування було побудовано контекстну діаграму та діаграми декомпозиції процесу розробки, структурно-функціональну модель у нотації IDEF та концептуальну модель імітаційної моделі OFD.

Для збереження вхідних даних до моделі було побудовано концептуальну, логічну та фізичну моделі бази даних, створено базу даних manufacture за допомогою веб-додатку PhpMyAdmin та СУБД MySQL.

Засобами ПЗ FlexSim було розроблено імітаційну модель виробничої системи відповідно до вимог, визначених у технічному завданні, підключено базу даних, створено інтерфейс користувача для зміни параметрів та вхідних даних моделі. Було досліджено ефективність виробничої системи, перевірено гіпотези. Результати перевірки були зібраними у таблицю для порівняння.

Надання практичних рекомендації щодо покращення її роботи на основі отриманих результатів дослідження системи за допомогою побудованої імітаційної моделі несе практичне значення даного проекту.

Проект орієнтований на здобувачів вищої освіти, які вивчають дисципліну Моделювання систем, а також зацікавлених питаннями дослідження та підвищення ефективності роботи технічних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Securing the future of German manufacturing industry. Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report // The Industrie 4.0 Working Group; National Academy of Science and Engineering; German Research Center for Artificial Intelligence, 2011, 80 p.
2. Michael Grieves, Digital twin: Manufacturing excellence through virtual factory replication 2014 — Режим доступу: http://web.archive.org/web/20170517031855/http://innovate.fit.edu/plm/documents/doc_mgr/912/1411.0_Digital_Twin_White_Paper_Dr_Grieves.pdf
3. Omelyanenko V. Analysis of Institutional and Evolutionary Aspect of High-tech Sectors Development on Instrument Engineering Example // Her. Econ. Sci. Ukr. Academy of Economic Sciences of Ukraine, 2019. № 2(37). P. 93–100.
4. Иванова М., Санникова С. Скрытые тенденции экономического развития промышленности Украины // Траектория науки. Харьков, 2016. Vol. 2, № 4(9). P. 12–16.
5. Борщев А.В. Практическое агентное моделирование и его место в арсенале аналитика // Exponenta PRO. 2004. №3/4 (7/8) С. 38-47.
6. RAYCHAUDHURI S. Introduction to Monte Carlo simulation // Proc. of the Winter Simulation Conf. Miami (USA), 7#10 Dec. 2008. P. 91#100.
7. LAW A. M. How to build valid and credible simulation models // Proc. of the Winter Simulation Conf. Miami (USA), 7-10 Dec. 2008. P. 39#47.
8. Чибіряк Я. І., Овчаренко К.В. Моделювання складальних процесів з використанням програмного забезпечення FlexSim. Інформатика, математика, механіка : тези доп. міжвуз. наук.-техн. конф. викладачів, співробітників, аспірантів і студентів, м. Суми : СумДУ, 2020. С. 181
9. Чибіряк Я. І., Ніколаєнко К.О. Визначення оптимальної програми виробництва на основі моделювання дискретних подій. Інформатика, математика, механіка :

тези доп. міжвуз. наук.-техн. конф. викладачів, співробітників, аспірантів і студентів, м. Суми : СумДУ, 2020. С. 183-184

10. Чибіряк Я., Баранова І., Ніколаєнко К. Метод наскрізного навчання студентів ІТ-спеціальностей імітаційному моделюванню у середовищі FlexSim для пошуку резервів підвищення ефективності автоматизованих систем. // Comput. Technol. Educ. Sci. Prod. 2021. № 42. Р. 119–129
11. Чибіряк Я. І., Заболотний Д.В., Ніколаєнко К.О. Імітаційне моделювання та дослідження діяльності відділення банку. Інформатика, математика, механіка : тези доп. міжвуз. наук.-техн. конф. викладачів, співробітників, аспірантів і студентів, м. Суми : СумДУ, 2021. С. 152-153
12. Чибіряк Я. І., Нестеренко М.В. Моделювання параметрів системи передачі даних за наявності випадкових збоїв. Інформатика, математика, механіка : тези доп. міжвуз. наук.-техн. конф. викладачів, співробітників, аспірантів і студентів, м. Суми : СумДУ, 2021. С.147
13. Чибіряк Я.І., Ніколаєнко К.О., Овчаренко К.В. Дослідження впливу послідовності обробки завдань на завантаженість ресурсів інформаційної системи. Інформатика, математика, механіка : тези доп. міжвуз. наук.-техн. конф. викладачів, співробітників, аспірантів і студентів, м. Суми : СумДУ, 2021. С.148-149
14. Чибіряк Я.І., Овчаренко К.В., Ніколаєнко К.О. Імітаційна модель функціонування мережі зв'язку. Інформатика, математика, механіка : тези доп. міжвуз. наук.-техн. конф. викладачів, співробітників, аспірантів і студентів, м. Суми : СумДУ, 2021. С.150-151
15. FlexSim User Manual Version 2019 // URL: <https://docs.flexsim.com/en/19.0/Introduction/Welcome/>
16. Nahorny V. V., Lavrov E.A., Chybiriak Y.I. FORECASTING INDIVIDUAL RESOURCE OF TECHNICAL SYSTEMS // Radio Electron. Comput. Sci. Control. 2019. № 1.

Додаток А

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку імітаційної моделі
«Імітаційна модель виробничої системи у програмному середовищі FlexSim»

ПОГОДЖЕНО:

Доцент кафедри комп'ютерних наук

_____ Чибіряк Я.І.

Студент групи ІТ-72

_____ Ніколаєнко К.О.

1 Призначення й мета створення імітаційної моделі

1.1 Призначення імітаційної моделі

Імітаційна модель призначена для дослідження показників ефективності роботи виробничої системи за встановленими показниками – завантаженість обладнання, кількість деталей, тривалість обробки яких перевищила допустимий час перебування на робочій ділянці, прибуток від реалізації деталей.

1.2 Мета створення імітаційної моделі

За допомогою моделі перевірити гіпотези:

- Гіпотеза А. Збільшення швидкості обробки деталей на верстатах на 20%.
- Гіпотеза В. Збільшення кількості однотипних верстатів на виробничій ділянці.
- Гіпотеза С. Встановлення послідовності обробки деталей за динамічним пріоритетом.

Перевірити кожен гіпотезу та за результатами перевірки надати практичні рекомендації, щодо підвищення ефективності роботи виробничої системи.

1.3 Цільова аудиторія

Цільовою аудиторією можуть бути здобувачі вищої освіти, які вивчають дисципліну Моделювання систем, а також представники з виробництва (головні інженери, керівники структурних підрозділів та ін.), які цікавляться питаннями дослідження та підвищення ефективності роботи технічних систем.

2 Вимоги до імітаційної моделі

2.1 Вимоги до імітаційної моделі в цілому

2.1.1 Вимоги до структури й функціонування імітаційної моделі

Необхідно розробити імітаційну модель виробничої системи, яка:

- повністю відображає структуру реальної системи, описаної в постановці задачі;
- алгоритм функціонування та процеси, що відбуваються в моделі, повинні повністю відповідати алгоритму роботи реальної системи, описаної в постановці задачі;
- модель повинна мати зручний інтерфейс користувача для зміни вхідних даних;
- модель повинна бути налаштованою для перевірки гіпотез.

2.1.2 Системні вимоги

Для запуску, дослідження моделі та перевірки гіпотез потрібно мати програмне забезпечення FlexSim; вимоги до операційної системи комп'ютера – Windows 7 і вище.

2.1.3 Вимоги до збереження інформації

Вхідні дані, необхідні для роботи моделі зберігаються у табличному форматі у базі даних, що реалізована засобами системи управління базами даних MySQL; підключення до БД та зчитування даних організовано програмно та відбувається в автоматичному режимі.

2.2 Опис виробничої системи

2.2.1 Схеми виробничої системи

Схеми виробничої системи зображена на рис. А.1.

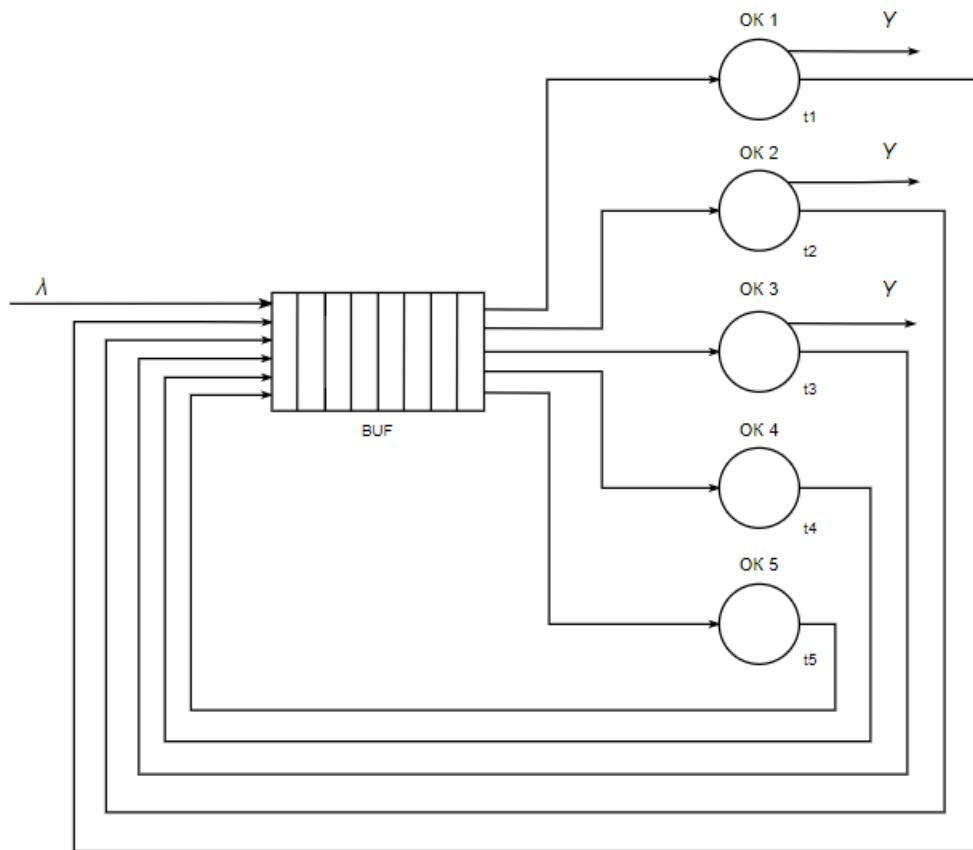


Рисунок А.1 – Схема виробничої системи

Умовні позначення на рис. А.1:

λ – інтенсивність надходження деталей на обробку;

BUF – буферна зона, призначена для очікування деталями процесу обробки;

ОК 1 – операційний комплекс токарних верстатів;

ОК 2 – операційний комплекс хонінгувальних верстатів;

ОК 3 – операційний комплекс шліфувальних верстатів;

ОК 4 – операційний комплекс свердлильних верстатів;

ОК 5 – операційний комплекс фрезерних верстатів;

$t_1 = t_2 = t_3 = t_4 = t_5 = \text{exponential}(\text{mean}(\text{type}, \text{step}))$ – час обробки деталей;

$\text{mean}(\text{type}, \text{step})$ – параметр експоненціального закону розподілу (середнє значення);

type – тип деталі;

step – етап обробки деталі;

Y – деталі, що пройшли усі етапи обробки, виходять із системи.

2.2.2 Структура виробничої системи

Виробнича дільниця має п'ять типів верстатів: токарні, свердлильні, шліфувальні, фрезерні, хонінгувальні. На ділянці обробляються деталі чотирьох типів. Кожен тип деталі має послідовність обробки, яка задається маршрутною картою. Технологічні маршрути деталей, кількість етапів обробки, послідовність обробки, середній час обробки для кожного типу деталей задано згідно маршрутної карти руху деталей (табл. А.1).

Часові інтервали надходження деталей у систему представлені у табл. А.2. Типи деталей розподілені рівномірно. Характеристики часу виготовлення деталі кожного типу і прибуток від реалізації наведені в табл. А.3.

Таблиця А.1 – Маршрутна карта руху деталей по дільниці

Тип деталі	Кількість етапів обробки	Послідовність проходження деталей	Час обробки, хв
1	7	Токарний Фрезерний Свердлильний Шліфувальний Свердлильний Хонінгувальний Токарний	8,8 12 12 13 10,5 11 11,5
2	5	Фрезерний Шліфувальний Фрезерний Свердлильний Хонінгувальний	20 14 14,5 16 10
3	5	Токарний Свердлильний Фрезерний Токарний Шліфувальний	17,6 19 14 11,6 30
4	4	Свердлильний Токарний Фрезерний Шліфувальний	19 16,8 13 19

Таблиця А.2 – Часові інтервали надходження деталей у систему

11	15	18	21	18	24	16	20	18	20
15	17	18	15	13	18	22	19	29	26
17	12	10	14	13	12	16	23	21	16
16	16	15	21	20	21	21	29	22	16
12	19	18	16	21	24	13	14	17	21
21	18	14	13	22	20	5	18	22	26
20	16	25	13	21	10	20	22	17	17
12	13	19	16	24	17	14	14	19	18
15	25	23	21	12	18	18	13	24	22
19	16	19	17	19	20	22	22	20	21

Таблиця А.3 – Загальний час виготовлення деталей та прибуток

Тип деталі	Прибуток, грн.	Собівартість, грн.	Допустимий час виготовлення Max_time , хв	Витрати, якщо $time > Max_time$, грн.
1	1550	350	1890	80
2	1850	420	1600	120
3	1350	280	2300	160
4	1450	315	1400	100

Умовні позначення в табл. А.3:

$time$ – тривалість виробничого циклу деталі.

Для покращення роботи виробничої системи, необхідно перевірити наступні гіпотези:

Гіпотеза А. Збільшити швидкість обробки деталей на верстатах на 20%.

Такі зміни можуть привести до деякої втрати якості, яка зменшить прибуток. По причині збільшення загальної продуктивності і скорочення витрат, пов'язаних із затримкою деталі під час обробки, прибуток може зрости. Зміна прибутку у 7 залежності від режиму роботи обладнання наведена у табл. А.4.

Таблиця А.4 - Зниження ціни деталі при зміні режиму роботи верстатів

Кількість відділів, у яких на станках збільшено швидкість обробки	Зменшення ціни, залежно від типу деталі, %			
	Тип 1	Тип 2	Тип 3	Тип 4
1	1,5	1,2	2,8	2
2	2,5	1,5	3	2,8
3	3	2	3,5	3,2
4	3,5	2,8	3,8	3,6
5	4	3	4,2	4

Гіпотеза В. Збільшення кількості однотипних верстатів на виробничій дільниці.

При цьому передбачається, що якість виготовлення деталей не буде зменшуватись, але можливим є зменшення прибутку по причині появи додаткових витрат на обслуговування нових верстатів (амортизаційні відрахування) (табл. А.5).

Таблиця А.5 – Збільшення собівартості виготовлення деталі у залежності від кількості доданих верстатів

Кількість доданих станків	Збільшення собівартості деталі, виготовленої на верстаті, %				
	Токарному	Шліфува-льному	Хонінгува-льному	Свердлиль-ному	Фрезерному
1	28	30	15	25	32
2	50	55	22	39	64

Гіпотеза С.

Якщо виконувати послідовність обробки деталей за динамічним пріоритетом, то припускається зменшення загальної суми штрафів.

Динамічний пріоритет передбачає таку послідовність обробки на верстатах, при якій деталі обробляються у порядку збільшення тривалості їх обробки.

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача представлені у табл. А.6.

Таблиця А.6 – Потреби користувача

ІД	Потреби користувача	Джерело
UN-01	Можливість зміни вхідних даних до моделі	Дослідник
UN-02	Перегляд моделі та процесу імітації	Дослідник
UN-03	Перевірка гіпотез за допомогою моделі	Дослідник
UN-04	Перегляд статистичних даних моделі	Дослідник
UN-05	Можливість зміни структури моделі (за потреби)	Дослідник

2.3.2 Функціональні вимоги

На основі потреб користувача були визначені такі функціональні вимоги:

- можливість змінювати структуру моделі;
- зміна вхідних даних;
- перевірка гіпотез;
- збір та відображення статистичних даних;
- керування процесом імітації.

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в табл. А.7.

Таблиця А.7 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Інтерфейс користувача	М	Виконує роль зв'язку з користувачем
SR-02	База даних із вхідними параметрами	М	Зберігає вхідні дані
SR-03	Наявність модуля зміни вхідних даних моделі	М	Надає можливість користувачу змінювати вхідні дані моделі
SR-04	Наявність модуля збору статистики	S	Відповідає за збір та виведення статистичних даних
SR-05	Наявність модуля управління процесом імітації	М	Надає можливість запускати, зупиняти модель, масштабувати модельний час по відношенню до реального та ін.
SR-06	Наявність модуля візуалізації у 3D просторі	C	Відповідає за відображення моделі у тривимірному вигляді як у статичному стані так і в динаміці
SR-07	Модуль аналізу даних ExpertFit	М	Надає можливість виконати ідентифікацію закону розподілу

Умовні позначення в табл. А.7:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація імітаційної моделі та її дослідження відбувається з використанням:

- FlexSim 19.0.0
- MySQL 8.0
- Мови програмування C++
- Модуля аналізу даних ExpertFit
- 3Ds Max

2.4.2 Вимоги до лінгвістичного забезпечення

Інтерфейс користувача має бути виконаний англійською мовою.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Операційна система Windows 7 і вище.
- Програма FlexSim версії 19.0.0.

3 Склад і зміст робіт зі створення імітаційної моделі

Докладний опис етапів роботи зі створення імітаційної моделі наведено в табл. А.8.

Таблиця А.8 – Етапи створення імітаційної моделі

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Аналіз предметної області	15 днів
2	Постановка задачі	5 днів
3	Написання технічного завдання	29 днів

Продовження таблиці А.8

№	Склад і зміст робіт	Строк розробки (у робочих днях)
4	Побудова плану реалізації	7 днів
5	Проектування імітаційної моделі	6 днів
6	Проектування бази даних	2 дні
7	Розробка 3D моделей	7 днів
8	Побудова моделі	7 днів
9	Розробка інтерфейсу користувача	5 днів
10	Перевірка гіпотез	4 дні
11	Написання документації	7 днів
	Загальна тривалість робіт	94 дні

4 Вимоги до складу й змісту робіт із введення імітаційної моделі в експлуатацію

Для використання створеної моделі користувач повинен мати на своєму комп'ютері встановлену програму FlexSim версії 19.0.0. Потрібно відкрити файл формату .fsm, який містить побудовану та налаштовану модель, та запустити інтерфейс користувача для зручнішого використання.

Додаток Б

Планування робіт

Деталізація мети проекту методом SMART. У результаті дипломного проектування потрібно отримати імітаційну модель виробничої системи та за її допомогою перевірити гіпотези.

Результати деталізації методом SMART розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити імітаційну модель виробничої системи, перевірити гіпотези, за результатами надати практичні рекомендації щодо покращення роботи системи.
Measurable (вимірювана)	Результатом виконання проекту є оцінка замовника.
Achievable (досяжна)	Для реалізації системи використовується середовище FlexSim, модуль аналізу даних ExpertFit, мова програмування C++, для збереження даних – СУБД MySQL 8.0, розробка 3D моделей здійснюється у програмі Autodesk 3ds Max.
Relevant (реалістична)	Наявні усі необхідні технічні та програмні засоби. Розробники мають достатню кваліфікацію для виконання поставлених задач.
Time-framed (обмежена у часі)	Проект має часові обмеження і повинен бути реалізований згідно з календарним планом.

Планування змісту структури робіт. Визначення змісту проекту – це процес розробки детального опису проекту та продукту. Зміст проекту визначається під час планування і описується більш детально в міру надходження інформації. Для планування змісту структури робіт використовується WBS діаграма. WBS (Work Breakdown Structure) – ієрархічна структура робіт (ICP), або структура декомпозиції – являє собою схему, де відображені усі задачі проекту, які ієрархічно пов'язані одна з одною та проектом у цілому.

На діаграмі WBS (рис. Б.1) представлено структуру та опис робіт проекту.

Планування структури організації, для впровадження готового проекту (OBS). Організаційна структура проекту (Organizational Breakdown Structure, OBS) – ієрархічна структура управління проектом з організованими внутрішніми процесами, ролями та зонами відповідальності. OBS показує відносини між учасниками проекту.

Діаграма OBS зображена на рис. Б.2. Список виконавців, що функціонують в проекті знаходиться в табл. Б.2.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Ніколаєнко К.О.	Розробка імітаційної моделі та інтерфейсу користувача, розробка та перевірка гіпотез.
Проектувальник	Ніколаєнко К.О.	Проектування 3D елементів моделі.
Косультант проекту	Чибіряк Я.І.	Формує завдання на розробку проекту.
Менеджер проекту	Ніколаєнко К.О.	Планування робіт, моніторинг виконання проекту, розподіл ресурсів та задач між учасниками.

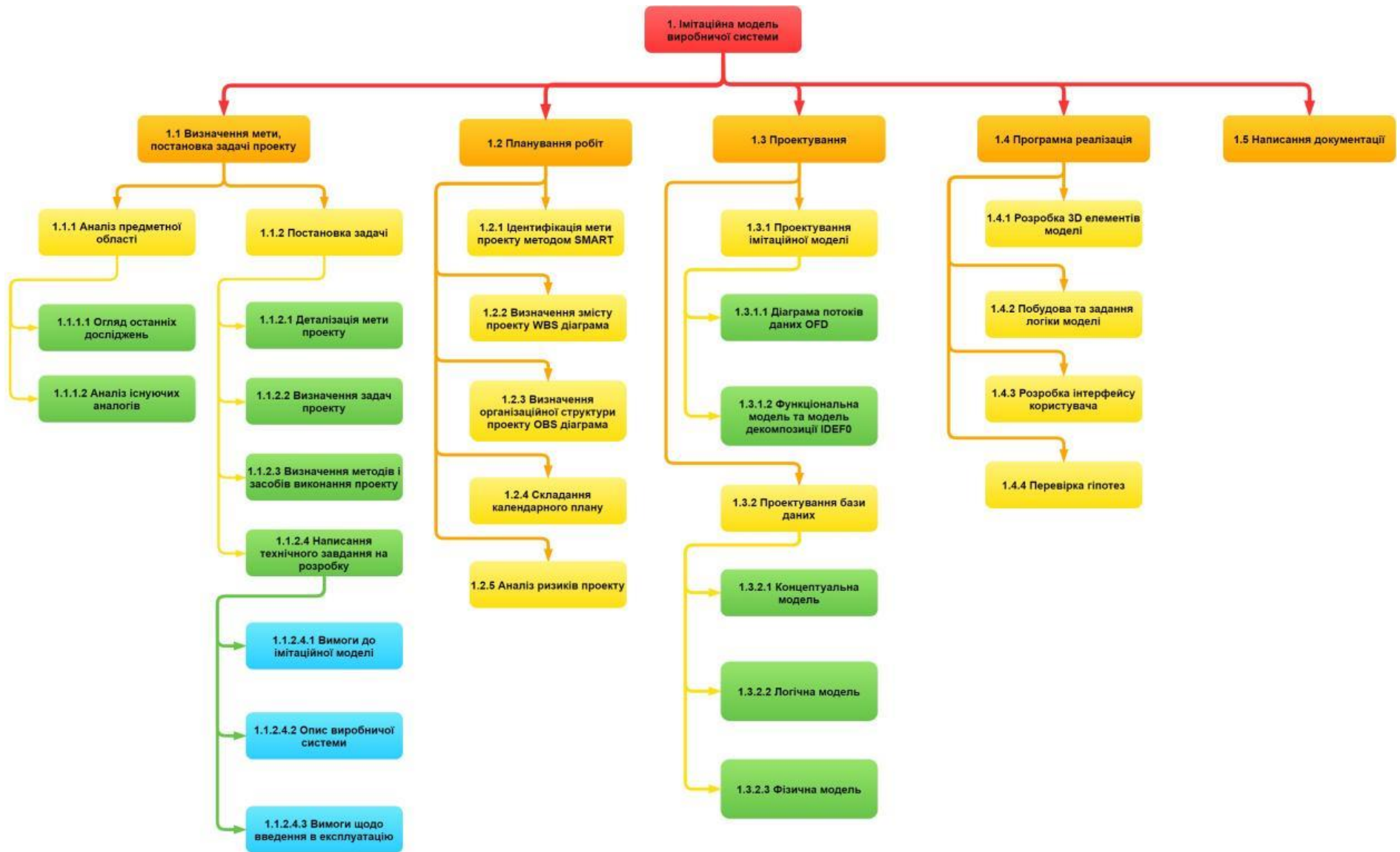


Рисунок Б.1 – WBS Структура робіт проекту

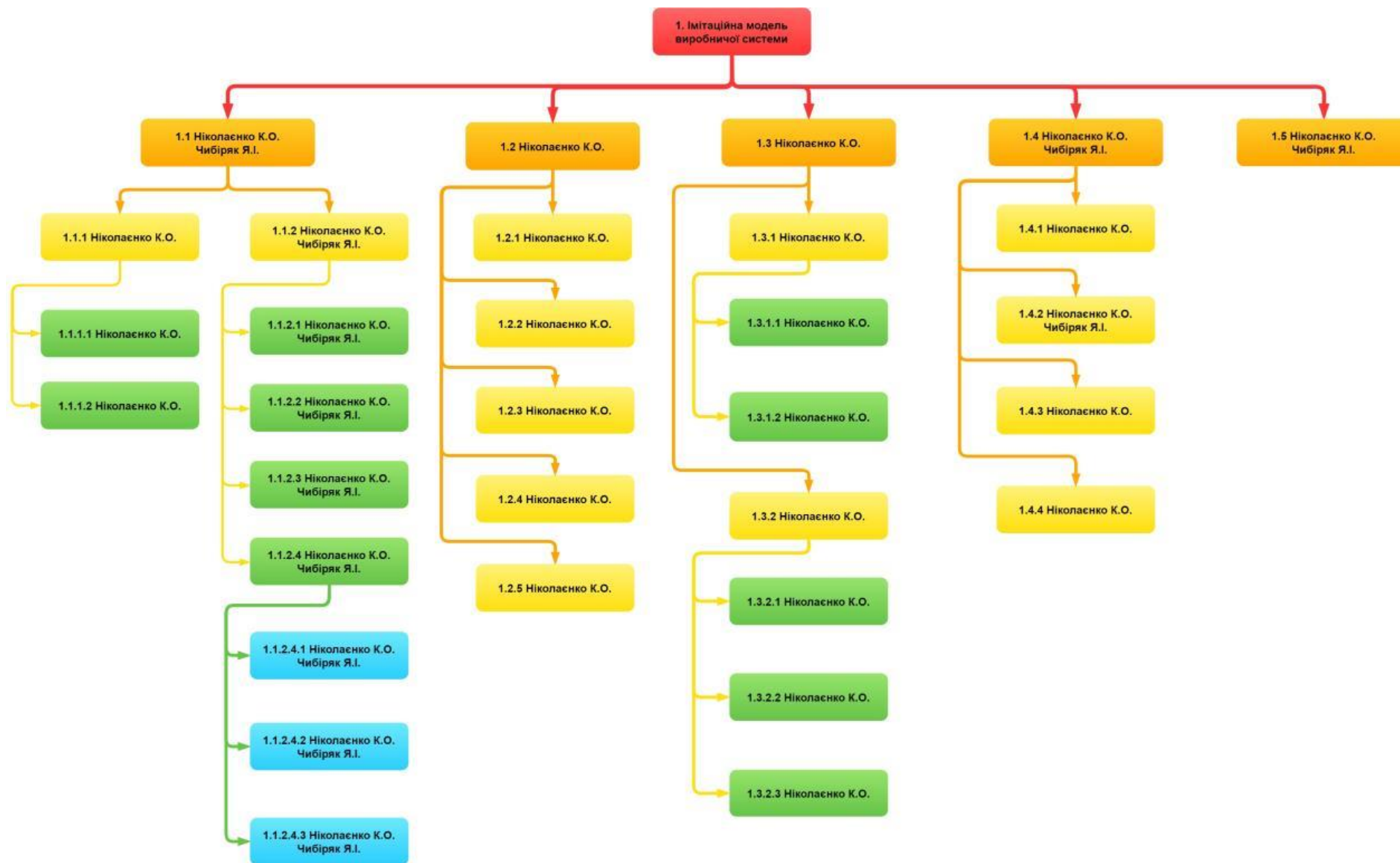


Рисунок Б.2 – Організаційна структура проекту (OBS)

Діаграма Ганта. Для складання календарного графіку проекту використаємо діаграму Ганта. Діаграма Ганта - це візуальне відображення задач проекту на проміжку часу.

Календарний план побудовано з урахуванням вихідних днів, свят та робочого часу. Діаграма Ганта зображена на рис. Б.3.

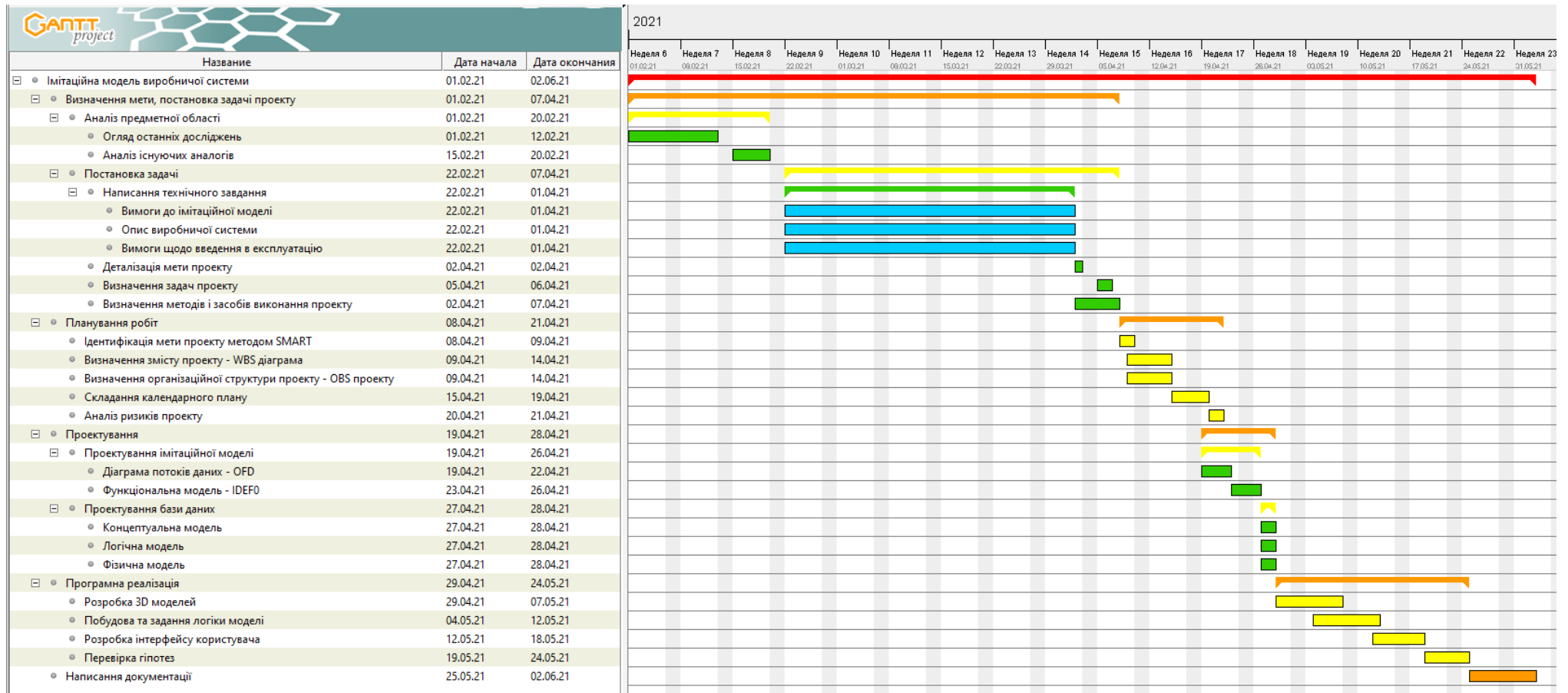


Рисунок Б.3 – Діаграма Ганта

Аналіз ризиків. Аналіз ризиків включає якісну і кількісну оцінку ризиків.

Якісний аналіз дозволяє виявити можливі види ризиків, визначити й описати причини і фактори, що впливають на рівень ризику. Також необхідно визначити й оцінити наслідки від реалізації виявлених ризиків і запропонувати способи мінімізації негативного впливу на виконання проекту.

При кількісній оцінці визначається ймовірність виникнення та вплив наслідків від реалізації ризиків на проект, а також ті ризики, що потребують швидшого реагування. Оцінка ризиків за показниками ймовірності виникнення та впливу на проект зазначена у табл. Б.3.

Розподіл ризиків по категоріях представлений у вигляді матриці ризиків (табл. Б.4).

Ризики, що можуть виникнути під час виконання проекту, оцінка вірогідності їх виникнення та впливу, рівень ризику та план дій по мінімізації впливу зазначені у табл. Б.6.

Таблиця Б.3 – Оцінка ризиків за показниками ймовірності виникнення та впливу на проект

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

Таблиця Б.4 – Матриця ймовірності виникнення та впливу ризиків

Вірогідність виникнення	Матриця ризиків		
1			
2			
3			
Ступінь впливу	1	2	3

- зелений колір – мінімальні ризики;
- жовтий колір – допустимі ризики;
- червоний колір – недопустимі (критичні) ризики.

На основі побудованої матриці класифікуємо ризики за їх рівнем (табл. Б.5).

Таблиця Б.5 – Шкала оцінювання за рівнем ризику

№	Назва	Межі
1	Мінімальні	1 – 2
2	Допустимі	3 – 4
3	Недопустимі	6 - 9

Таблиця Б.6 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

ID	Опис ризику	Ймовірність виникнення	Вплив ризику	Рівень ризику	Реакція на виникнення ризикованої ситуації
RS_1	Поява альтернативного продукту	Низька	Високий	3	Провести дослідження альтернативного продукту, виявити його недоліки та вдосконалити проект на основі зібраної інформації, якщо це можливо.
RS_2	Невизначеність щодо завдання при виконанні проекту	Середня	Високий	6	Чітко визначити та погодити із замовником завдання на розробку, значення використовуваних термінів, функціональні та нефункціональні вимоги до продукту, методи та засоби реалізації.
RS_3	Недостатня кваліфікація розробників проекту	Середня	Середній	4	Надати персоналу можливість підвищити кваліфікацію, забезпечивши додатковими курсами та іншими заходами.
RS_4	Нераціональне використання часу на виконання проекту	Висока	Високий	9	Побудувати календарний план робіт проекту, визначити часові проміжки для кожного етапу, звернувши увагу на більш важливі та складні етапи, що потребують більше часу. Чітко дотримуватись розробленого плану.
RS_5	Внесення змін у ТЗ під час виконання проекту	Середня	Високий	6	Визначити завдання проекту, усі вимоги до продукту та можливі додаткові функції до початку реалізації. Під час етапу розробки вносити тільки ті зміни, що не впливають на реалізацію основного функціоналу продукту.
RS_6	Обраний метод розробки виявився неефективним	Середня	Високий	6	Провести аналіз існуючих методів, спроб та результатів їх використання й обрати найбільш ефективний.
RS_7	Помилки проектування	Висока	Високий	9	Виділити перевірку та виправлення помилок як окремий етап розробки проекту. Великі задачі розбити на підзадачі, перевіряти та зберігати проміжні результати на кожному етапі, погоджувати їх із замовником.
RS_8	Втрата даних через збої в роботі програмного забезпечення	Низька	Високий	3	Якмога частіше зберігати резервні копії проекту, особливо після суттєвих змін. Перед використанням програмного забезпечення перевірити сумісність ОС та загальних характеристик комп'ютера з використовуваним ПЗ.

