

ЗМІСТ

Скорочення та умовні позначки.....	4
Вступ.....	6
1 Імітаційне моделювання як метод дослідження складних систем.....	8
1.1 Поняття моделей і моделювання.....	8
1.2 Приклади використання імітаційних моделей.....	14
1.3 Переваги імітаційного моделювання.....	15
2 Основні підходи в імітаційному моделюванні.....	16
2.1 Системна динаміка.....	17
2.2 Дискретно-подієве моделювання.....	19
2.3 Агентне моделювання.....	19
3 Агентне моделювання в AnyLogic.....	22
3.1 Агенти в AnyLogic.....	24
3.2 Поведінка агентів.....	24
3.3 Інтерфейси агентів.....	26
3.4 Архітектура агентних моделей.....	26
3.5 Взаємодія агентів з середовищем.....	28
4 Моделювання процесу постачання товарів.....	29
4.1 Постановка завдання.....	29
4.2 Декомпозиція задачі на етапи для побудови моделі в AnyLogic.....	29
4.3 Реалізація моделі.....	30
4.3.1 Формування вихідних даних.....	30
4.3.3 Обробка замовлення.....	40
4.4. Запуск та оптимізація моделі.....	44
Висновки.....	50
Перелік джерел посилання.....	51

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АОМ	– агент-орієнтовані моделі
ГІС	– географічна інформаційна система
СД	– системна динаміка
ТЗ	– технічне завдання
PBF	–Protocolbuffer Binary Format (Протокольний буферізований двійковий формат)
GPSS	– General Purpose Simulation System (Система моделювання загального призначення)
OSM	– Open Street Map (Відкрита карта вулиць)

ВСТУП

Моделювання можна розглядати як один із способів вирішення проблем, що виникають в реальному світі: в техніці, виробництві, обслуговуванні, маркетингу, фінансах, охороні здоров'я, транспорті тощо. Воно застосовується у разі, якщо експерименти з реальними об'єктами/ системами неможливі або дуже дорогі. Моделювання дозволяє оптимізувати систему до її реалізації.

Основні підходи в імітаційному моделюванні – це системна динаміка, дискретне-подієве моделювання, під яким треба розуміти будь-який розвиток ідей GPSS, агентне моделювання. Системна динаміка і дискретне-подієве моделювання – традиційні усталені підходи, агентне моделювання – відносно новий підхід. Агентне моделювання може застосовуватися практично на будь-якому рівні абстракції і в будь-яких масштабах. Агенти можуть представляти пішоходів, автомобілі або роботів в фізичному просторі, клієнта або продавця на середньому рівні, або ж конкуруючі компанії на високому рівні абстракції.

Практично всі присутні на ринку інструменти імітаційного моделювання розроблені для підтримки одного певного підходу. Для системної динаміки є всього чотири інструменти. Дискретно-подієве ж моделювання підтримується десятками різних інструментів. У світі динамічних систем домінує MATLAB Simulink. Для агентного моделювання до останнього часу не існувало жодного комерційного інструменту, тільки бібліотеки на Java або C ++, розроблені в різних університетах.

У цей час для створення професійних агентних моделей можна використовувати пакет AnyLogic. Він підтримує мовні конструкції для завдання поведінки агентів, їх взаємодії, моделювання середовища, а також має найбагатші анімаційні можливості. AnyLogic дозволяє описувати різні частини великих гетерогенних систем, використовуючи різні підходи, об'єднуючи системну динаміку, дискретне-подієве моделювання і агентне моделювання в одній

моделі. Це надає можливості використовувати розробникам багатопідхідне імітаційне моделювання.

Метою дипломної роботи є створення імітаційної моделі ланцюгів постачання комплектуючих з використанням багатоагентного підходу в середовищі моделювання AnyLogic.

Для досягнення поставленої мети в роботі необхідно вирішити наступні завдання:

- провести дослідження різних підходів імітаційного моделювання;
- обґрунтувати вибір агентної моделі для вирішення завдання;
- обґрунтувати вибір середовища моделювання;
- вивчити особливості використання ГІС карт в середовищі моделювання;
- побудувати модель постачання комплектуючих;
- виконати оптимізацію моделі.

Структура дипломної роботи складається з вступу, чотирьох розділів, висновків, переліку посилань на 12 найменувань. Повний обсяг проекту становить 52 сторінки, містить 34 рисунка.

1 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ЯК МЕТОД ДОСЛІДЖЕННЯ СКЛАДНИХ СИСТЕМ

1.1 Поняття моделі і моделювання

Модель в широкому сенсі – це будь-який образ, аналог уявний або встановлене зображення, опис, схема, креслення, карта і тощо., процесу або явища, який використовується в якості його заміника або представника. Сам об'єкт, процес або явище називається оригіналом цієї моделі.

Моделювання – це дослідження об'єкту або системи об'єктів, шляхом побудови і вивчення їх моделей. Це використання моделей для визначення або уточнення характеристик і раціоналізації способів побудови знову конструйованих об'єктів.

На ідеї моделювання базується будь-який метод наукового дослідження, при цьому, в теоретичних методах використовуються різного роду знакові, абстрактні моделі, в експериментальних – предметні моделі.

При дослідженні складне реальне явище замінюється деякою спрощеною копією або схемою, іноді така копія служить тільки для того, щоб запам'ятати і при наступній зустрічі упізнати потрібне явище. Іноді побудована схема відбиває які – то істотні риси, дозволяє розібратися в механізмі явища, дає можливість передбачити його зміну. Одному і тому ж явищу можуть відповідати різні моделі.

Завдання дослідника – передбачати характер явища і хід процесу. Для того, щоб виконати моделювання.

Іноді, буває, що об'єкт доступний, але експерименти з ним дорогостоящі або привести до серйозних екологічних наслідків. Знання про такі процеси отримують за допомогою моделей.

Важливий момент – сам характер науки припускає вивчення не одного конкретного явища, а широкого класу споріднених явищ. Припускає необхідність формулювання якихось загальних категоричних тверджень, які називаються законами. Природно, що при такому формулюванні багатьма под-

робицями нехтують. Щоб чіткіше виявити закономірність свідомо йдуть на огрублення, ідеалізацію, схемну, тобто вивчають не з само явище, а більш менш точну її копію або модель. Усі закони – це закони про моделі, а тому немає нічого дивного в тому, що з часом деякі наукові теорії визнаються непридатними. Це не призводить до краху науки, оскільки одна модель замінилася іншої сучаснішою.

Особливу роль в науці грають математичні моделі, будівельний матеріал і інструменти цих моделей – математичні поняття. Вони накопичувалися і удосконалювалися в течії тисячоліть. Сучасна математика дає виключно потужні і універсальні засоби дослідження. Практично кожне поняття в математиці, кожен математичний об'єкт, починаючи від поняття числа, є математичною моделлю. При побудові математичної моделі, об'єкту, що вивчається, або явища виділяють ті його особливості, риси і деталі, які з одного боку містять більш менш повну інформацію про об'єкт, а з іншою допускають математичну формалізацію. Математична формалізація означає, що особливостям і деталям об'єкту можна поставити у відповідність відповідні адекватні математичні поняття: числа, функції, матриці і так далі. Тоді зв'язки і стосунки, виявлені і передбачувані в об'єкті, що вивчається, між окремими його деталями і складовими частинами можна записати за допомогою математичних стосунків: рівності, нерівностей, рівнянь. В результаті виходить математичний опис процесу, що вивчається, або явище, тобто його математична модель.

Вивчення математичної моделі завжди пов'язаної з деякими правилами дії над об'єктами, що вивчаються. Ці правила відбивають зв'язки між причинами і наслідками.

Побудова математичної моделі – це центральний етап дослідження або проектування будь-якої системи. Від якості моделі залежить увесь наступний аналіз об'єкту. Побудова моделі – це процедура не формальна. Сильно залежить від дослідника, його досвіду і смаку, завжди спирається на певний дос-

відчений матеріал. Модель має бути досить точною, адекватною і має бути зручна для використання.

Математичні моделі можуть бути детерменированными і стохастичними. Детерменированные моделі – це моделі, в яких встановлена взаємно-однозначна відповідність між змінними що описують об'єкт або явища.

Такий підхід заснований на знанні механізму функціонування об'єктів. Часто модельований об'єкт складний і розшифровка його механізму може виявитися дуже трудомісткою і довгою в часі. В цьому випадку поступають таким чином: на оригіналі проводять експерименти, обробляють отримані результати і, не вникаючи в механізм і теорію модельованого об'єкту за допомогою методів математичної статистики і теорії вірогідності, встановлюють зв'язки між змінними, що описують об'єкт. В цьому випадку отримують статистическую модель. Встатистической моделі зв'язок між змінними носить випадковий характер, іноді це буває принципово. Дія величезної кількості чинників, їх поєднання приводить до випадкового набору змінних об'єкт, що описують, або явище. По характеру режимів модель бувають статистичними і динамічними.

Статистична модель включає опис зв'язків між основними змінними модельованого об'єкту в сталому режимі без урахування зміни параметрів в часі.

У динамічній моделі описуються зв'язки між основними змінними модельованого об'єкту при переході від одного режиму до іншого.

Моделі бувають дискретними і безперервними, а також змішаного типу. У безперервних змінні набувають значень з деякого проміжку, в дискретних змінні набувають ізольованих значень.

Лінійні моделі – усі функції і стосунки, що описують модель лінійно залежать від змінних і не лінійні інакше.

Вимоги, що пред'являються до моделей:

- універсальність – характеризує повноту відображення моделлю властивостей реального об'єкту, що вивчаються;

- адекватність – здатність відбивати потрібні властивості об'єкту з погрішністю не вище за задану;
- точність – оцінюється мірою збігу значень характеристик реального об'єкту і значення цих характеристик отриманих за допомогою моделей;
- економічність – визначається витратами ресурсів ЕОМ пам'яті і часу на її реалізацію і експлуатацію.

Основні етапи моделювання:

1) Постановка завдання. Визначення мети аналізу і шляху її досягнення і вироблення загального підходу до досліджуваної проблеми. На цьому етапі потрібно глибоке розуміння істоти поставленого завдання. Іноді, правильно поставити завдання не менш складно чим її вирішити. Постановка – процес не формальний, загальних правил немає.

2) Вивчення теоретичних основ і збір інформації про об'єкт оригіналу. На цьому етапі підбирається або розробляється відповідна теорія. Якщо її немає, встановлюються причини – слідчі зв'язки між змінними що описують об'єкт. Визначаються вхідні і вихідні дані, приймаються спрощуючі припущення.

3) Формалізація. Полягає у виборі системи умовних позначень і з їх допомогою записувати стосунки між складовими об'єкту у вигляді математичних виразів. Встановлюється клас завдань, до яких може бути віднесена отримана математична модель об'єкту. Значення деяких параметрів на цьому етапі ще можуть бути не конкретизовані.

4) Вибір методу рішення. На цьому етапі встановлюються остаточні параметри моделей з урахуванням умови функціонування об'єкту. Для отриманого математичного завдання вибирається який-небудь метод рішення або розробляється спеціальний метод. При виборі методу враховуються знання користувача, його переваги, а також переваги розробника.

5) Реалізація моделі. Розробивши алгоритм, пишеться програма, яка відлагоджується, тестується і виходить рішення потрібної задачі.

б) Аналіз отриманої інформації. Зіставляється отримане і передбачуване рішення, проводиться контроль погрішності моделювання.

7) Перевірка адекватності реальному об'єкту. Результати, отримані по моделі зіставляються або з наявною про об'єкт інформацією або проводиться експеримент і його результати зіставляються з розрахунковими.

Процес моделювання є ітеративним. У разі незадовільних результатів етапів 6 або 7 здійснюється повернення до одного з ранніх етапів, який міг привести до розробки невдалої моделі. Цей етап і усі наступні уточнюються і таке уточнення моделі відбувається до тих пір, поки не будуть отримані прийнятні результати.

Моделювання можна розглядати як заміщення досліджуваного об'єкту (оригіналу) його умовним чином, описом або іншим об'єктом, що іменується моделлю і що забезпечує близьку до оригіналу поведінку у рамках деяких допущень і прийнятних погрішностей. Моделювання зазвичай виконується з метою пізнання властивостей оригіналу шляхом дослідження його моделі, а не самого об'єкту. Зрозуміло, моделювання виправдане у тому разі коли воно простіше за створення самого оригіналу або коли останній з якихось причин краще взагалі не створювати.

Під моделлю розуміється фізичний або абстрактний об'єкт, властивості якого в певному значенні схожі з властивостями досліджуваного об'єкту. При цьому вимоги до моделі визначаються вирішуваною задачею і наявними засобами. Існує ряд загальних вимог до моделей:

- адекватність – досить точне відображення властивостей об'єкту;
- повноти – надання одержувачеві усієї необхідної інформації про об'єкт;
- гнучкість – можливість відтворення різних ситуацій в усьому діапазоні зміни умов і параметрів;
- трудомісткість розробки має бути прийнятною для наявного часу і програмних засобів.

На практиці застосовують різні методи моделювання. Залежно від способу реалізації, усі моделі можна розділити на два великі класи: фізичні і математичні.

Математичне моделювання прийнято розглядати як засіб дослідження процесів або явищ за допомогою їх математичних моделей.

Під фізичним моделюванням розуміється дослідження об'єктів і явищ на фізичних моделях, коли процес, що вивчається, відтворюють із збереженням його фізичної природи або використовують інше фізичне явище, аналогічне тому, що вивчається. При цьому фізичні моделі припускають, як правило, реальне втілення тих фізичних властивостей оригіналу, які є істотними в конкретній ситуації. У зв'язку з цим фізичне моделювання називають також макетуванням.

Напівнатурне моделювання є дослідженням керованих систем на моделюючих комплексах з включенням до складу моделі реальної апаратури. При дослідженні моделей з включенням реальної апаратури використовується поняття динамічного моделювання, при дослідженні складних систем і явищ – еволюційного, імітаційного і кібернетичного моделювання.

Очевидно, дійсна користь від моделювання може бути отримана тільки при дотриманні двох умов:

- модель забезпечує коректне (адекватне) відображення властивостей оригіналу, істотних з точки зору досліджуваної операції;
- модель дозволяє усунути перелічені вище проблеми, властиві проведенню досліджень на реальних об'єктах.

Імітаційна модель – це комп'ютерна програма, яка описує структуру і відтворює поведінку реальної системи в часі. Імітаційна модель дозволяє отримувати детальну статистику про різні аспекти функціонування системи в залежності від вхідних даних [1]¹⁾.

¹⁾ [1] Хемди А. Таха Имитационное моделирование. Введение в исследование операций. М.: Вильямс, 2007. 917 с.

1.2 Приклади використання імітаційних моделей

Застосування методу імітаційного моделювання можна продемонструвати на прикладі роботи відділення банку по обслуговуванню фізичних осіб. Припустимо, що необхідно визначити мінімальну кількість обслуговуючого персоналу, яке забезпечує необхідну якість сервісу.

Критерій якості сервісу задамо правилом: середній розмір черги клієнтів не повинен перевищувати N чоловік. Очевидно, що для вирішення поставленого завдання необхідно мати достатні знання про систему: які клієнти відвідують банк, яка кількість клієнтів приходить протягом робочого дня, а також скільки часу займає обслуговування одного клієнта.

Хоча дана задача і може здатися спеціалізованою, схожі проблеми виникають у багатьох областях, де задіяні людські і технічні ресурси. Оплата часу роботи кваліфікованого працівника і часу використання складної техніки становить чималу частку витрат компаній. Визначення оптимального графіка використання ресурсів, що дозволяє системі ефективно виконувати поставлені завдання, дозволяє знизити витрати, а значить збільшити прибутковість.

На першому етапі рішення задачі створюється модель, яка відповідає структурі і бізнес-процесів відділення банку. В ході розробки моделі враховуються тільки ті деталі, які істотно впливають на досліджувані аспекти роботи системи. Наприклад, наявність відділення обслуговування юридичних осіб або кредитного відділу не впливає на обслуговування фізичних осіб, оскільки вони фізично і функціонально відокремлені від останнього. Схематично таку модель можна представити у вигляді послідовності дій, як це показано на рис. 1.1.

На другому етапі на вхід моделі подаються вихідні дані: інтенсивність приходу клієнтів, середній час обслуговування клієнтів, кількість доступного персоналу. На підставі цих даних модель імітує, або відтворює, роботу банку протягом заданого проміжку часу, наприклад, робочого дня.

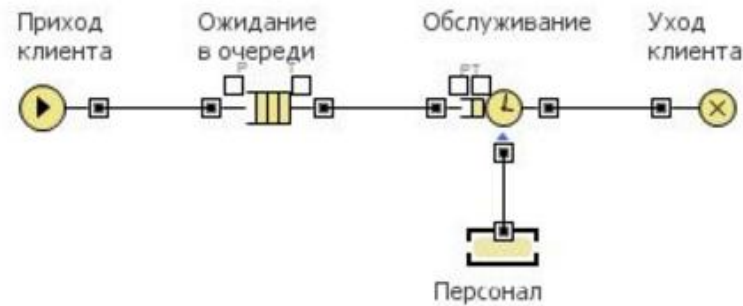


Рисунок 1.1 – Модель обслуговування клієнтів у відділенні банку

Наступний етап полягає в аналізі статистики, зібраної і представленою моделлю. Якщо середній розмір черги клієнтів перевищує обрану межу в N чоловік, то кількість доступного персоналу слід збільшити і виконати новий експеримент.

В результаті проведення серії експериментів над моделлю користувач може визначити оптимальну кількість персоналу. Процес підбору параметрів може бути здійснений також і за допомогою вбудованого оптимізатора, який в автоматичному режимі перевіряє різні поєднання і знаходить краще рішення.

1.3 Переваги імітаційного моделювання

Застосування імітаційних моделей дає безліч переваг в порівнянні з виконанням експериментів над реальною системою і використанням інших методів.

1) Вартість. Припустимо, компанія звільнила частину співробітників, що в подальшому призвело до зниження якості обслуговування і втрати частини клієнтів. Прийняти обґрунтоване рішення допомогла б імітаційна модель, витрати на застосування якої складаються лише з вартості програмного забезпечення та вартості консалтингових послуг.

2) Час. У реальності оцінити ефективність, наприклад, нової мережі розповсюдження продукції або зміненої структури складу можна лише через

місяці або навіть роки. Імітаційна модель дозволяє визначити оптимальність таких змін за лічені хвилини, необхідні для проведення експерименту.

3) Повторюваність. Сучасне життя вимагає від організацій швидкої реакції на зміну ситуації на ринку. Наприклад, прогноз обсягів попиту продукції повинен бути складений в термін, і його зміни критичні. За допомогою імітаційної моделі можна провести необмежену кількість експериментів з різними параметрами, щоб визначити найкращий варіант.

4) Точність. Традиційні розрахункові математичні методи вимагають застосування високого ступеня абстракції і не враховують важливі деталі. Імітаційне моделювання дозволяє описати структуру системи і її процеси в природному вигляді, не вдаючись до використання формул і строгих математичних залежностей.

5) Наочність. Імітаційна модель має можливості візуалізації процесу роботи системи в часі, схематичного завдання її структури і видачі результатів в графічному вигляді. Це дозволяє наочно уявити отримане рішення і довести закладені в нього ідеї до клієнта і колег.

6) Універсальність. Імітаційне моделювання дозволяє вирішувати завдання з будь-яких областей: виробництва, логістики, фінансів, охорони здоров'я і багатьох інших. У кожному разі модель імітує, відтворює, реальне життя і дозволяє проводити широкий набір експериментів без впливу на реальні об'єкти.

2 ОСНОВНІ ПІДХОДИ В ІМІТАЦІЙНОМУ МОДЕЛЮВАННІ

У імітаційному моделюванні виділяють три основних підходи:

- системна динаміка;
- дискретно-подієве моделювання (процесно-орієнтоване);
- агентне моделювання.



Рисунок 2.1 – Підходи в імітаційному моделюванні на шкалі рівня абстракції

Такий поділ можна застосувати для опису конкретного пакета моделювання, так як використання будь-якого з описаних підходів тягне за собою певний спосіб побудови, як пакетів моделювання, так і моделей обраного типу. Більш того, для більшості завдань визначено найкращі підходи до їх моделювання, з чого випливає, що такий розподіл розбиває безліч завдань на практично непересічні області.

Наведена нижче класифікація підходів до імітаційного моделювання була запропонована і описана в [2]¹⁾.

2.1 Системна динаміка

Цей підхід був розроблений і запропонований Джейм Форрестер в кінці 1950-х років як дослідження інформаційних зворотних зв'язків в промисловій діяльності з метою показати як організаційна структура, посилення (в політиках) і затримки (в прийнятті рішень і діях) взаємодіють, впливаючи на ус-

¹⁾ [2] Борщев А.В. Практическое агентное моделирование и его место в арсенале аналитика. URL: <http://www.gpss.ru/immod05/p/borshev/> (дата звернення 13.05.2019)

підприємства. Додатки системної динаміки включають також соціальні, урбаністичні, екологічні системи. Процеси, що відбуваються в реальному світі, в системній динаміці представляються в термінах накопичувачів (stocks), потоків між цими накопичувачами (flows), та інформації, яка визначає величину цих потоків. Системна динаміка абстрагується від окремих об'єктів і подій і передбачає "агрегатний" погляд на процеси, концентруючись на політиках, що цими процесами керують. Структура і поведінка системи представляється як безліч взаємодіючих позитивних і негативних зворотних зв'язків і затримок.

Математично, системно-динамічна модель – це система диференціальних рівнянь. Звідси випливають важливі властивості моделей, що відносяться до системної динаміки:

- оскільки модель оперує тільки кількостями, агрегатами, об'єкти, що знаходяться в одному накопичувачі, невиразні, позбавлені індивідуальності;
- аналітику пропонується міркувати в термінах глобальних структурних залежностей і тому він повинен мати відповідні вихідні дані.

Системна динаміка – це підхід імітаційного моделювання, що своїми методами і інструментами дозволяє зрозуміти структуру і динаміку складних систем. Також системна динаміка – це метод моделювання, що використовується для створення точних комп'ютерних моделей складних систем для подальшого використання з метою проектування більш ефективної організації і політики взаємин з даною системою. Разом, ці інструменти дозволяють нам створювати мікросвіти-симулятори, де простір і час можуть бути стиснуті і уповільнені так, щоб ми могли вивчити наслідки наших рішень, швидко освоїти методи і зрозуміти структуру складних систем, спроектувати тактики і стратегії для більшого успіху [3]¹⁾.

¹⁾ [3] Строгалева В. П., Толкачева И. О. Имитационное моделирование. МГТУ им. Баумана, 2008. 697 с.

Системна динаміка головним чином використовується в довгострокових, стратегічних моделях і приймає високий рівень абстракції. Люди, продукти, події та інші дискретні елементи представлені в моделях системної динаміки не як окремі елементи, а як система в цілому. Якщо ж окремі елементи моделі важливі, то для повної або часткової обробки моделі краще скористатися агентним або дискретно-подієвим моделюванням.

2.2 Дискретно-подієве моделювання

В основі даного підходу лежить концепція заявок (транзактов, entities), ресурсів і потокових діаграм (flowcharts), що визначають потоки заявок і використання ресурсів. Цей підхід був запропонований Джеффри Гордоном, який в 1960-х придумав і розвинув GPSS (General Purpose Simulation System) і реалізував її, працюючи в IBM [4,5]¹⁾. Заявки (транзакти в GPSS) – це пасивні об'єкти, що представляють людей, деталі, документи, завдання, повідомлення і т.п. Вони подорожують через структурну діаграму, стоячи в чергах, обробляючи, захоплюючи і звільняючи ресурси, розділяючись, з'єднуючись і т.п. До 2000 р. на світовому ринку пропонувалося більше 60 програмних продуктів дискретно-подієвого моделювання виробничих і економічних систем, таких як ARENA, AUTOMOD, Extend, MicroSaint, Enterprise Dynamics, eM-Plant .

Щоб аналізувати процеси, що протікають в світі, іноді зручно розглядати їх як послідовність окремих важливих моментів – подій. Підхід до побудови імітаційних моделей, що пропонує уявити реальні дії такими подіями і називається "дискретно-подієвим" моделюванням (discrete event modeling) Discrete Event Modeling

Ось деякі приклади подій: покупець увійшов в магазин, на складі закінчили розвантажувати вантажівку, конвеєр зупинився, у виробництво за-

¹⁾ [4] Шрайбер Т. Дж. Моделирование на GPSS. М.: Машиностроение, 1980. 592 с.

[5] Боев В. Д. Моделирование систем. Инструментальные средства GPSS WORLD. Учеб. пособие. СПб.: БХВ-Петербург, 2004. 368 с.

пущено новий продукт, рівень запасів досяг певного порогу та інш. У дискретно-подієвому моделюванні руху поїзда з точки А в точку Б буде представлено двома подіями: відправлення і прибуття, а сам рух стає "затримкою" (інтервалом часу) між ними.

Термін "дискретно-подієвого моделювання", зазвичай використовується для позначення "процесного" моделювання, де динаміка системи представляється як послідовність операцій над якимись сутностями, що представляють клієнтів, документи, транспортні засоби тощо. Ці сутності пасивні, вони самі не контролюють свою динаміку, але можуть володіти певними атрибутами, що впливають на процес їх обробки або накопичують статистику. Процесне моделювання використовується на середньому або низькому рівні абстракції: кожен об'єкт моделюється індивідуально, як окрема сутність, але безліч деталей "фізичного рівня" (геометрія, прискорення / уповільнення) опускається. Такий підхід широко використовується в моделюванні бізнес-процесів, виробництва, логістики, охорони здоров'я тощо [6,7]¹⁾.

2.3 Агентне моделювання

У літературі можна знайти безліч різних визначень агентного моделювання. З точки зору практичного застосування агентне моделювання можна визначити як метод імітаційного моделювання, який досліджує поведінку децентралізованих агентів і те, як це поведінка визначає поведінку всієї системи в цілому. При розробці агентної моделі, інженер вводить параметри агентів (це можуть бути люди, компанії, активи, проекти, транспортні засоби, міста, тварини тощо), визначає їх поведінку, поміщає їх в якесь навколишнє середовище, встановлює можливі зв'язки, після чого запускає моделювання. Ін-

¹⁾ [6] Боев В. Д, Сыпченко Р. П. Компьютерное моделирование. Элементы теории и практики. Учеб. пособие. СПб.: Военная академия связи, 2009. 432 с.

[7] Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб.: БХВ-Петербург, 2005. 312 с.

дивідуальна поведінка кожного агента утворює глобальну поведінку, що моделюється.

Традиційні підходи імітаційного моделювання розглядають службовців компанії, проекти, продукти, клієнтів, партнерів як середнє арифметичне або як пасивні заявки / ресурси в процесі. Наприклад, моделі системної динаміки повні припущень, таких як "у нас є 120 службовців, вони можуть проектувати приблизно 20 нових продуктів на рік", або "у нас тобто 1200 вантажівок, вони можуть перевезти певну кількість вантажу в місяць, і 5% з них списуються щороку і замінюються новими ". У процесному моделюванні (також відомому як дискретно-подієве моделювання) організація розглядається як різні процеси, такі як: " клієнт дзвонить в телефонний інформаційний центр, дзвінок обробляється оператором А, який витрачає, в середньому, 2 хвилини на виклик, після чого 20% запитів повинні бути переадресовані ... ". Ці методи перевершують " аналітичне моделювання " в можливості розглядати динаміку підприємства, нелінійності, але вони ігнорують той факт, що всі ці люди, проекти, продукти, обладнання та активи є різними – вони мають власну історію, наміри, бажання, властивості, а також складні відносини. Наприклад, люди можуть бути з різними кар'єрами і доходами, вони можуть мати різну продуктивність праці; проекти взаємодіють і конкурують, можуть залежати один від іншого; у літаків є індивідуальні графіки технічного обслуговування, при недотриманні яких машина може вийти з ладу; споживачі можуть консультуватися з членами своєї сім'ї, перш ніж прийняти рішення про покупку. Агентне моделювання не володіє такими обмеженнями, оскільки воно передбачає зосередження безпосередньо на окремих об'єктах, їх поведінці і комунікації. Агентна модель – це ряд взаємодіючих активних об'єктів, які відображають об'єкти і відносини в реальному світі. Таким чином, агентне моделювання робить крок вперед у розумінні та управлінні сукупністю складних соціальних і бізнес процесів.

Гарний приклад використання агентного моделювання – споживчий ринок. У дуже динамічному, конкурентному і складному середовищі ринку

вибір покупця часто залежить від індивідуальних особливостей, вродженої активності споживача, мережі контактів, а також зовнішніх впливів, які найкраще описуються за допомогою агентного моделювання.

Інший стандартний приклад – це епідеміологія. Тут агенти це люди, які можуть бути імунними, носіями інфекції, які перехворіли або сприйнятливими до хвороби. Агентне моделювання допоможе спроектувати в світ моделей соціальні мережі, різноманітні контакти між людьми і в підсумку отримати об'єктивні прогнози поширення інфекції.

Однак, не слід думати, що агентне моделювання можна застосувати тільки для вирішення завдань комунікативного характеру. Завдання, пов'язані з логістикою, виробництвом, ланцюгами поставок або бізнес-процесами, також вирішуються за допомогою агентного моделювання. Наприклад, поведінка складної машини може бути ефективно змодельовано окремим об'єктом (агентом) з картами станів, що описують її систему таймерів, внутрішніх станів, різного роду реакції в різних ситуаціях тощо. Подібна модель може бути необхідна для відтворення технологічних процесів на виробництві.

Учасники ланцюжка поставок (компанії-виробники, оптові торговці, роздрібні продавці) можуть бути представлені як агенти з індивідуальними цілями і правилами. Агенти можуть також бути проектами або продуктами в межах однієї компанії, при цьому мати власну динаміку і внутрішніми станами, конкурувати за ресурси компанії.

3 АГЕНТНЕ МОДЕЛЮВАННЯ В ANYLOGIC

Існує безліч визначень поняття агента. Спільним у всіх цих визначеннях є те, що агент – це деяка сутність, яка володіє активністю, автономною поведінкою, може приймати рішення відповідно до деяких наборів правил, може взаємодіяти з оточенням і іншими агентами, а також може змінюватися (еволюціонувати). Багатоагентні (або просто агентні) моделі використовуються для дослідження децентралізованих систем, динаміка функціонування

яких визначається не глобальними правилами і законами, а навпаки, ці глобальні правила і закони є результатом індивідуальної активності членів групи. Мета агентних моделей – отримати уявлення про ці глобальні правила, загалом про поведінку системи, виходячи з припущень про індивідуальну, приватну поведінку її окремих активних об'єктів і взаємодію цих об'єктів в системі. Зростання продуктивності комп'ютерів і досягнення в інформаційних технологіях, використані в AnyLogic, зробили можливим реалізацію агентних моделей, що містять десятки і навіть сотні тисяч активних агентів [8]¹⁾.

Моделювання агентів і багатоагентних систем не представляє складності в AnyLogic ні в концептуальному, ні в технічному аспекті: все розглянуті раніше властивості агентів легко реалізуються в цій системі. Основною концепцією AnyLogic є те, що модель складається з активних об'єктів, які мають свої правила поведінки і взаємодіють через певні інтерфейси. Тому агентний підхід до побудови моделей є в AnyLogic абсолютно природним: в цьому середовищі можна швидко створювати моделі з агентами, взаємодіючими як один з одним, так і з середовищем.

При створенні агентної моделі логіка поведінки агентів і їх взаємодія не завжди можуть бути виражені чисто графічними засобами, тут часто доводиться використовувати програмний код. Це не є особливістю лише AnyLogic: наприклад, пакети Swarm і RePast для розробки агентних моделей використовують тільки програмний код на мові Java [9,10]²⁾.

На відміну від них, AnyLogic надає для розробки агентних моделей всю міць візуальної графічної розробки: карти станів (statechart), події, таймери, синхронне і асинхронне планування подій, бібліотеки раніше визначених активних об'єктів – все це виявляється зручним і природним при розробці аген-

¹⁾ [8] Рьжиков Ю.И. Имитационное моделирование. Теория и технология. СПб.: КОРОНА принт, 2004. 384 с.

²⁾ [9] XJ Technologies. Офіційний сайт. URL: <http://www.xjtek.com> (дата звернення 17.05.2019)

[10] Java For AnyLogic Users. URL: http://www.xjtek.com/files/book/Java_for_AnyLogic_users.pdf (дата звернення 17.05.2019)

тних моделей. Однак розробник зазвичай повинен включати в агентну модель фрагменти коду на мові Java.

3.1 Агенти в AnyLogic

Агента можна реалізувати за допомогою базового об'єкта AnyLogic – активного об'єкта. У моделі на AnyLogic можна створювати класи активних об'єктів і далі використовувати в моделі будь-яке число примірників цих класів. Активний об'єкт має параметри, які можна змінювати ззовні, змінні, які можна вважати пам'яттю агента, а також поведінку.

Параметри можуть вказувати стать агента, дату народження тощо. Перемінними можна висловити, наприклад, вік агента, його координати в просторі, соціальні властивості. Карти станів можуть висловлювати поведінку: стан агента і зміна станів під впливом подій і умов. Крім того, агент може мати інтерфейс для взаємодії з оточенням (рис.3.1).

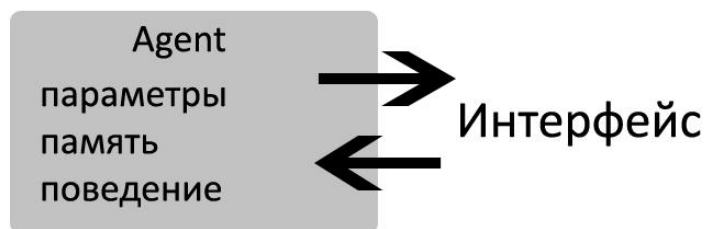


Рисунок 3.1 – Взаємодія агента із зовнішнім світом

3.2 Поведінка агентів

Поведінка може виражати, наприклад, правила дій агента або закони переміщення агента в просторі, зміни його соціального статусу, переходи в різні вікові або соціальні групи, зміни освіти і доходу, сімейного стану тощо. Для подання дискретної поведінки природно використовувати карти станів. Різні ролі агента можуть виражатися різними картами станів.

На рис. 3.2 представлена карта станів, що виражає поведінку агента, в розглянутій нижче моделі. Дана карта станів демонструє можливості роботи AnyLogic з картами станів: а саме вони можуть містити наступні елементи.

- 1) Простий стан. Управління завжди знаходиться в одному з простих станів. Користувач також може визначити дії на вході і виході зі стану.
- 2) Перехід. Спрацьовує за зовнішніми та внутрішніми подіями. Можливо визначення дій на переході, а також використання умов.
- 3) Складений стан. Об'єднання декількох простих станів в одне.
- 4) Історія (псевдостан). Зберігає останній стан в групі.
- 5) Розгалуження (псевдостан). Умовне розгалуження переходів.

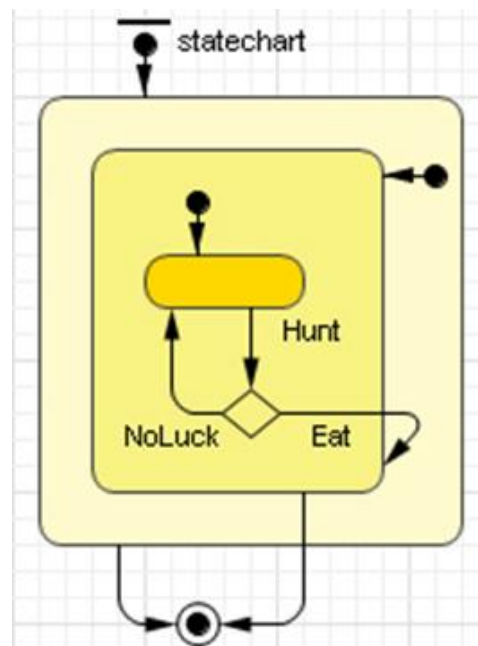


Рисунок 3.2 – Приклад карти станів

Поряд з картами станів, для специфікації поведінки агентів можуть використовуватися таймери, функції – весь арсенал засобів, доступний в AnyLogic, в будь-яких їх комбінаціях.

3.3 Інтерфейси агентів

Всі засоби взаємодії об'єктів, що доступні в AnyLogic, можуть використовуватися для побудови агентних моделей. Рис. 3.3 показує ці можливості.

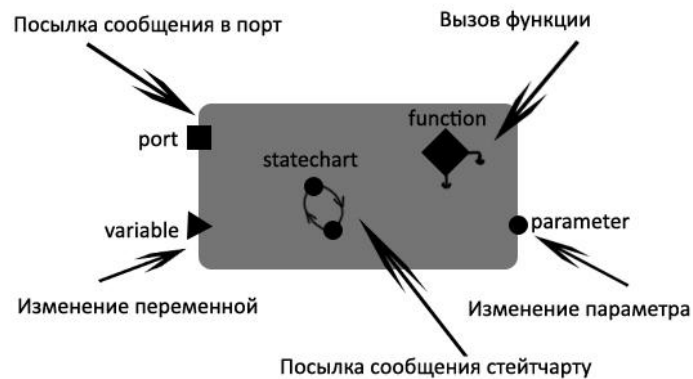


Рисунок 3.3 – Інтерфейси для взаємодії агента з середовищем

Якщо агент визначений в деякому зовнішньому середовищі, для нього доступні наступні інтерфейси для взаємодії з цим навколишнім середовищем:

- Здійснення повідомлень в порт агента;
- Зміна інтерфейсної змінної агента;
- Здійснення повідомлення в стейтчарт агента;
- Зміна параметра агента (зазвичай при ініціалізації);
- Виклик функції, визначеної в агента.

3.4 Архітектура агентних моделей

Архітектура будь-яких виконуваних моделей в AnyLogic має вигляд дерева. Для агентних моделей ця архітектура є цілком природною. Наприклад, розглянемо структуру моделі Air Defense System. Робота моделі полягає в захисті деяких цінностей (assets) від ворожих літаків (aircrafts) за допомогою

радарів і ракет (radar). Природним методом створення такої моделі є уявлення кожної сутності у вигляді окремого об'єкта, як і показано на рис.3.4.

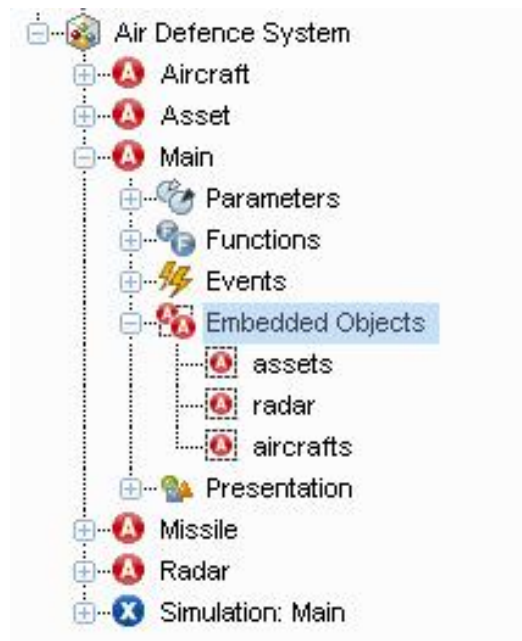


Рисунок 3.4 – Ієрархія об'єктів

На рис. 3.4 видно, що об'єкт Main містить в собі екземпляри об'єктів: assets, radar, aircrafts. Також середовище AnyLogic дозволяє розмножувати (репліцирувати) екземпляри агентів. Так в даній моделі необхідно деяку кількість радарів, літаків і об'єктів для захисту. Для цього потрібно помістити екземпляри відповідних класів в клас Main і в параметрах кожного, в поле Replication задати бажану їх кількість. При цьому, ці екземпляри приймуть вигляд в графічному редакторі, як показано на рис.3.5.



Рисунок 3.5 – Реплікація об'єктів

3.5 Взаємодія агентів з середовищем

Агенти зазвичай функціонують в деякому середовищі, і взаємодія з середовищем є важливим завданням агента. Роль середовища для агентів грає або активний об'єкт, в який вкладені агенти, або спеціально призначений для цього клас Environment. Його застосування розглянемо на прикладі моделі поширення інфекції. Об'єкт Main даної моделі містить, окрім іншого, реплікований об'єкт people і середу для взаємодії агентів Environment (рис.3.6).

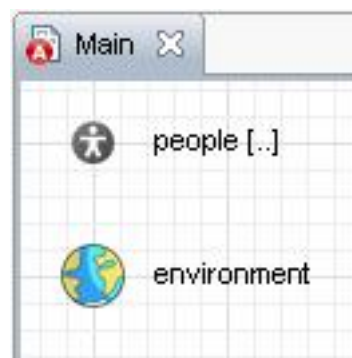


Рисунок 3.6 – Середовище Environment

Навколишнє середовище Environment надає зручні засоби взаємодії між агентами. Так, наприклад, в даній моделі інфікований агент може заразити тільки будь-якого зі своїх сусідів. Замість того, що б вводити деяку метрику і з її урахуванням реалізовувати функцію вибору сусіда, користувач може викликати функцію

```
send ("Infection", RANDOM_NEIGHBOR);
```

де RANDOM_NEIGHBOR – покажчик на будь-якого сусіда даного агента, наданий середовищем Environment.

Починаючи з версії AnyLogic 7.1 середовищем взаємодії агентів можуть бути ГІС карти. Це дуже важливо, наприклад, для моделювання ланцюгів постачання товарів, рішення завдань тип «виробництво-склад».

4 МОДЕЛЮВАННЯ ПРОЦЕСУ ПОСТАЧАННЯ ТОВАРІВ

4.1 Постановка завдання

Підприємство JB-PLAST, що знаходиться за адресою м.Запоріжжя, вул. Східна, 1 виготовляє товари і комплектуючих для виробництва сучасного взуття. Замовниками JB-PLAST є взуттєві фабрики і цехи, що працюють в Україні. Основні регіональні представництва підприємства:

- 1) м. Харків, пр. Гагаріна, 84/86
- 2) м. Житомир, вул. Героїв пожежних, 70-а
- 3) м.Бровари, вул. Металргів, 4
- 4) м. Кривий Ріг, пл. Жовтнева, 1
- 5) м. Львів, вул. Залізнична, 7
- 6) м. Миколаїв, вул. Комсомольська, 6
- 7) м. Полтава, вул. Степового фронту, 48
- 8) м. Дніпропетровськ, вул. Бабушкіна, 54
- 9) м. Одеса, вул. Онезька, 3

Даним представництвам поставляються товари і комплектуючі два рази на тиждень. Оповіщення від регіональних представництв приходять на підприємство у формі замовлення. Після чого на завантаження вантажного автомобіля потрібно від двох до трьох годин. Стільки ж часу потрібно і на розвантаження вантажівки в місці прибуття. Після отримання комплектуючих представництво сповіщає про це підприємство повідомленням, далі вантажівка відправляється назад на підприємство.

Потрібно змоделювати процес доставки комплектуючих для оцінки оптимальної кількості транспортних засобів, враховуючи, що всього на підприємстві є п'ять вантажівок.

Результат моделювання процесу вантажоперевезень у середовищі AnyLogic надасть можливість компанії ефективніше використовувати свій транспортний парк та заощадити гроші на покупку чи оренду вантажівок.

4.2 Декомпозиція задачі на етапи для побудови моделі в AnyLogic

Для вирішення даного завдання необхідно послідовно виконати такі дії:

- ввести інформацію про розташування і маршрути до всіх регіональних представництв підприємства JB-PLAST;
- описати процес оформлення замовлення нових комплектуючих, вважаючи, що кожне представництво відправляє запит однакової форми;
- описати логіку обробки заявки підприємством, де необхідно врахувати: отримання заявки, час на завантаження транспорту, відправку до клієнта, розвантаження, сповіщення про доставку і повернення вантажівки на підприємство;
- провести оптимізацію, з метою встановлення необхідної кількості вантажівок для підприємства, щоб завантаженість при доставці комплектуючих становила не більше 85%.

4.3 Реалізація моделі

4.3.1 Формування вихідних даних

Для рішення задачі необхідно виділити вихідні дані. В даному випадку, вихідними даними є координати розташування представництв, перерахованих в тексті завдання, координати підприємства, що виробляє і доставляє комплектуючі для них, і шляхи сполучення між ними, за якими і здійснюється доставка. Так як мова в задачі йде про реально існуючі об'єкти, зручно використовувати компонент «ГІС Карта» з палітри «Розмітка простору». За замовчуванням середня точка на карті і масштаб будуть обрані, як на рис.4.1.

На вкладці «Тайли» компонента «ГІС Карта» можна вибрати постачальника карти, що змінить зовнішній вигляд об'єкта; на вкладці «Маршрути» задаються параметри маршрутів (по аналогії з автомобільними навігаторами): можна вибрати джерело маршрутів (з сервера OSM або з файлу PBF),

сервер з якого маршрути завантажуються, критерій вибору маршруту (найшвидший, найкоротший), мережу доріг (автомобільна, велосипедна, пішохідна). Встановимо властивості об'єкта «ГІС Карта» згідно рис. 4.2.



Рисунок 4.1 – Відображення компонента «ГІС Карта» за замовчуванням

map - ГИС Карта

Имя: Отображается на верхнем уровне

Блокировать

Видимость: да

Тайлы

Отображать тайлы

Поставщик тайлов:

Маршруты

Маршруты: Загружаются с сервера OSM
 Считываются из файла PBF
 Строятся по прямой

Сервер маршрутов:

Выбирается маршрут:

Сеть дорог:

Если маршрут не найден: Строить маршрут по прямой
 Отображать сообщение об ошибке

Рисунок 4.2 – Властивості компонента «ГІС Карта»

Далі необхідно перейти в режим редагування карти для вибору конкретної області відображення (в контексті даного завдання – території України). Це можна зробити за допомогою миші.

Вибравши область відображення карти, необхідно перейти до її розмітці, скориставшись в режимі редагування вбудованою функцією пошуку.

Додати розташування географічних об'єктів на карті можна за допомогою вбудованої бази даних, постачальником даних для неї може бути файл Excel. В базі даних можуть зберігатися або координати об'єктів, або географічні їх назви. На рис. 4.3 наведений вигляд таблиці «Представительства» з бази даних моделі. При додаванні нових об'єктів модель буде автоматично включати їх до маршрутів постачання комплектуючих і відображати на карті.

	city
1	Харків
2	Житомир
3	Кривий Ріг
4	Львів
5	Миколаїв
6	Полтава
7	Дніпропетровськ
8	Одеса
*	

Рисунок 4.3 – Таблиця «Представительства»

На рис. 4.4 показана карта з мітками всіх об'єктів з тексту завдання. Для зручності розміщення агентів в зазначених точках, об'єднаємо їх в колекцію. Створити колекцію всіх відміток регіональних представництв. У властивостях колекції задамо ім'я `officesLocation`. Як зрозуміло з назви в цій колекції будуть зберігатися координати представництв (рис.4.5).

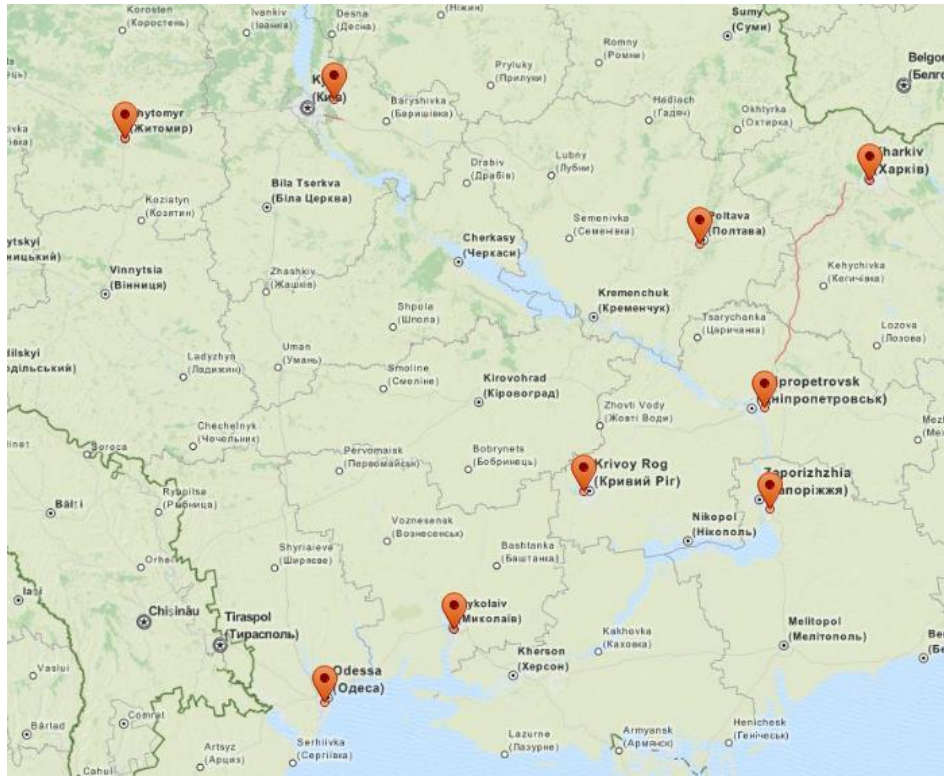


Рисунок 4.4 – Карта з відмітками всіх об'єктів з тексту завдання

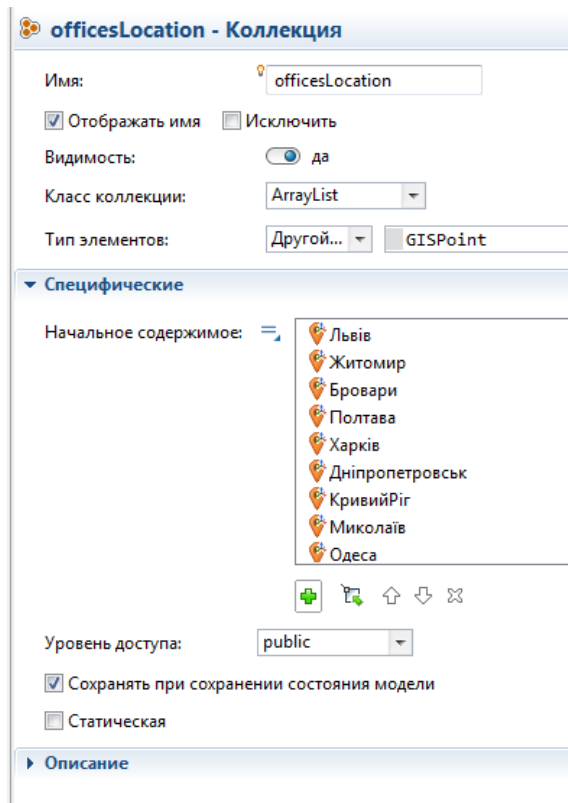


Рисунок 4.5 – Властивості колекції officesLocation

Для зв'язку представництв з їх координатами, в пункті властивостей, що відповідає початковому розташуванню оберемо: у вузлі і додамо вираз: `officesLocation.get(index)`. Ця функція повертає адреси (індекси) елементів в колекції `officesLocation` (тих самих міток на карті). Властивості популяції представництва представлені на рис.4.6.

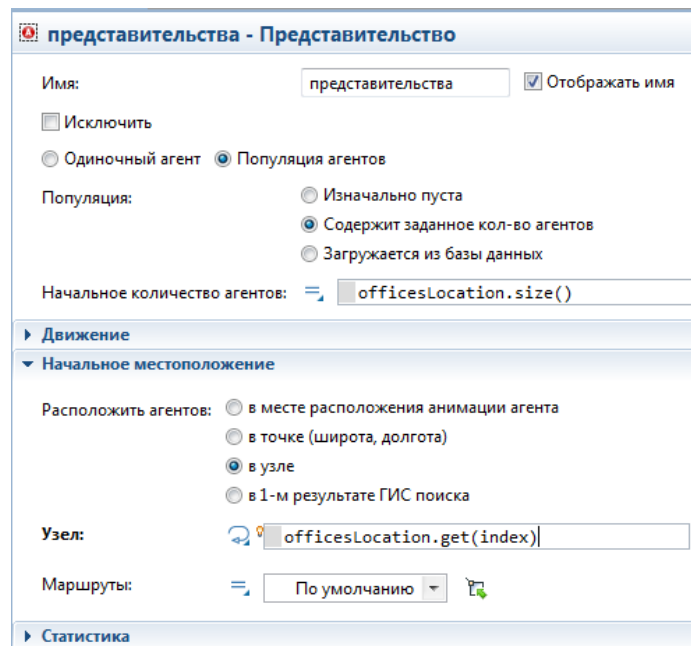


Рисунок 4.6 – Властивості популяції агента «Представительство»

У ці координати необхідно помістити представництва. Для цього створимо популяцію агентів, які і будуть моделями представництв.

Далі необхідно пов'язати створену популяцію агента «Представительство» з координатами представництв і їх кількістю, тобто пов'язати з колекцією `officesLocation`. Для цього у властивості популяції представительства у графі з початкової кількістю агентів додамо наступний вираз: `officesLocation.size()`. Ця функція повертає кількість елементів в колекції `officesLocation` (в даному випадку 9).

Далі створимо тип агента з ім'ям «Производство» для розміщення підприємства, яке знаходиться в м.Запоріжжя. Це буде не популяція, а єдиний об'єкт, який необхідно пов'язати з міткою на карті. Для цього у властивості

агента «Производство» в пункті початкове місце розташування оберемо розташувати агентів: у вузлі і в графі Вузол оберемо з випадуючого меню мітку Запоріжжя, як на рис.4.7

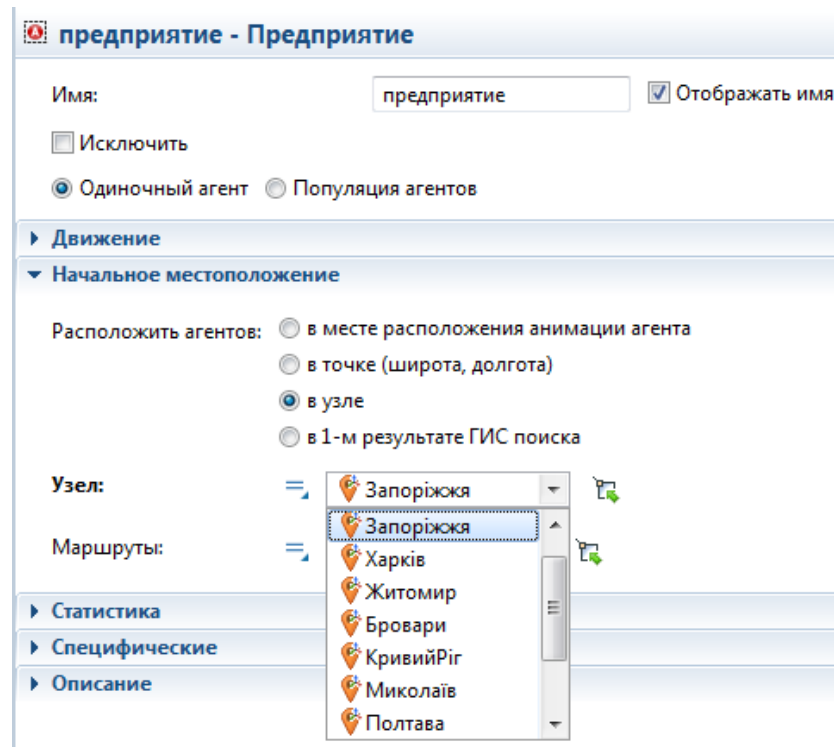


Рисунок 4.7 – Властивості агента «Производство»

Виконаємо запуск моделі. Вигляд моделі з розташованими представництвами і виробництвом наведено на рис.4.8.

З технічного завдання (ТЗ) відомо, що доставка здійснюється вантажівками, щоб внести їх в модель, необхідно створити нову популяцію агентів «Грузовик». Агенту вкажемо параметр client типу Представительство. Параметр буде вказувати на окремі представництва з відповідної популяції.

Для подальшої оптимізації, треба створити окремий параметр кількість вантажівок. Для цього потрібно витягнути з палітри Агент компонент Параметр. Встановіть наступні властивості:

- ім'я: колГрузовиков;
- тип: int;

– значення за замовчуванням: 5.

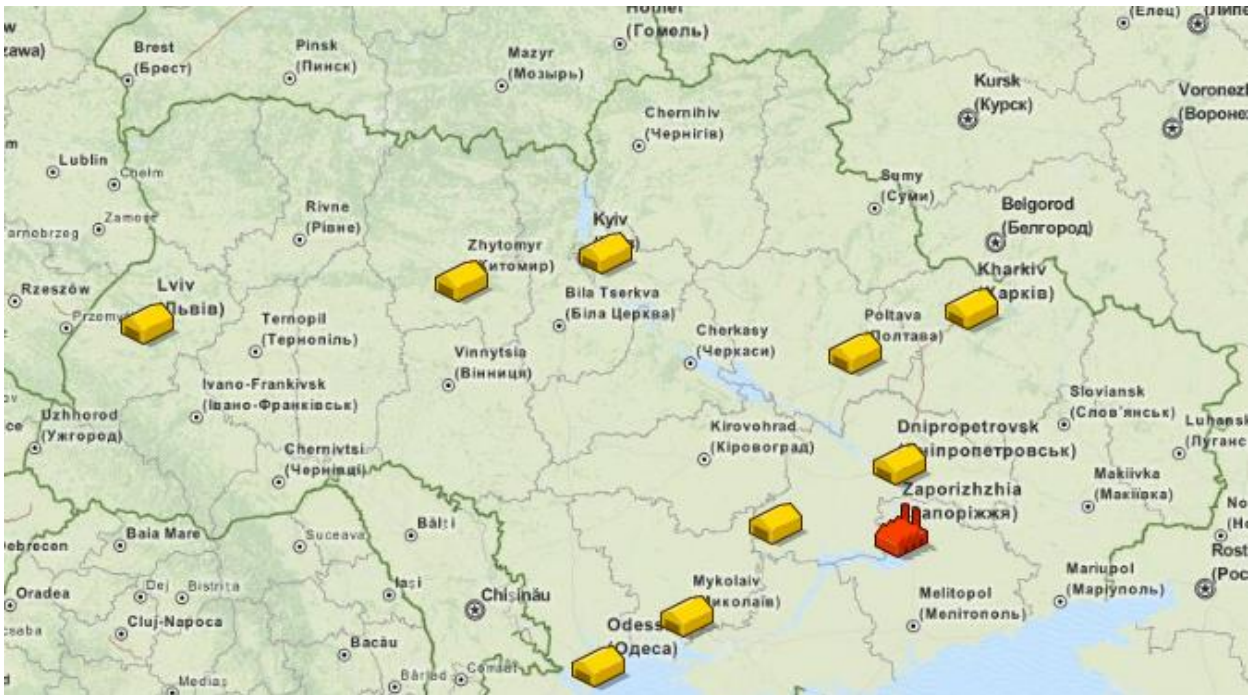


Рисунок 4.8 – Виконання моделі з зазначеними представництвами та виробництвом

Зрозуміло, що все вантажівки належать виробництву (агент «Производство»), необхідно у властивостях популяції агента trucks вказати початкове місце розташування, як на рис.4.9.

З ТЗ відомо, що для отримання товарів і комплектуючих представництва повинні сформувати замовлення. Щоб відобразити це в моделі створимо новий тип агента з ім'ям «Заказ», до якого додамо параметр customer типу Представительство.

Тепер необхідно довести, що вантажівки є ресурсом виробництва. Для цього з вкладки Проекти (або з об'єкта main) відкриємо тип агента «Производство» і перенесемо на нього з палітри Бібліотека моделювання процесів блок Resource Pool, який задає набір доступних ресурсів, в даному випадку – вантажівок. Далі необхідно задати параметри блоку Resource Pool, згідно з рис.4.10.

грузовики - Грузовик

Имя: Отображать имя

Исключить

Одиночный агент Популяция агентов

Популяция: Изначально пуста
 Содержит заданное кол-во агентов
 Загружается из базы данных

Начальное количество агентов:

client: =

▶ Движение

▼ Начальное местоположение

Расположить агентов: в месте расположения анимации агента
 в точке (широта, долгота)
 в узле
 в 1-м результате ГИС поиска

Узел:

Маршруты:

Харків
 Житомир
 Бровари
 КривийРіг
 Миколаїв
 Полтава

Статистика
 Специфические
 Описание

Рисунок 4.9 – Властивості агента «Грузовик»

грузовики - ResourcePool

Имя: Отображать имя

Исключить

Тип:

Количество задано:

Количество ресурсов:

При уменьшении кол-ва:

Новый ресурс:

Скорость: м/с

Базовое местоположение (узлы):

Смены, перерывы, аварии, обслуживание...
 ▼ Специфические

Добавить ресурсы в: Популяцию по умолчанию
 Другую популяцию агентов

Популяция агентов:

Включить сбор статистики:

Рисунок 4.10 – Властивості блоку Resource Pool

На рис.4.10 видно, що значення кількості ресурсів відповідає значенням створеного раніше параметра колГрузовиков, а так само необхідно відзначити, що цей блок додає до нових ресурсів популяцію агентів main.грузовики.

На цьому введення вихідних даних можна вважати закінченим. В якості вихідних даних описані: «представительства» (їх координати), «производство» (координати), «грузовики» (як ресурс виробництва), створена заявка для замовлення запасних частин (як тип агента «Заказ»). Результат запуску моделі на цьому етапі її створення наведений на рис.4.11.

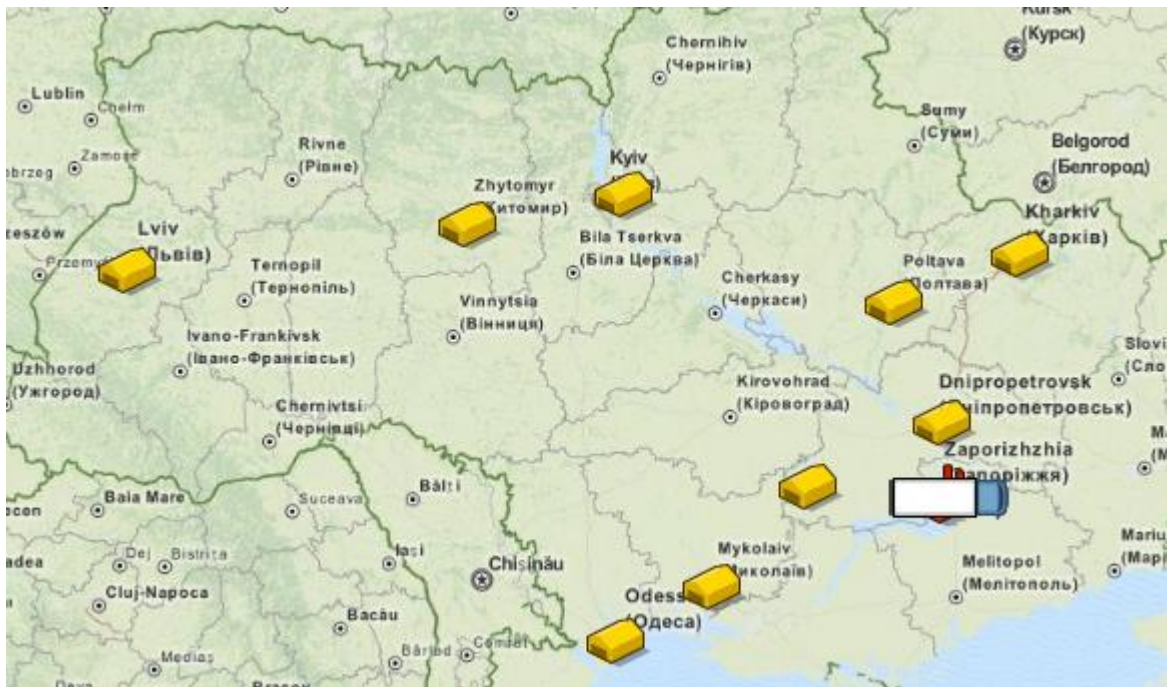


Рисунок 4.11 – Результат розташування агентів в об’єкті ГІС карта

Тепер перейдемо до логіки роботи представництв і виробництва.

4.3.2 Формування замовлення

Логіку роботи комерційних представництв, в даній задачі, можна розглядати як послідовні переходи зі стану нормальної роботи в стан очікування

комплектуючих і назад. Щоб змодельовати цей процес необхідно побудувати діаграму станів.

Будь-яка діаграма станів повинна починатися з елемента Початок діаграми станів – він є точкою входу в діаграму станів (особливо це важливо для багаторівневих діаграм). Його потрібно з'єднати з елементом Стан, який буде відповідати за режим нормальної роботи представництва (той період часу, коли комплектуючі не потрібні). Назвемо цій стан «нормальнаяРабота».

Як було сказано вище робота представнитв ще включає в себе режим очікування комплектуючих, тому необхідно винести ще один елемент Стан в робочу область. Назвемо цей стан «ожиданиеКомплектующих».

Як сказано в тексті завдання, деталі потрібні представнитвам 2 рази на тиждень – це значить, що перехід зі стану нормальної роботи в стан очікування запчастин відбувається із заданою інтенсивністю, щоб відзначити це в діаграмі станів перетягнемо з палітри Діаграма станів об'єкт Перехід, у властивостях якого вкажіть наступні параметри:

- відбувається: з заданої інтенсивністю;
- інтенсивність: 2 рази на тиждень.

Як тільки представництво переходить в стан очікування комплектуючих, повинно бути сформовано і відправлено на підприємство замовлення. Щоб відобразити це в діаграмі станів повернемося до властивостей елемента «ожиданиеКомплектующих». В графі Дія при вході напишемо такий вираз:

```
Заказ заказ = new Заказ (this);
send (заказ, main.производство);
```

У першому рядку формується нове замовлення з параметром `this`, який вказує на представництво, з якого замовлення надходить. Другий рядок коду потрібний для відправки – функція `send`, яка має два аргументи: що відправити (створений вище замовлення) і кому відправити (на виробництво – агенту «производство»).

Щоб діаграма станів повністю відображала логіку роботи представництва, в контексті даної задачі, необхідний ще один компонент Перехід зі стану

«ожиданиеКомплектующих» в стан «нормальнаяРабота». Як відомо з тексту завдання, перехід в нормальний режим роботи здійснюється при отриманні повідомлення. Для цього властивості компонента Перехід повинні відповідати властивостям, зазначеним на рис.4.12.

transition1 - Переход

Имя: transition1 Отображать имя

Исключить

Происходит: При получении сообщения

Тип сообщения: Object

Осуществлять переход: Безусловно
 При получении заданного сообщения:
 Если выполняется условие

Сообщение: "Доставлено!"

Действие:

Доп. условие:

Рисунок 4.12 – Властивості елемента Перехід

На рис.4.13 показано описана вище діаграма станів. Тепер представництво може відправляти замовлення на комплектуючі.

Переходимо до обробки і виконання даного замовлення.

4.3.3 Обробка замовлення

Після отримання замовлення, на виробництві виділяють ресурс (вантажівка) для виконання даного замовлення. Його завантажують замовленими комплектуючими, на що потрібно від двох до трьох годин, і відправляють в представництво. Там машину розвантажують (протягом двох-трьох годин),

після чого надсилається сповіщення про доставку, і вантажівка повертається на виробництво, стаючи вільним ресурсом.

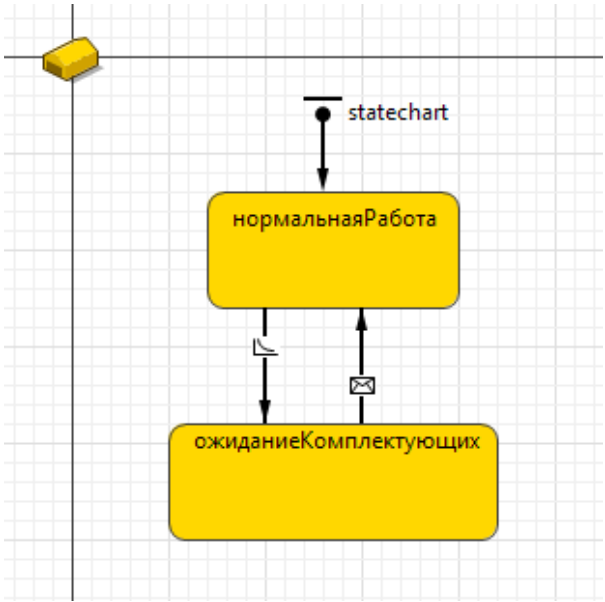


Рисунок 4.13 – Логіка роботи агента «Представительство»

На рис. 4.14 наведена модель даного процесу, далі кожен блок розглянемо окремо.

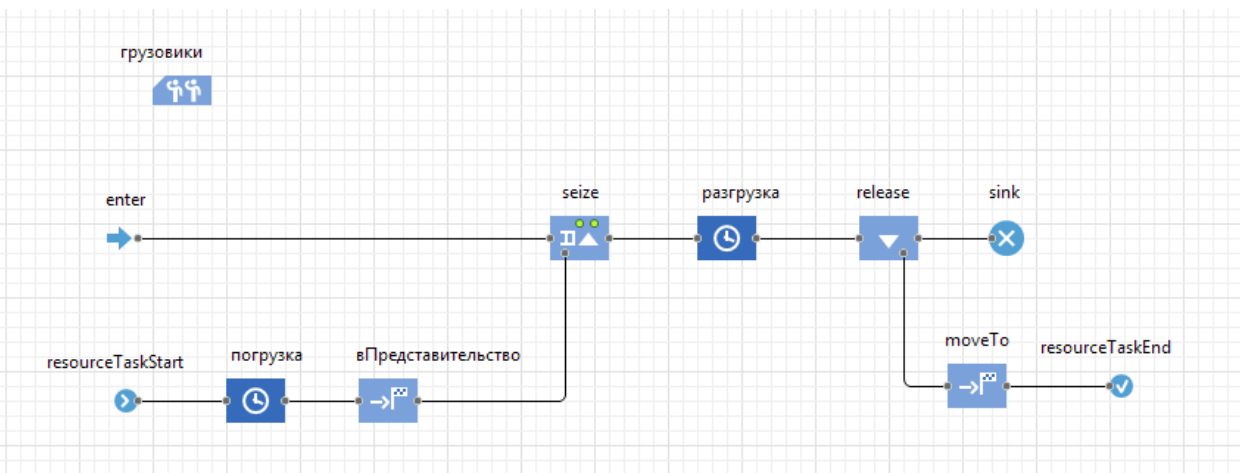


Рисунок 4.14 – Обробка і виконання замовлення на підприємстві

Щоб реалізувати показаний вище процес, була використана бібліотека моделювання процесів.

Вхід в процес здійснюється через блок Enter, в які надходять замовлення. У його властивостях заданий тип агента: Заказ.

Далі отримане замовлення надходить в чергу на очікування ресурсів, для цього використовується блок Seize, який і відповідає за захоплення ресурсів. У властивості блоку Seize в пункті Набір ресурсів задані вантажівки.

Перед тим, як ресурс буде захоплений, він повинен бути підготовлений, що в нашому випадку означає завантаження машини. Підготовка ресурсів і їх відправка відбувається в спеціальному підпроцесу для ресурсів, який починається з блоку Resource Task Start. У властивостях даного блоку необхідно вказати:

- тип ресурсу: Грузовик;
- починати завдання: всім ресурсам.

На завантаження ресурсу потрібен час, який бде моделювати блок Delay. Властивості блоку Delay наведені на рис.4.15.

The screenshot shows the configuration window for a 'Delay' block in a simulation environment. The window title is 'погрузка - Delay'. The settings are as follows:

- Имя:** погрузка (with a checkbox 'Отображать имя' checked).
- Исключить:** checkbox is unchecked.
- Тип:** radio buttons for 'Определенное время' (selected) and 'Пока не вызван метод stopDelay()'.
- Время задержки:** text field contains 'uniform(2,3)' and a dropdown menu is set to 'часы'.
- Вместимость:** text field contains '1'.
- Максимальная вместимость:** text field is empty.
- Место агентов:** dropdown menu is empty.
- Специфические:**
 - Выталкивать агентов:** checkbox is unchecked.
 - Вернуть агента в исходную точку:** checkbox is checked.

Рисунок 4.15 – Властивості блоку Delay (завантаження)

Для часу затримки вказано рівномірний закон розподілу між двома і трьома годинами: uniform (2, 3).

Після того, як ресурс завантажений, його відправляють в представництво. Змоделюємо це за допомогою елементу Move To. В його властивостях необхідно вказати пункт призначення, як показано на рис.4.16.

Рисунок 4.16 – Властивості блоку Move To (вПредставительство)

Ресурс (вантажівка) відправлений у блок Seize, тепер цьому ресурсу повинна бути передана інформація всередині замовлення, щоб він знав, куди саме повинен бути доставлений вантаж. Для цього у властивості об'єкта Seize і в пункті Дії в графу При захопленні ресурсу напишемо наступний вираз:

```
((Грузовик) unit) .client = agent.customer;
```

В даному вираженні значенням параметра client, який зберігається всередині ресурсу Грузовик, присвоєно значення параметра customer агента Заказ.

Як тільки вантажівка приїде в представництво, необхідно здійснити її розвантаження. Цей процес моделюється аналогічно процесу завантаження машини, а саме блоком Delay (рис.4.14).

При моделюванні логіки роботи представництва було зазначено, що перехід зі стану очікування запчастин в стан нормальної роботи відбувається при отриманні відповідного повідомлення. Дане повідомлення відправляється, як тільки закінчена розвантаження, тобто відбувається вихід з блоку «разгрузка». Для відправки використана вбудована функція Send, яка має два аргументи (що відправити, кому відправити).

Після виконання замовлення ресурс стає вільним (блок Release, рис.4.13), а агент (Заказ) вирушає в блок Sink, де буде видалений. Так як ресурс звільнений, він повинен бути відправлений на виробництво, для цього скористаємося ще раз блоком Move To, в якому вкажемо місце призначення агент «производство», який розташований на агента main (рис.4.16).

Підпроцес для ресурсу повинен бути закінчений блоком Resource Task End, для того, щоб ресурс повернувся до загального пулу з ресурсами і був доступний для нового захоплення.

На цьому процес обробки замовлень готовий, залишилося тільки зробити так, щоб всі вхідні замовлення надходили на блок Enter обробки.

Усі вхідні повідомлення обробляються в стандартному блоці connections, який за замовчуванням існує всередині кожного агента. Властивості блоку connections агента «производство» наведено на рис. 4.17.

В поле Дія при отриманні повідомлення введена команда, яка вказує, що блок Enter повинен взяти на обробку (функцією take) повідомлення, що прийшло (використовуючи локальну змінну msg). Тип повідомлення вказується згідно зі створеним раніше в агента «производство».

Модель процесу доставки комплектуючих до представництв готова і тепер можна перейти до запуску і оптимізації створеної моделі.

4.4 Запуск та оптимізація моделі

Перед тим, як запустити модель, необхідно змінити одиниці модельного часу на дні. Результат виконання моделі представлений на рис.4.18.

connections - Связь с агентами

Имя:

Отображать имя Исключить

Видимость: да

Это стандартная связь соединения агентов

Тип агента:

Набор ссылок
 Одиночная ссылка

Это стандартная связь является двунаправленной

▼ Взаимодействие

Эти действия выполняются для сообщений, поступающих во все связи агента

Тип сообщения:

Действие при получении сообщения:

Перенаправлять сообщение в:

Рисунок 4.17 – Властивості блоку connections

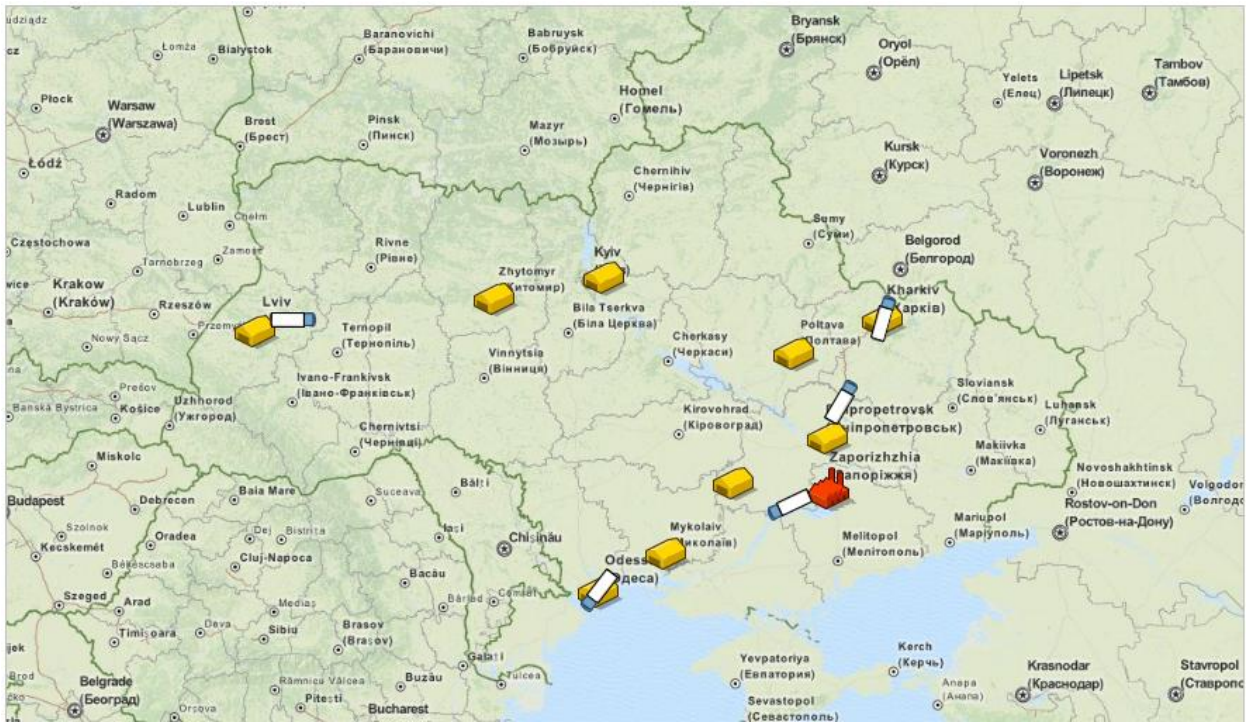


Рисунок 4.18 – Модель постачання комплектуючих до представництв

Модель, що виконується, дозволяє побачити, як вантажівки виїжджають з підприємства, доїжджають до представництв і повертаються назад. Результат запуску діаграми процесів для агента «производство» представлений на рис. 4.19.

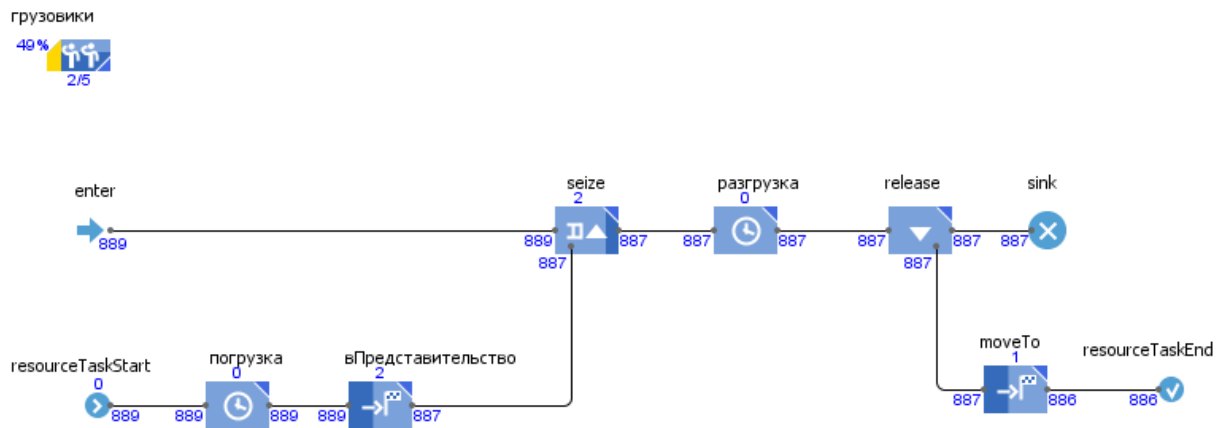


Рисунок 4.19 – Результат запуску діаграми процесу постачання товарів

Змінюючи масштаб карти, можна побачити, що вантажівки рухаються за реальними маршрутами, які, як і сама карта, завантажуються з мережі під час виконання моделі (рис.4.20).

Щоб оцінити завантаженість транспортного парку підприємства відкриємо агента «производство» при запусненій моделі. Там розміщений блок Resource Pool. По ходу обробки замовлення фіксується кількість захоплених ресурсів, по якому можна оцінити раціональність використання ресурсів. При використанні п'яти вантажівок, середня завантаженість ресурсів становить приблизно 50 відсотків, тобто тільки три з п'яти вантажівок задіяні (рис.4.21).

За умовою завдання, завантаженість ресурсів при доставці повинна бути не більше 85%, отже, частина ресурсів простоює і приносить збитки підприємству. Проведемо оптимізацію.

Щоб провести оптимізацію необхідно викликати новий експеримент, тип якого оптимізація.

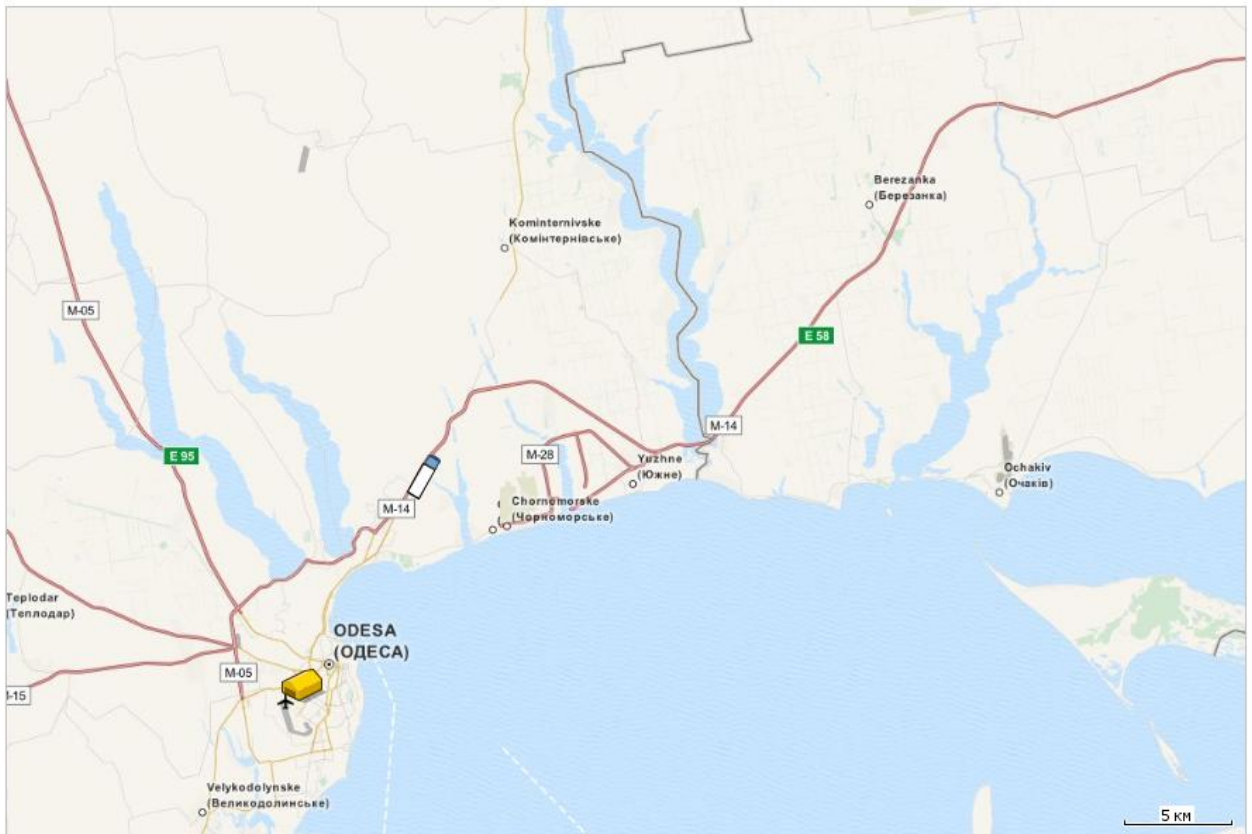


Рисунок 4.20 – Рух вантажівки за реальним автомобільним шляхом на карті

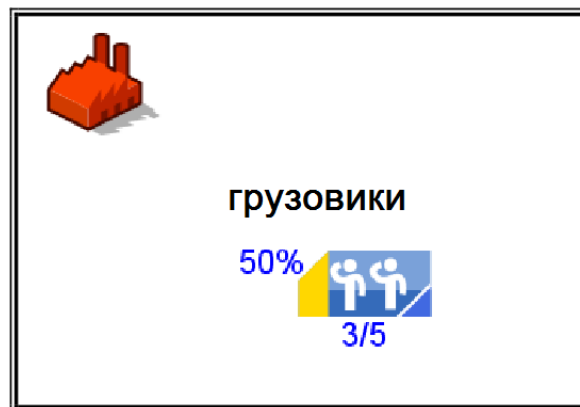


Рисунок 4.21 – Використання ресурсів при постачанні комплектуючих

Перейдемо у властивості експерименту і змінимо їх відповідно до рис.4.22.

Змінюваним параметром обраний колГрузовиков (кількість вантажівок), який змінюється дискретно від 1 до 5. За цільову функцію прийнята середне

завантаження всіх вантажівок, яка максимізована з обмеженням (максимум 85%) в пункті вимоги. Рядок коду `root.предприятие.грузовики.utilization()` дозволяє звернутися до параметру середнього завантаження `utilization()` ресурсу «грузовики» агента «производство», що знаходиться на кореневому агенті `root` (в даному випадку `main`).

Optimization - Оптимизационный эксперимент

Имя: Исключить

Агент верхнего уровня:

Целевая функция: минимизировать максимизировать

`root.предприятие.грузовики.utilization()`

Количество итераций:

Автоматическая остановка

Максимальный размер памяти: Мб

Параметры

Параметры:

Параметр	Тип	Значение			
		Мин.	Макс.	Шаг	Начальное
колГрузовиков	дискретный	1	5	1	

Требования

Требования (проверяются после "прогона" для определения того, допустимо ли найденное решение):

Вкл.	Выражение	Тип	Гран...
<input checked="" type="checkbox"/>	<code>root.предприятие.грузовики.utilization()</code>	<=	0.85

Случайность

Рисунок 4.22 – Властивості оптимізаційного експерименту

На рис.4.23 наведено приклад виконання експерименту.

Як видно з рис. 4.23, краще допустиме значення завантаженості ресурсів знаходиться на позначці 77% при використанні 3-х вантажівок. при використанні 4-х вантажівок цільова функція перевищує ліміт, що заданий у властивості експерименту.

Доставка : Optimization

Запустить

	Текущее	Лучшее
Итерация:	5	1
Функционал: ↑	0.506	0.767

Параметры

колГрузовиков: 4

Копировать лучшее решение в буфер

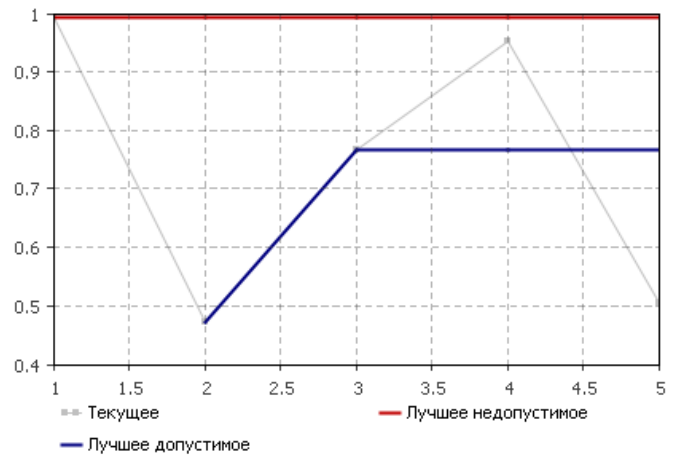


Рисунок 4.23 – Оптимізація процесу доставки комплектуючих

Відкриємо властивості параметра колГрузовиков і встановимо значення за замовчування: 3.

Запустимо симуляцію моделі. Середнє значення завантаженості ресурсів на підприємстві коливатиметься між 75-77%.

ВИСНОВКИ

В результаті виконання дипломної роботи була побудована імітаційна модель, яка моделює ланцюг поставок товарів і комплектуючих підприємством JB-PLAST, яке знаходиться в м. Запоріжжя, своїм комерційним представництвом розташованим в дев'яти містах України.

Підприємство має в розпорядженні свій парк вантажних автомобілів. Коли виробник отримує замовлення від представництва, він перевіряє кількість готового товару на складі і відправляє вантажівку з цим товаром до представництва. Модель є багатопідхідною. Представництва, вантажівки і підприємство (виробництво) є агентами, кожен з яких має свою поведінку: карта станів задає поведінку представництва, а діаграма процесів – відправку і обробку замовлення підприємством.

Агенти живуть в просторі ГІС. Імена географічних місць, в яких розташовані представництва, можуть зчитуватися з файлу Excel. Вбудований пошук по ГІС карті знаходить місця і поміщає в них агентів. Вантажівки рухаються по існуючій мережі доріг, а маршрути створюються, коли автомобілі починають рух до місця призначення. Постачальником карт є ресурс OSM, тому можна завантажувати файли формату PBF.

В роботі був виконаний оптимізаційний експеримент. Визначена оптимальна кількість вантажівок. За цільову функцію прийняте середнє завантаження всіх вантажівок, яка максимізована з обмеженням (максимум 85%). Оптимізаційний експеримент показав, що краще допустиме значення завантаженості ресурсів знаходиться на позначці 77% при використанні 3-х вантажівок. При використанні 4-х вантажівок цільова функція перевищує ліміт, що заданий у властивості експерименту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Хемди А. Таха Имитационное моделирование. Введение в исследование операций. М.: Вильямс, 2007. 917 с.
2. Борщёв А.В. Практическое агентное моделирование и его место в арсенале аналитика. URL: <http://www.gpss.ru/immod05/p/borshev/> (дата звернення 13.05.2019)
3. Строгалев В. П., Толкачева И. О. Имитационное моделирование. МГТУ им. Баумана, 2008. 697 с.
4. Шрайбер Т. Дж. Моделирование на GPSS. М.: Машиностроение, 1980. 592 с.
5. Боев В. Д. Моделирование систем. Инструментальные средства GPSS WORLD. СПб.: БХВ-Петербург, 2004. 368 с.
6. Боев В. Д, Сыпченко Р. П. Компьютерное моделирование. Элементы теории и практики. СПб.: Военная академия связи, 2009. 432 с.
7. Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб.: БХВ-Петербург, 2005. 312 с.
8. Рыжиков Ю.И. Имитационное моделирование. Теория и технология. СПб.: КОРОНА принт, 2004. 384 с.
9. XJ Technologies. Офіційний сайт. URL: <http://www.xjtek.com> (дата звернення 13.05.2019)
10. Java For AnyLogic Users. URL: http://www.xjtek.com/files/book/Java_for_AnyLogic_users.pdf (дата звернення 13.05.2019)
11. OpenStreetMap. URL: <https://ru.wikipedia.org/wiki/OpenStreetMap> (дата звернення 13.05.2019)
12. PBF Format. URL: http://wiki.openstreetmap.org/wiki/PBF_Format (дата звернення 13.05.2019)