

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра Комп'ютерних інтелектуальних технологій та систем
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Агентська модель логістичних послуг

Виконав:
студент II курсу, групи КІТм-19-1
Дольнев Р.О.

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми Освітньо-професійна
Освітня програма Комп'ютерні інтелектуальні технології
(повна назва освітньої програми)

Керівник: проф. Аксак Н.Г.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТС Руденко О.Г.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
Кафедра _____ Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти _____ другий (магістерський)
Спеціальність _____ 123 – Комп'ютерна інженерія
Тип програми _____ Освітньо-професійна
Освітня програма _____ Комп'ютерні інтелектуальні технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Дольневу Руслану Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Агентська модель логістичних послуг

затверджена наказом по університету від “ 11 ” листопада 2020 р. № 1582 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 10 грудня 2020 р.

3. Вхідні дані до роботи _____ Агентно-орієнтоване моделювання

Програмне забезпечення AnyLogic

Логістичні послуги

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз проблемної області і постановка задачі

Агентно-орієнтована модель логістичних послуг

Реалізація та тестування моделі за допомогою AnyLogic

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайдів 13

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд стану проблеми та постановка задачі	11.11 – 13.11	
2	Аналіз літератури за напрямком магістерської роботи	13.11 – 20.11	
3	Аналіз логістичної ситуації в регіоні	21.11 – 23.11	
4	Вибір методів рішення для реалізації та їхнє обґрунтування	24.11 – 26.11	
5	Розробка агентної моделі логістичних послуг	26.11 – 30.12	
6	Оформлення пояснювальної записки	01.12 – 07.12	
7	Оформлення графічної частини	08.12 – 10.12	

Дата видачі завдання 11 листопада 2020 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Аксак Н.Г.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 87 с., 22 рис., 1 табл., 8 прикл., 2 дод., 37 джерел.

ЛОГІСТИКА, АГЕНТ, АГЕНТНО-ОРІЄНТОВАНИЙ ПІДХІД, ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ, СИСТЕМНА ДИНАМІКА, ANYLOGIC.

Метою атестаційної роботи є створення агентно-орієнтованої моделі логістичних послуг.

У ході виконання атестаційної роботи було розроблено агентну модель логістичних послуг Харківщини, а саме послуг вантажних перевезень, що дозволяє більш детально прослідити за логістичною ситуацією. Методом вирішення поставлених завдань є широке застосування сучасних інформаційних технологій. Агентна модель порівнюється з моделлю, побудованою за допомогою системної динаміки.

Моделі побудовано в програмному засобі Anylogic.

ABSTRACT

Master's thesis: 87 pages, 22 figures, 1 tables, 8 examples, 2 appendices, 37 sources.

LOGISTICS, AGENT, AGENT-ORIENTED APPROACH, SIMULATION MODELING, SYSTEM DYNAMICS, ANYLOGIC.

The purpose of master`s thesis is to create an agent-oriented model of logistics services.

In the course of attestation work, an agent model of logistics services of Kharkiv region was developed, namely freight transportation services, which allows to follow the logistics situation in more detail. The method of solving the tasks is the widespread use of modern information technology. The agent model is compared with the model, which built by using system dynamics.

The models are built in the Anylogic software.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 АКТУАЛЬНІСТЬ ПРОБЛЕМИ ТА ОГЛЯД ЛІТЕРАТУРИ	11
1.1 Дослідження транспортно-технологічного процесу	11
1.2 Агентно-орієнтоване моделювання в логістиці	13
1.3 Приклади застосування агентів для проблем логістики	15
1.4 Постановка задачі.....	22
2 АНАЛІЗ ІНСТРУМЕНТАРІЮ ДЛЯ АГЕНТНО-ОРІЄНТОВАНОГО МОДЕЛЮВАННЯ	23
2.1 Основні парадигми імітаційного моделювання.....	23
2.2 Методи імітаційного моделювання складних систем	25
2.2.1 Моделювання з різними рівнями абстракції	27
2.3 Моделювання в AnyLogic	29
3 РОЗРОБКА АГЕНТСЬКОЇ МОДЕЛІ ЛОГІСТИЧНИХ ПОСЛУГ В ХАРКІВСЬКІЙ ОБЛАСТІ	34
3.1 Методологія моделювання складних реальних систем	34
3.1.1 Модель дифузії Басса.....	36
3.1.2 Модель «Хижак-жертва».....	38
3.2 Напрямки розвитку агентно-орієнтованого моделювання	38
3.3 Агентна модель логістичних послуг	40
3.3.1 Модель Main	42
3.3.2 Блок замовлення Order	43
3.3.3 Модель Transport.....	45
3.3.4 Модель Handling.....	49
3.3.5 Модель Warehouse.....	53
3.3.6 Модель Link	54

4 ЕКСПЕРЕМЕНТАЛЬНЕ МОДЕЛЮВАННЯ.....	57
4.1 Моделювання агентної моделі логістичних послуг	57
4.2 Тестування агентної моделі логістичних послуг	64
ВИСНОВКИ.....	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	69
ДОДАТОК А Графічний матеріал атестаційної роботи	73
ДОДАТОК Б	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

ТТП – транспортно-технологічний процес

ЄІП – єдиний інформаційний простір

ІР – інформаційний ресурс

ІС - інтегровані інформаційні системи

СД – системна динаміка

СДМ – системно-динамічна модель

АОМ – агентно-орієнтована модель

TEU (twenty-foot equivalent unit) – умовна одиниця вимірювання
місткості грузових транспортних засобів

ІМ – імітаційне моделювання

ДПМ – дискретно-подійне моделювання

ШІ – штучний інтелект

ВВУ – з вуст в уста

ВСТУП

Ефективне функціонування сучасних транспортно-технологічних процесів неможливо уявити без постійного використання новітніх інформаційних технологій. Можливості швидкого реагування на потреби ринку і активний інформаційний обмін між ланками транспортної мережі є невід'ємною характеристикою поставок товарів будь-якого роду. Програмні комплекси та інформаційні системи, створені для планування і підтримки прийняття різних комерційних рішень, є ключовим фактором для забезпечення високої якості обслуговування і постійного вдосконалення транспортних операцій. З розвитком інтернет-технологій очевидним стає той факт, що перспективні напрямки розвитку ринку логістичних послуг орієнтовані на активне використання електронних форм забезпечення ттп [1, 2]. Історично склалося так, що накопичення і використання інформаційних ресурсів здійснювалося окремо в кожному виді транспорту з метою забезпечення його ефективного управління і функціонування. Аналогічно ситуація складалася в інших сферах логістики. Традиційно застосування різних інформаційних технологій в логістичних системах відбувалося розрізнено, тільки в інтересах конкретного учасника ТТП і з плином часу стало очевидно, що такий підхід неефективний і надзвичайно важлива інтеграція ІР для отримання єдиного інформаційного простору транспортно-логістичної мережі, оскільки ЄІІ забезпечить необхідні швидкості і обсяги обміну інформацією, а також надасть найбільш точні дані, необхідні користувачеві, дозволить детально аналізувати техніко-економічні параметри різних варіантів, моделювати різні процеси з метою прийняття оптимальних рішень [1, 2]. Однак інтеграція ІР в транспортній логістиці є надзвичайно складною проблемою. У міру розвитку інтеграційних процесів виникають більш актуальні проблеми оптимізації та організації інформаційних потоків, що, безумовно, пов'язано з особливостями

предметної області ТТП:

- високий динамізм предметної області;
- складність отримання повної, точної і достовірної інформації;
- стресовий характер ситуацій;
- дефіцит часу прийняття рішень;
- збої, відмови і виходи з ладу технічних засобів;
- висока ймовірність помилкових дій людей;
- форс-мажорні обставини.

Значною перешкодою також є низька ефективність організаційного, технічного, інформаційного і програмного забезпечення взаємодії учасників ТТП. Кожен суб'єкт орієнтується на досягнення своїх цілей, пов'язаних з нерозумінням системного ефекту від взаємодії, оцінює ефективність свого функціонування за типовими критеріями і показниками. Таким чином, з огляду на системний характер проблеми, очевидно, що її рішення полягає в розробці та реалізації на основі єдиної методології комплексу узгоджених заходів правового, організаційного, фінансово-економічного [36], методичного, інформаційного та технологічного характеру, що враховують інтереси всіх учасників ТТП. Для забезпечення ефективності цих процесів необхідно використовувати інтегровані інформаційні системи, що дозволяють реалізувати сучасні вимоги щодо зниження експлуатаційних витрат і підвищення рівня якості транспортних послуг. У цих умовах зазначені ІС дозволяють ефективно вирішувати основну логістичну задачу при організації ТТП, а саме узгодження режимів виробництва і споживання з сервісним режимом надання транспортних засобів, прикордонної та митної обробкою, зберіганням, переробкою і розподілом продукції. На базі ІС з'являється можливість побудувати ЄІП, яке забезпечить координацію робіт усіх організацій при здійсненні інтермодальних перевезень відповідно до прийнятої логістикою обробки вантажів [37]. Для підтримки ЄІП ІС повинні забезпечувати взаємодію з різних транспортних і інших організацій всередині країни і за кордоном [3].

1 АКТУАЛЬНІСТЬ ПРОБЛЕМИ ТА ОГЛЯД ЛІТЕРАТУРИ

1.1 Дослідження транспортно-технологічного процесу

Мотивація цього дослідження цілком обумовлена практичними проблемами, що відчуваються у логістичній галузі. Будь-хто буде глибоко вражений складністю управління логістикою та тим, як важко підтримувати ефективність логістичних операцій, важливість яких багато компаній ігнорує. Та навіть для компаній, що це розуміють, логістика є найбільшим обмеженням, поряд з розширеним ланцюгом поставок, що відображається високою вартістю, тривалим часом доставки замовлень, частим дефіцитом потужності або низьким коефіцієнтом використання, непередбачуваністю попиту та нестабільним обслуговуванням тощо. Проблеми логістики суттєво перешкоджають іншим операціям у ланцюгу поставок і заважають компаніям займати хоч якусь ланку на висококонкурентному ринку. Більш конкретно, спостерігається дуже низький рівень використання та затримки щоденних логістичних операцій. Це безпосередньо спричиняє високу вартість доставки для самих компаній та поганий сервіс доставки для клієнтів. Основною причиною такого низького використання є відсутність достатнього масштабу. Наприклад, для того, щоб підтримувати частоту доставки та зобов'язання для замовників, компаніям доводиться щодня відправляти замовлення незалежно від того, чи ефективно використовувались вантажівки, контейнери чи простір літака. Крім того, транспортні рішення, такі як залізничний транспорт, часто вимагають мінімального порогового обсягу для надання послуги. Це призводить до труднощів із залученням достатнього обсягу за дуже короткий проміжок часу і, отже, затримкою транспортування. Висока нестабільність замовлень, а також їх незбалансований темп та розмір у споживачів, теж є однією з основних причин низького використання та затримок. Логістичний потенціал потрібно планувати заздалегідь згідно з

прогнозом, але великі коливання попиту призводять до прийняття консервативної та високої буферної стратегії, при цьому пропускна спроможність зберігається, щоб справлятися з невизначеністю. Однак емність на логістичному ринку не схожа на інвентар, який можна перенести: вона може використовуватися лише протягом певного часу, а якщо не використовується, вона видихається, спричиняючи часті відходи та низький рівень використання. З іншого боку, незбалансовані хвили замовлення можуть тиснути на запас потужності. Клієнти, як правило, випускають великі замовлення, часто в кінці циклу отримання замовлень, створюючи надзвичайні проблеми при підготовці потужностей як у виробництві, так і в логістиці. За цих обставин дефіцит поставок також є неминучим, що спричиняє затримки відвантаження.

Ці проблеми глибоко вкорінені в багатьох компаніях в поточній конфігурації ланцюга поставок та логістики і свідчать про те, що використання традиційних та внутрішніх методів оптимізації важко і навряд чи призведе до значних поліпшень. На прикладі Китаю, проаналізувавши мережу вантажних потоків, було встановлено, що вихідні логістичні мережі багатьох виробників є паралельними та сильно перекриваються. Всі ці виробництва централізовано скупчені в одній частині Китаю, виробляючи та відвантажуючи замовлення в ті самі регіони або тих самих замовників. Багато з них також перебувають у відносинах співпраці у постачанні та виробництві (так званий "вертикально-скупчений ланцюжок поставок"). Отже, виникає питання в тому, що якщо їх можна згрупувати для виробництва та постачання, чому їх не можна згрупувати для просування спільної логістики виїзду? Насправді було визначено багато можливостей для співпраці в галузі логістики, яка могла б принципово змінити поведінку поточної логістичної системи. Дивлячись на успіх випробувань в даному напрямку стає очевидно, що цей новий стиль управління логістикою має великий потенціал.

На прикладі того ж самого Китаю, можна побачити, наскільки

важливою може бути логістика. При порівнянні витрат на логістику щодо ВВП Китаю, США та Японії (три найбільші економіки) помітно бачимо, що Китай витрачає більше ніж удвічі більше на логістику для підтримки свого зростаючого ВВП. Відсоток високих витрат (близько 1/5 ВВП) сильно вказує на низький рівень ефективності поточних логістичних операцій у різних галузях промисловості, що становлять китайську економіку. Проблеми втрати енергії, заторів транспорту та забруднення навколишнього середовища стають серйозним і жахливим фактом у Китаї, незважаючи на, або, можливо, через його швидке зростання. Погано впроваджена логістика значною мірою сприяла зростанню гостроти цих питань, і це потрібно терміново вирішувати. Реально, китайська економіка повинна змінити свою модель зростання з великої на інтенсивну, збільшуючи якість, а не кількість. Це вимагає логістики як ключового гравця в економічній діяльності, щоб вона стала більш інтенсивним, а не екстенсивним типом розвитку.

З деякими з цих проблем стикаються і в Україні, тобто це справедливо не тільки для Китаю.

1.2 Агентно-орієнтоване моделювання в логістиці

Сучасні логістичні процеси та системи можуть мати надзвичайно складну динаміку. Агентно-орієнтоване моделювання постає потужним інструментом моделювання для проектування, аналізу та контролю таких логістичних систем. Однак складність самої моделі може бути надзвичайною і потрібні інструменти математичного метамоделювання, які узагальнюють інформацію та дозволяють швидко і точно приймати рішення та проектувати системи контролю.

Як і будь-яка інновація людини, мислення, засноване на агентах, розвивалося не в порожньому просторі. Було пов'язане обговорення в декількох областях, але найважливішими галузями досліджень для зародження агентного мислення були штучний інтелект, об'єктно-орієнтоване програмування та дизайн інтерфейсу

людина-комп'ютер [22]. Поява дизайну, заснованого на агентах, у штучному інтелекті відбулася в 1985р., названа на честь "автономних досліджень агента", "досліджень на основі поведінки", "ШІ знизу вгору" або "підходу анімату". Всі ці назви описують характеристики нового виду моделювання: це підхід, коли системи проектуються знизу вгору, і, отже, «автономні» компоненти, а саме агенти, діють відповідно до своєї внутрішньої схеми, а не із зовнішніми командами. Це породжує непередбачувану поведінку, яка певною мірою нагадує життя, а отже, ближче до природного інтелекту, ніж поведінка, яку виробляють попередні підходи [23].

Існування нових властивостей є мотивацією для моделювання та імітації на основі агентів. Часто агентне моделювання спрямоване на виявлення деяких виникаючих властивостей та процесу, що призводить до них. Моделювання взаємодії між агентами часто є «дивним, оскільки важко передбачити всі наслідки навіть простих форм взаємодії» [24]. Важливими аспектами цих систем є складність та децентралізація. Трохи важко знайти чітку різницю між дизайном програмного забезпечення на основі агента та більш загальним об'єктно-орієнтованим дизайном. Поняття агента, однак, є суперечливим навіть у обмеженому середовищі комп'ютерних вчених, які займаються дослідженнями моделей агентів та технологій. Франклін і Граессер роблять висновок, що «автономний агент - це система, яка знаходиться всередині середовища та є частиною середовища, яка сприймає це середовище та діє на нього з часом, дотримуючись власного порядку денного, і таким чином здійснювати те, що відчуває в майбутньому» [23]. Це визначення відокремлює автономних агентів від інших фрагментів коду, оскільки об'єкт не має, наприклад, власного порядку денного. Також зазначено, що агенти відрізняються від інших програм тим, що вони автономні, гнучкі, вважається, що вони мають власну нитку контролю та демонструють контроль над своєю поведінкою.

Одне з найпоширеніших визначень агента [24] визначає набір властивостей, які повинні характеризувати суб'єкт, щоб ефективно називати його агентом, і, зокрема, автономність (можливість діяти без втручання

людей та певна ступінь контролю над власним станом), соціальні здібності (можливість взаємодії, використовуючи якусь агентурну мову спілкування), реактивність (можливість сприймати середовище, в якому він знаходиться, і реагувати на сприйняті зміни) та активність (можливість проявити ініціативу, починаючи якусь діяльність відповідно до внутрішніх цілей, а не як реакція на зовнішній стимул). Автори вважають це слабким визначенням агентства, хоча воно, як правило, вже занадто обмежене, щоб характеризувати як агентів більшість сутностей, що заповнюють моделі, засновані на агентах, для моделювання в різних сферах [25].

1.3 Приклади застосування агентів для проблем логістики

Застосування агентного моделювання охоплює широкий спектр областей та дисциплін, починаючи від моделювання поведінки агентів на фондовому ринку та ланцюгах поставок до прогнозування розповсюдження епідемій та загрози біовійни, від моделювання соціально-економічних систем до розуміння купівельної поведінки споживачів, від оптимізації логістики до моделювання залучення сил на полі бою чи в морі та багато інших. Деякі з цих додатків мають масштабний характер, в яких система моделюється дуже детально, тобто використовуються докладні дані, моделі перевіряються, а результати мають на меті інформувати про політику та прийняття рішень. Ці додатки стали можливими завдяки прогресу в розробці спеціалізованого програмного забезпечення для моделювання на основі агентів, нових підходах до розробки агентів на основі агентів, доступності даних на зростаючих рівнях деталізації та вдосконаленні продуктивності комп'ютера.

На відміну від системної динаміки, яка головним чином зосереджується на взаємозв'язку між макропроменливими елементами системи зверху вниз, агентне моделювання намагається змоделювати, як локальні та передбачувані взаємодії між мікрокомпонентами системи можуть генерувати складну поведінку на рівні системи.

В останні роки агентне моделювання було прийнято як головний інструмент вирішення складних проблем у галузі логістики та транспорту. Це пов'язано не тільки з обговореною вище причиною, але також з кількома особливостями, які роблять цей підхід більш цінним, ніж системно-динамічний, який головним чином зосереджується на взаємозв'язку між макро-змінними системи зверху вниз, на відміну від моделювання на основі агентів, де відбуваються локальні взаємодії між мікрокомпонентами системи, створюючи таким чином складну поведінку на рівні системи. Насправді, агентне моделювання є особливо ефективним у разі взаємодії між агентами, які є нелінійними, розривними або дискретними, коли простір має вирішальне значення і позиції агентів не є фіксованими, коли популяція неоднорідна і кожен індивід демонструє відмінності, коли топологія взаємодій є складною, коли агенти виявляють складну поведінку, включаючи навчання та адаптацію.

Спеціалісти з міської логістики лише нещодавно звернулися до АОМ, щоб моделювати та імітувати різні аспекти теми. Основною метою більшості статей про АОМ в літературі ЛМ є зображення взаємодії між агентами через грошові потоки і товарів, а потім оцінка запровадження стратегічних заходів з точки зору економіки та впливу на навколишнє середовище.

В одній з перших концептуальних реалізацій АОМ у контексті ЛМ [27], імітують рух транспортних потоків та вперше включили поведінку та цілі зацікавлених сторін при оцінюванні. З подібними зусиллями, комбінований підхід на основі агентів для проблеми маршрутів транспортних засобів запропонований у роботі [28]. Адаптивне агентне навчання за попереднім досвідом змодельовали в роботі [29] з використанням алгоритму Q-навчання для обчислення функції значення агента, а саме прибуток, включаючи очікувані значення майбутніх станів агента та поведінки, а також швидкості навчання, завдяки якій агенти адаптують свою поведінку. Однак калібрування параметрів у цій попередній роботі не пропонується. Рішення агентів у попередніх моделях в основному обумовлені витратами та тільки

основні транспортні сервіси змінилися усереднені.

Тільки у роботі [30] ввели поняття бізнес-моделі в рамках їх концептуального каркасу для моделювання міських ланцюгів поставок, щоб визначити ділові та оперативні рішення при зміні логістичних послуг серед підприємств. Таким чином, існуюча література зосереджена на оцінці впливу стратегічних заходів на системи ЛМ, і не вдається представити бізнес-модель агентів ЛМ як комплексний інструмент ідентифікації ділових та операційних факторів, а також оцінити вдосконалення сервісів та успіх ЛМ ініціатив.

В агентському моделюванні транспортних коридорів ланцюга поставок [6] представляють використання мультиагентних систем для моделювання та імітування деяких аспектів транспортних коридорів, зосереджуючи увагу на витратах на розвантаження контейнерів, перевантаження та час очікування з використанням наборів даних. Таким чином, агенти забезпечують відповідний механізм для моделювання та моделювання управління трафіком коридорів, оскільки він пропонує персоналізований спосіб абстрагувати та описати кожен автономну структуру на індивідуальному рівні, аналізуючи вузькі місця та покращуючи операційну ефективність.

У моделюванні логістичних систем за агентурною моделлю та динамічними графіками [8] відтворюють шляхом моделювання поведінку логістичних систем, визначених як коридори, встановлені між портом шлюзу, куди імпортуються товари, та міськими районами, де розташовані кінцеві дистриб'ютори. Для проектування моделі була обрана модель, заснована на агентах, у поєднанні з динамічними графіками, яка показує, як макрошаблони виникають із мікро правил.

В агентській оптимізації контейнерних терміналів [7] описують емуляційну платформу контейнерного терміналу, що включає функціонування розподілених об'єктів (наприклад, набережні крани, транспортні носії). Це отримане за допомогою агентурної структури, разом з двома оптимізаціями, пов'язаними з переговорами та розподілом ресурсів.

При побудові агентських моделей контейнерних терміналів морського

порту [9], надали потужний інструмент, який оператори терміналів можуть використовувати для оцінки ефективності різних передбачуваних стратегій обслуговування кранів, а також ефекту від наявності додаткових кранів або меншої кількості кранів до механічних неполадок або планового технічного обслуговування.

Під час оцінки пропускної спроможності аеропорту за допомогою агентного моделювання, у роботі [10] запропонували засновану на агентах архітектуру для моделювання руху повітряних суден з метою прогнозування пропускної спроможності аеропорту та задоволення власності аеропорту.

В імітаційному агентно-орієнтованому дослідженні для поліпшення результатів роботи логістичного складу [11], провели агентне моделювання для аналізу поведінки автоматичних складських приміщень під впливом певних факторів, отримавши тим самим показники для підтримки прийняття рішень під час покращення продуктивності складу.

У роботі [12] агентне моделювання застосовується до проблеми аналізу операційної політики щодо перевалки контейнерів у портовому контейнерному терміналі. Менеджери, що беруть участь у перевантажувальних операціях, тобто керівник терміналу, капітан порту, портові вантажники та капітани кораблів, а також деякі оператори фізичних ресурсів, такі як крани та транспортні носії, моделюються як агенти. Потім порівнюються політики щодо послідовності розподілу.

У роботі [4] було запропоновано імітаційну модель для логістики лікарні, яка представляє підходящий метод для тестування різноманітних сценаріїв, що дозволить виявити шлях для вдосконалення якості доставки як ліків, так і пацієнтів.

У роботі [5] представлено структуру SimMobility Freight, яка є частиною SimMobility, багатомасштабної платформи для імітації міського транспорту на базі агентів. SimMobility Freight може повністю моделювати імітацію товарних контрактів, логістики та планування експлуатації транспортних засобів та рішення щодо паркування. Це дозволяє оцінити альтернативні логістичні рішення та виміряти їх вплив.

Для демонстрації його можливостей, проводиться аналіз норм строків доставки, оцінюючи вплив політики.

Робота [13] посвячена процесу валідації поведінки агентів шляхом оцінки процесів прийняття рішень агентами. Для цього пропонується система перевірки, заснована на симуляційній грі з участю. Використовуючи цю структуру, залучається людський гравця (тобто зацікавлена сторона домену), щоб дозволити зібрати інформацію про вибір та перевірити поведінку окремого агента. Для тестування фреймворку розроблена гра доказів концепції для міської логістичної АОМ.

У роботі [14] пропонується підхід на основі гібридного агента для планування та синхронізації парків логістики електронної комерції (ПЛЕК). Це досягається інтеграцією інтелектуальних центрів розподілу в середовищі електронної комерції. Запропонована платформа розроблена на основі агентурної технології, яка не тільки служить для децентралізації та синхронізації, але й оптимізована для транспортування та логістики загальної системи. Крім того, були розроблені механізми зв'язку на основі мобільних агентів між апаратними агентами та програмними агентами, а запропонована гібридна платформа на основі агента була впроваджена та протестована на основі конкретного дослідження. Після цього результати порівняли із звичайною системою, заснованою на чотирьох основних показниках.

У роботі [15] роботі представлена симуляційна модель, заснована на агентах, для кількісної оцінки ефектів міських логістичних схем на декількох учасників. Надається детальне математичне подання у формі процесу прийняття рішення Маркова. На основі великого літературного дослідження збираються дані для представлення різних суб'єктів типових західноєвропейських міст. Проводяться чисельні експерименти, щоб отримати уявлення про схеми міської логістики. Результати показують, що більшість схем дають значні екологічні покращення, але досягнення довгострокової фінансової життєздатності є складною для міських центрів консолідації. Також демонструється, що такі заходи, як субсидії та

обмеження доступу, не завжди дають передбачуваний ефект. В експерименті з беккастингу визначаються умови та схеми досягнення фінансово життєздатного центру консолідації міст.

У роботі [16] розглядається проблему співпраці між об'єктами логістики. Ця співпраця має на меті ефективно планувати, впроваджувати та контролювати потік послуг та товарів між пунктом походження та пунктом призначення, забезпечуючи при цьому безпечний, оптимізований та ефективний розподіл споживачам у будь-який час та в будь-якому місці. З цією метою пропонується рішення на основі багатьох агентів, де архітектура кожного агента включає одночасні індивідуальні та колективні структури Belief-Desire-Intension (BDI) для підтримки та збалансування власних та спільних цілей. Це рішення, яке адаптоване та застосовується до контексту розумної логістики, приділяє особливу увагу важливому питанню управління логістичним ризиком.

Робота [17] пропонує імітаційну модель, засновану на агентах, що вивчає розвиток коаліції з часом з урахуванням різних питань, пов'язаних з довірою. Впроваджуються різні ступені співпраці, правила розширення коаліції з новими членами, а також методологія розподілу заощаджень на основі Шеплі. Для розрахунку такої економії додатково інтегровані процедури рішення маршруту руху транспортних засобів. Це дає змогу детально дослідити наслідки горизонтального співробітництва як з економічної, так і з екологічної точки зору. Експериментальні результати підкреслюють, що можна досягти значної економії завдяки ступеню співпраці та питанням довіри, що вказують на найвищу важливість.

У дослідженні [18] розробляється прототип просторово заснованої на агентах моделі та використовує її для оцінки відхилення кінцевої продукції між попитом та попитом на товари гуманітарної допомоги в районах, що постраждали від стихійних явищ протягом певного часу. Модельований тут випадок - постачання рафінованої нафти в райони, постраждалі від землетрусу в Венчуані в Китаї 2008 року. Підхід фокусується на постачанні

та використовує соціально-просторові масиви даних, а також дані про транспорт та перероблену нафту. Дослідження виявляє, що такий підхід є життєздатним, незважаючи на обмеження загальнодоступних наборів даних про товари для надання допомоги в Китаї. Дослідження стверджує, що такий підхід може допомогти розробникам політики, які прагнуть виявити вразливості системи та підвищити стійкість у районах, схильних до катастроф.

Головна мета роботи [19] – оптимізувати діяльність, пов'язану з переміщенням міських товарів, та зменшити загальну соціальну вартість цих видів діяльності. Метою цього дослідження є розробка детального опису та розуміння руху порожніх контейнерів у Мельбурні з використанням підходу моделювання, призначеного для представлення складних систем. Цей документ досліджує причини та наслідки перестановки порожнього контейнера на суші. Потім ключові характеристики системи були зафіксовані в концептуальній моделі, яка документує та формулює логістичні процеси, взаємодії, владні відносини, проблеми та потенційні рішення вантажної системи. Концептуальна модель була перетворена на імітаційну модель у вигляді агентної моделі. Модель, заснована на агентах, була доказом концепції, яка надала можливість графічно імітувати логістичні процеси та взаємодії системи природним, гнучким та динамічним способом. Порожні контейнери становлять значну частку логістичної діяльності, однак попередні дослідження щодо контейнерних вантажних перевезень у Мельбурні, як правило, фокусувались на комерційно цінних завантажених контейнерах, а не на порожніх контейнерах. Ця стаття усуває прогалину в розумінні управління порожніми контейнерами в Мельбурні, як це представлено у концепт-моделі та моделі на основі агентів.

У роботі [20] представлено метод, заснований на агентах, який використовує навчання підкріплення для оцінки ефективності системи портових спільнот. Оцінюється метод, використовуючи двотижневі спостереження за імпортними контейнерами в порту Брісбена як тематичне

дослідження. Розглядаються три сценарії. Перший сценарій оцінює спостережену доставку контейнерів окремими лініями судноплавства та оцінює партії, розподілені різними автомобільними перевізниками, на основі оптимізації загальної вартості логістики окремого вантажовідправника. Другий сценарій передбачає, що в оптимальному випадку всі агенти (транспортні лінії та автомобільні перевізники) спілкуються та співпрацюють через єдиний портал. Мета співпраці полягає у спільному користуванні транспортними засобами та організації турів для доставки вантажів кільком імпортерам з метою зменшення загальних логістичних витрат, а також враховуються фізичні та часові обмеження. Третій сценарій дозволяє деяким агентам час від часу вирішувати діяти на основі індивідуальних витрат, а не загальних сукупних логістичних витрат. Результати цього дослідження вказують на підвищення ефективності всього логістичного процесу за рахунок співпраці, а дослідження забезпечує прототип системи портового співтовариства для підтримки логістичних рішень.

1.4 Постановка задачі

Метою даної роботи є розробка агентської моделі логістичних послуг в Харківській області.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- дослідити методи та моделі побудови агентно-орієнтованих систем;
- дослідити логістичну ситуацію в регіоні;
- розробити агентну модель для імітації деяких логістичних аспектів;
- проаналізувати результати, отримані під час моделювання.

2 АНАЛІЗ ІНСТРУМЕНТАРІЮ ДЛЯ АГЕНТНО-ОРІЄНТОВАНОГО МОДЕЛЮВАННЯ

2.1 Основні парадигми імітаційного моделювання

Люди часто недооцінюють, як багато вони використовують моделі. Насправді, ментальні моделі будуються людьми щодня, щоб зрозуміти та проаналізувати, як все працює. Однак, коли завдання ускладнюються, цифрові моделі стають набагато кращими за розумові моделі, і саме тому побудова моделей за допомогою комп'ютерних програмних засобів стала стандартом у бізнес та технічних процесах.

Очевидно, що найпопулярнішим підходом цифрового моделювання є використання електронних таблиць. Моделі електронних таблиць зазвичай використовуються для фінансового аналізу та інвестиційних досліджень, а також для бізнес-планування та використання визначених користувачем формул, електронні таблиці спроможні виконувати математичні, статистичні та організаційні перетворення на наборах даних. Електронні таблиці також допомагають представити дані в організованому вигляді, що може допомогти донести майбутні заплановані рішення. Однак, вони роблять зусилля вирішити складні проблеми, оскільки їм бракує можливості розглядати такі теми, як зворотний зв'язок, затримка, взаємозалежності та нелінійність. Просто, маючи справу зі складними системами, вони відстають від імітаційних моделей, які можна розглядати як набір правил (рівняння, блок-схеми, стан машини, стільникові автомати), що визначають, як система, що моделюється, зміниться в майбутньому, враховуючи її нинішній стан [32].

Імітаційне моделювання виділяється в межах модельних інструментів і технологій, оскільки воно дозволяє вводити динаміку в моделі. Кожна імітаційна модель розвивається з часом, дискретно або постійно змінюючи свій стан відповідно до набору заданих правил. Імітаційне моделювання є

дуже потужним інструментом, що широко використовується для вирішення численних бізнес-завдань.

Це здебільшого тому, що концептуально це практично і легко зрозуміти: не потрібно апріорі відгадувати систему, просто запусіть імітаційну модель і перегляньте результат, що спливе. Нарешті, анімація, що забезпечується імітаційним моделюванням, є значною перевагою перед статичними моделями електронних таблиць. Коли люди бачать динамічну інформаційну панель або анімовану модель, подібну до фільму чи комп'ютерної гри, де можна буде побачити, що система працює, як зазначено, легше візуально перевірити модель і зрозуміти кращі його результати.

На рисунку 2.1, зображений типовий життєвий цикл ІМ, що запропонований Брукс та Робінсоном. Цей життєвий цикл, як можна побачити, базується на п'яти ітераційних етапах:

- а) визначення проблем у реальному світі;
- б) концептуальне моделювання;
- в) комп'ютерне моделювання;
- г) перевірка та звіряння;
- д) рішення та розуміння.

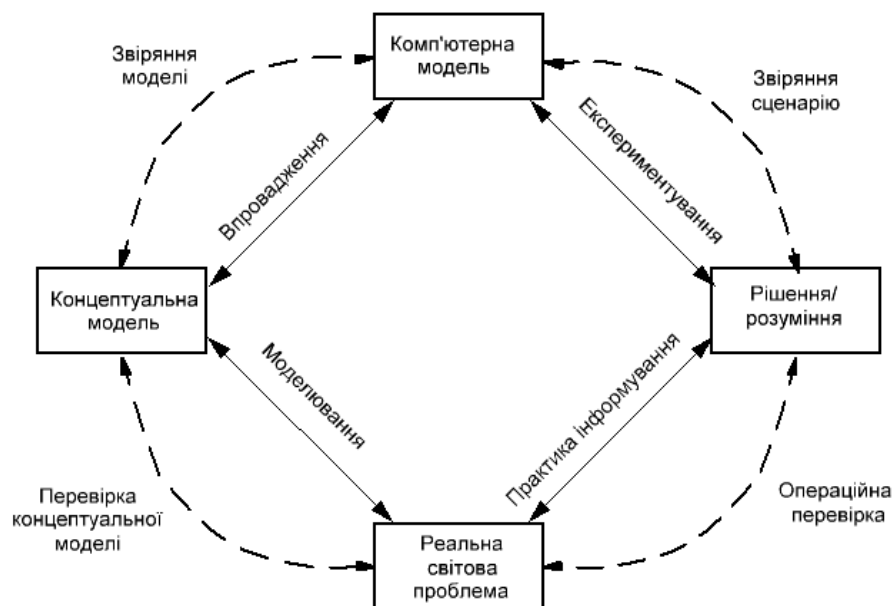


Рисунок 2.1 – Життєвий цикл імітаційного моделювання

Розглянемо різні підходи до імітаційного моделювання та визначено мультиметодне (також зване гібридним) моделювання. Переходячи від теорії до практики буде представлено програмне забезпечення AnyLogic. Нарешті, академічна література буде оглянута з подвійною метою: теорії, що лежать в основі програмних агентів та використання агентного моделювання для вирішення проблем у логістиці.

2.2 Методи імітаційного моделювання складних систем

Метод, в контексті ІМ, означає загальну основу для відображення реально-світової системи до своєї моделі. Метод пропонує тип мови, або "терміни і умови" для побудови моделі. У сучасному імітаційному моделюванні для складних систем існує три основні методи, кожен з яких обслуговує певний рівень абстракції (рисунок 2.2).

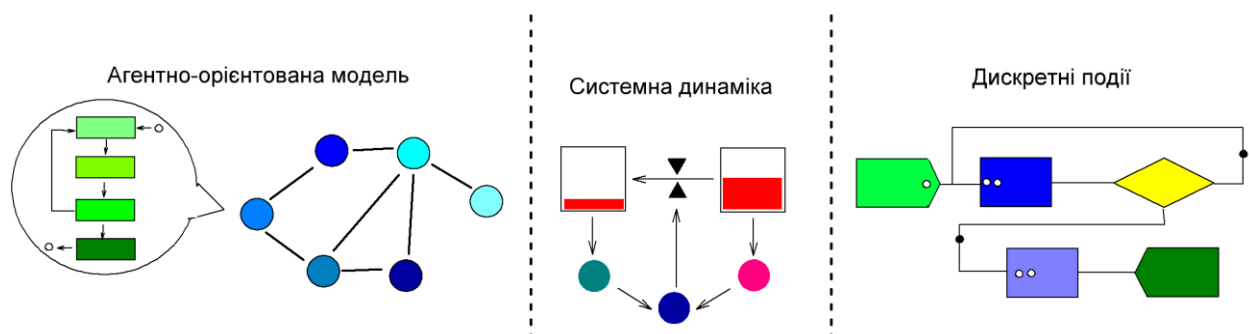


Рисунок 2.2 – Парадигми імітаційного моделювання

Системна динаміка - це найдавніший підхід до моделювання, коріння якого - у роботі, що виконана професором Массачусетського технологічного інституту Джейм Форестером в 1950-ті. З самого початку системна динаміка залишалася зосередженою на управлінському та організаційному рівнях бізнесу. Цей підхід діє на високому рівні абстракції і в основному використовується для стратегічного моделювання. Основним елементом є

діаграма запасів і потоків. Акція – це цифрове представлення чогось, що моделюється, наприклад, числа людей, набору вимог або ряду продуктів. Будь-який процес в системній динаміці моделюється як потік проміж запасів. Цей метод найчастіше використовується для моделювання стратегічного управління, маркетингу та макроекономічних проблем, екологічних та соціальних систем. Системна динаміка активно використовує поняття зворотного зв'язку і заснована на диференціальному рівнянні: при кожному моменті часу, потоки визначають проходження визначеної величини змінної від одного запасу до іншого.

Дискретно-подійний метод базується на підході, орієнтованому на процес, таким чином підтримуючи середній та середньо-низький рівень абстракції. У жовтні 1961 р. інженер IBM Джеффри Гордон представив першу версію GPSS (General Purpose Simulation System, originally Gordon's Programmable Simulation System), що вважається першою програмною реалізацією дискретно-подійного моделювання. Хоча моделювання дискретних подій майже так само старе, як і системна динаміка, але навіть у наші дні моделювання дискретних подій підтримується багатьма програмними засобами. Основною ідеєю ДПМ є розглядання системи як процесу, тобто, коли послідовність операцій виконується між сутностями. Модель вказується графічно як блок-схема процесу, де блоки представляють операції. Блок-схема зазвичай починається з "вихідних" блоків, які генерують сутності та вводять їх у процес, і закінчується блоками "зливу", які вилучають сутності з моделі. Цей тип діаграм відомий для ділового світу як діаграма процесу і є широко використовуваним для опису бізнес-процесів. Імітаційне ДПМ широко використовується у виробництві, системах обслуговування та галузі охорони здоров'я.

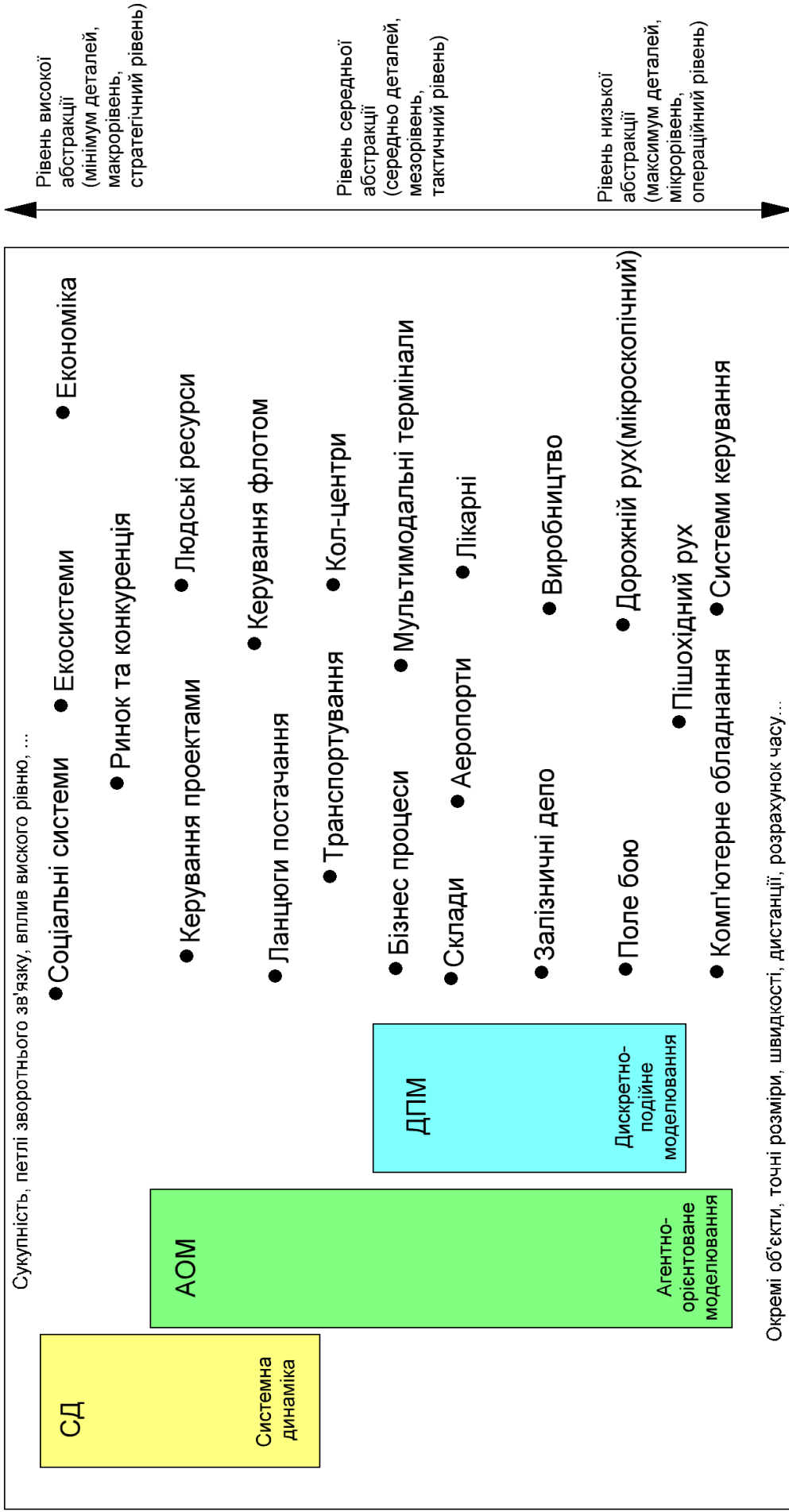
Моделювання на основі агента є найновішим з основних методів моделювання. У 1990-2000 рр. агентне моделювання залишалось суто академічною темою, але ХХІ століття, з його бумом серед можливостей обробки комп'ютерів, зробило агентне моделювання комерційно придатним

для вирішення масштабних ділових завдань. Більш того, порівняно з іншими підходами до моделювання, воно зараз демонструє найшвидший ріст використання. Агентне моделювання використовує підхід знизу вгору, де система описується як взаємодіючі об'єкти з власною поведінкою. Поведінка системи постає як підсумок індивідуальних дій агентів. Моделі, засновані на агентах, можуть варіюватися від самого детального, де агенти представляють фізичні об'єкти, до дуже абстрактних, де агенти є конкуруючими проектами або активами. Існують кілька конкретних галузей, де агентне моделювання особливо ефективно для вирішення проблем, такі як моделювання населення, пішоходів, дорожнього руху та моделювання епідемій. Але насправді агентне моделювання використовується для моделювання майже всього, від ринків, до ланцюгів поставок та логістики, у випадках, коли нам потрібно сфокусуватися на окремих об'єктах та описати їхні локальну поведінку та взаємодію.

2.2.1 Моделювання з різними рівнями абстракції

Побудова моделі вимагає рівня спрощення. Все залежить від системи, що моделюється і проблеми, яка потребує вирішення. Зображення нижче (рисунок 2.3) демонструє декілька програм моделювання, відсортованих за рівнем абстракції відповідної моделі.

Внизу знаходяться моделі фізичного рівня, що використовують високо деталізовані зображення реальних об'єктів. На цьому рівні у центрі уваги - фізичні взаємодії, розміри, швидкості, відстані, час. Моделі вгорі дуже абстрактні, і вони зазвичай використовують агрегати, а не окремі предмети, що дозволяє зрозуміти стосунки без необхідного моделювання проміжних етапів. Посередині розташовані агентні моделі, які можуть бути як дуже детальними, де агенти представляють фізичні об'єкти, так і дуже абстрактними, де агенти представляють цілі системи.



Окремі об'єкти, точні розміри, швидкості, дистанції, розрахунок часу...

Рисунок 2.3 – Моделювання з різними рівнями абстракції

Вибір правильного рівня абстракції має вирішальне значення для успіху моделі та іноді перегляд рівня абстракції може бути нормальним. У більшості випадків розробники починають з високого рівня абстракції, додаючи необхідні деталі, як тільки їм це потрібно, нехтуючи всім, що залишається нижче бажаного рівня абстракції. Використовуючи один метод, може бути важко провести моделювання на відповідному рівні абстракції. Можна моделювати дії автономних утворень за допомогою системної динаміки, але це не потрібно, коли інструменти на основі агента уникають необхідності додаткових абстракцій та припущень. Аналогічно, дискретні методи неефективні для моделювання безперервних змінних, коли доступні методи СД.

Іноді, в окремих випадках, може бути корисним моделювати окрему частину імітованої системи з різними методами – це зазвичай називають мультиметодною або гібридною моделлю [34]. Ідея багатометодового моделювання проста: легко інтегрувати різні методи моделювання та імітації, щоб подолати недоліки окремих підходів і отримати максимум від кожного. Поєднання різних методів призводить до ефективних і керованих моделей без використання обхідних шляхів. Історично першим комерційним програмним забезпеченням, що підтримує цей вид інтеграції було AnyLogic.

2.3 Моделювання в AnyLogic

Порівняно з системною динамікою та дискретно-подійним моделюванням, моделювання на основі агента є відносно новим методом, головним чином завдяки тому, що в основному залишався академічним до тих пір, поки практики моделювання не почали використовувати його на початку 2000-х.

Це було викликано трьома основними елементами. По-перше, бажанням глибше зрозуміти системи, які традиційні підходи погано фіксують. По-друге, вигоди технології моделювання, що стали можливими

завдяки комп'ютерній науці, наприклад, об'єктно-орієнтоване моделювання, UML (Unified Modeling Language) та діаграми будови. Останнє - швидке зростання потужності процесора та пам'яті поставили агентне моделювання вище, ніж системна динаміка та дискретно-подійні моделі.

AnyLogic історично розроблявся не моделістами симуляторів, а людьми з досвідом розподілених систем, теорії паралелізму та інформатики. Тому жодна з класичних парадигм імітаційного моделювання не використовувалася як фундамент. Натомість, підходи та мови, призначені для вирішення складності та прийняті у світі програмної інженерії. Виявилось, що схеми запасів і потоків та блок-схеми природно виражаються в об'єктно-орієнтованій основній мові AnyLogic, і тут є великі переваги навіть для класичних стилів моделювання: компактне структуроване представлення, гнучкі визначення даних та ін. Але найцікавіше - це здатність швидко складати індустріальні моделі, засновані на агентах, в одному візуальному середовищі. AnyLogic підтримує готові до використання конструкції для визначення поведінки агента, спілкування, модель середовища та має багаті можливості візуалізації [32].

З архітектурної точки зору, агенти є основними будівельними елементами AnyLogic моделі, але на практиці вони мало чим відрізняються від простих класів об'єктно-орієнтованого програмування. Насправді AnyLogic базується на програмуванні Java мовою, але інтуїтивно зрозумілий користувальницький інтерфейс дозволяє уникнути довгого написання блоків коду і поміщає необхідну інформацію в заздалегідь відформатовані коробки. Немає необхідності в професійних навичках розробки програмного забезпечення для створення хорошої моделі, а невеликі дії, такі як ініціалізація параметрів, надсилання повідомлень між агентами або переміщення агентів визначається додаванням декількох рядків кодових скриптів.

Типова модель, заснована на агентах в AnyLogic, має принаймні два типи агентів: Main тип для об'єкта верхнього рівня, який представляє

середовище та інший тип агента, що міститься всередині Main.

Діаграми стану є основним інструментом, що використовується для моделювання агентів у AnyLogic, оскільки вони є однією з найдосконаліших конструкцій, що використовуються для опису часової та станової поведінки. Точно так, як це було описано раніше для агентів, діаграма стану є адаптацією з автоматів UML, які, однак, послаблюють суворе визначення інформатики, а скоріше зосереджуються на практичному застосуванні ширшого спектру.

На рисунку 2.4 наведено діаграму стану AnyLogic.

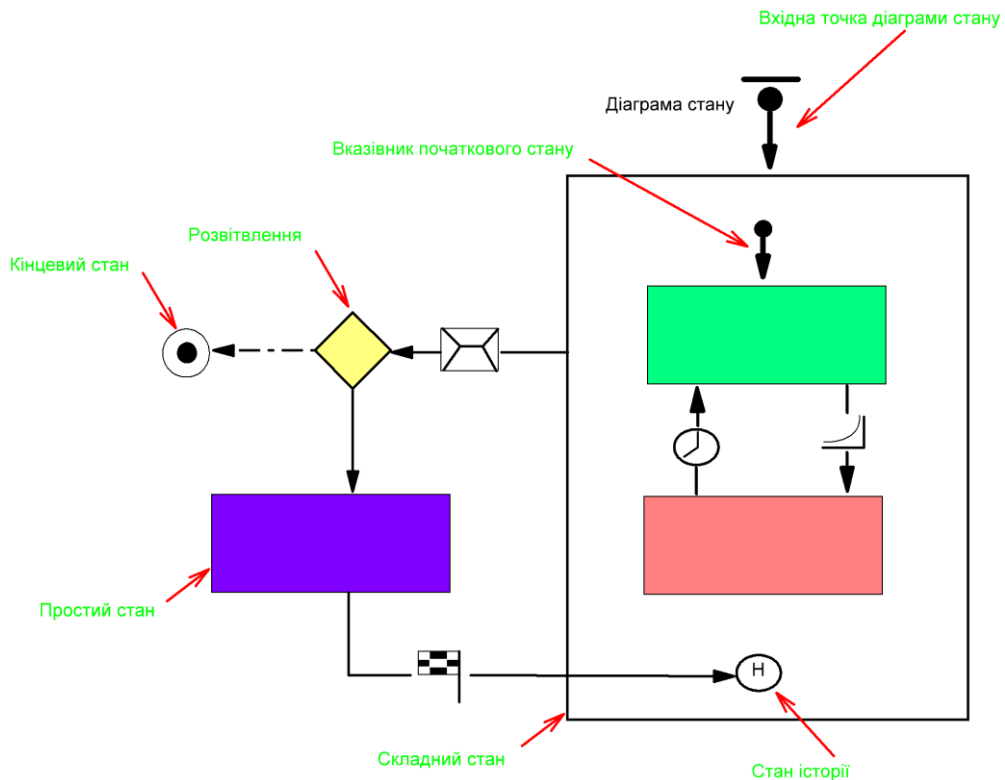


Рисунок 2.4 - Діаграма стану AnyLogic

Точка входу в діаграму стану позначає початковий стан і визначає ім'я діаграми. Загалом, агент може мати кілька діаграм стану, але, як правило, лише одна необхідна.

Стани в діаграмі є альтернативними, тому агент може бути лише в одному простому стані одночасно; однак прості стани можна включати до складених станів, які можуть мати ієрархічну структуру та бути включеними

до інших композицій вищого рівня станів. Для введення в складений стан потрібно використовувати початковий показник стану, оскільки агент завжди повинен перебувати в єдиному стані.

Переходи визначають, як агент змінює свої стани, отже, роблячи новий набір переходів активними. Перехід може бути спровокований різними типами подій, як зазначено нижче:

- час очікування, який говорить, що через визначений час інтервал відлічується з моменту, коли діаграма стану переходить у прямий стан. Зазвичай використовується для моделювання затримок;

- швидкість діє так само, як і час очікування, але інтервал часу береться за експоненціальним розподілом, параметризованим із заданою швидкістю;

- умова відстежує певний стан і реагує, коли воно стане істинним. Умова є довільним булевим виразом і може залежати від станів будь-яких агентів у цілій моделі з неперервною, а також дискретною динамікою;

- повідомлення, після отримання якого діаграмою стану або агентом ззовні перехід ініціюється. Шаблон повідомлення можна вказати у властивостях переходу і лише повідомлення, що підходить по шаблону, ініціює перехід;

- прибуття використовується лише для агента, що рухається, коли він прибуває до вказаного пункту призначення.

Існує особливий тип переходів, який називається внутрішнім переходом, при якому обидві початкова і кінцева точки переходу лежать на кордоні цього стану. Так як внутрішній перехід не виходить із стану, що оточує, ні дії виходу, ні входу не виконуються, коли здійснюється перехід. Тому внутрішній перехід дуже корисний для реалізації простих фонових завдань, які не повинні переривати основну діяльність складеного стану.

Гілки можна використовувати разом із переходами, які мають більше одного пункту призначення станів, відповідно до умови, що підлягає оцінці. Контроль ніколи не залишається у філіальному стані і завжди проходить

наскрізь. Тому тригери не можна вказати для переходів вихідних гілок та існування переходу за замовчуванням, прийнятого, якщо всі інші помилкові, завжди потрібно. Нарешті, остаточний стан являє собою точку припинення діаграми стану і за бажанням його можна використовувати для видалення агента із середовища за допомогою виклику `main.remove_agent (this)`.

Подія – це найпростіший спосіб запланувати якусь дію в моделі; таким чином, події зазвичай використовуються для моделювання затримок, тайм-аутів та перевірки стану. Події можуть активуватися трьома типами тригерів:

- час очікування, який використовується, коли потрібно запланувати дію на певний час або на дату. Подія, ініційована тайм-аутом, має більше особливостей (ви можете вказати, що термін її дії закінчується один раз, циклічно або повністю контролюється користувачем);

- стан використовується, коли потрібно відстежувати певний стан та виконати дію, коли ця умова стане істинною;

- тариф використовується для моделювання потоку незалежних подій пуассонівського розподілу. Часто потрібно моделювати прибуття за теорією черг.

Існує інший тип події, визначений іншим елементом моделі - динамічна подія – використовується для планування будь-якої кількості одночасних та незалежних подій. Модель може мати кілька екземплярів однієї і тієї ж динамічної події, запланованої одночасно, і вони можуть ініціалізуватися даними, які зберігаються в параметрах події.

Враховуючи той факт, що агенти є незалежними об'єктами, механізм зв'язку інформації стає можливим завдяки повідомленням. Таким чином, агент може надіслати повідомлення окремому агенту чи групі інших агентів, незалежно від того, чи є вони агентами того самого типу чи ні. Повідомлення може бути об'єктом будь-якого типу складності. Для реалізації цього механізму використовуються конкретні функції, такі як `sendToAll(msg)`, `sendToRandom(msg)` та `send(msg, agent)`.

3 РОЗРОБКА АГЕНТСЬКОЇ МОДЕЛІ ЛОГІСТИЧНИХ ПОСЛУГ В ХАРКІВСЬКІЙ ОБЛАСТІ

3.1 Методологія моделювання складних реальних систем

Системна динаміка та агентне моделювання – дві широко використовувані методології при моделюванні складних реальних систем, що характеризуються динамічними нелінійними стосунками. Навіть якщо внутрішня структура різниться, обидва мають на меті збільшення розуміння системи та сценаріїв тестування з метою підтримки прийняття рішень та реалізації політики. Будівельними блоками в описуваній моделі системної динаміки є запаси, потоки та допоміжні змінні. Запаси представляють накопичення матеріалу та інформації, викликані дією припливів і відтоків. Запаси математично описуються інтегральними рівняннями, тоді як потоки описуються диференціальними рівняннями. Рішення цих наборів рівнянь описує агрегований стан системи, який постійно змінюється з часом і залежить від попереднього стану системи.

Навпаки, основними будівельними блоками АОМ є автономні агенти, їх правила та дії, а також середовище, в якому вони взаємодіють. Хоча правила прийняття рішень агентами, як правило, регулюють поведінку агентів для досягнення особистих переваг, колективний інтелект може також з'явитися, коли агенти координують свої рішення для досягнення спільних цілей. Тому, аналізуючи виключно внутрішній механізм дії агентів, не можливо пояснити спостереження на макрорівні. Більше того, прийняття рішень агентами, як правило, базується на обмежених спостережуваних знаннях, ніж на повному знанні всього стану системи.

Взагалі кажучи, можна отримати відповідну модель, засновану на агентах, розпочавши з СД моделі, оскільки ці два варіанти не є взаємовиключними парадигмами. Тоді модель, засновану на агентах, може

бути додатково вдосконалено, щоб охопити набагато складніші поведінку, залежності та взаємодії, тим самим забезпечуючи глибше розуміння системи, що моделюється.

Порівняно з моделями СД, нема такого місця в АОМ, де визначалася б поведінка системи або глобальна динаміка. Натомість, людина, що моделює, визначає поведінку на індивідуальному рівні, а загальна поведінка виникає в результаті багатьох індивідуальностей, кожна з яких дотримується власних правил поведінки, співіснують в якомусь середовищі та спілкуються між собою та з навколишнім середовищем. Отже, агентське моделювання - це більш ментальність ніж технології.

Природною перевагою є те, що АОМ передбачає будівництво моделі за відсутності знань про глобальні взаємозалежності: одна може нічого або дуже мало знати про те, як речі сукупно впливають один на одного на загальному рівні, або яка загальна послідовність операцій і тому подібне. Але, якщо існує сприйняття того, як поведуться окремі учасники процесу, можливо побудувати модель, засновану на агентах, а потім отримати глобальну поведінку. Таким чином, навіть якщо існує, скажімо, модель системної динаміки, яка відповідає на питання, може буде значно простіше побудувати модель, засновану на агентах (рисунок 3.1).

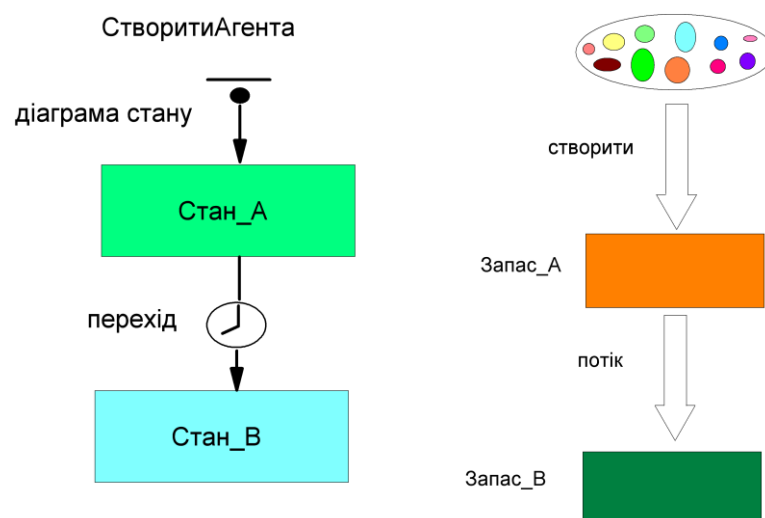


Рисунок 3.1 – Перетворення СД в АОМ

Ключовою відправною точкою є "дезагрегація" запасів (СД), дивлячись на них так, ніби це не ємності з рідиною, а коробки з кульками. Ці кульки стануть агентами, перебуваючи в станах, що відповідають вихідним запасам, в яких вони були; отже, потоки (СД) будуть перетворені на переходи. Момент часу, в який перехід відбуватиметься, інтуїтивно залежить від вихідної швидкості потоку i , отже, може бути промодельований шляхом подачі переходу з відповідними параметрами. Нарешті, джерела (СД) можна замінити подією, яка створює нового агента відповідно до часу очікування або передбаченою оцінкою.

3.1.1 Модель дифузії Басса

Модель дифузії Баса (рисунок 3.2), класична модель підручника для дифузії продуктів, заснована на нинішніх та потенційних усиновлювачах, пропонується як тривіальний приклад згаданого перетворення [25].

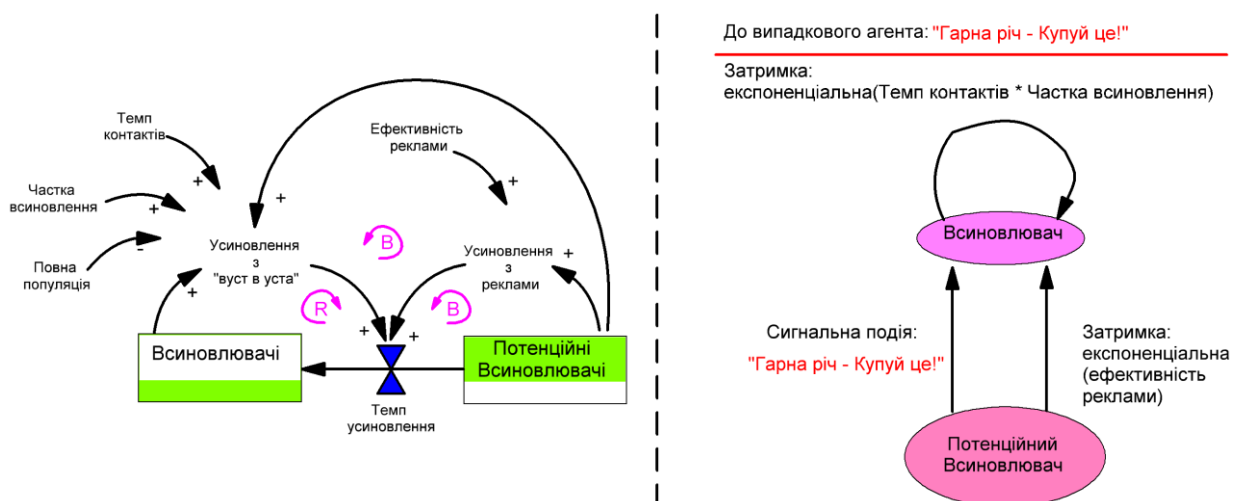


Рисунок 3.2 – Модель дифузії Басса

Як показано на рисунку, агент може перебувати одночасно у стані як «потенційного усиновлювача», так і в стані «усиновлювача», слідуючи відповідним запасам у моделі СД. Два виклики, що впливають на рівень усиновлення, а саме усиновлення з реклами та усиновлення з вуст в уста,

перекладаються на два окремі переходи. Перший просто моделюється як перехід ставки, другий як перехід повідомлення. Фактично, як тільки агент стає усиновлювачем, він починає надсилати повідомлення іншим агентам, у свою чергу, ініціюючи перехід, якщо потенційний усиновлювач отримує це повідомлення. Отримані результати, тобто добре відома S-форма, стає фактично неможливо відрізнити за достатньо великої кількості задіяних агентів. Однак розробка АОМ є більш природною і прямолінійною, ніж складання системи диференціальних рівнянь, зокрема, якщо немає попереднього знання про всю поведінку системи.

Моделі, засновані на агентах, також, як правило, простіше підтримувати, оскільки уточнення моделей призводить до дуже локальних, а не глобальних змін. Наприклад, припустимо, що вплив ВВУ на конкретну людину залежить від того, як давно він придбав товар, так що людина активно його рекламує незабаром після покупки і менш часто через деякий час. Для охоплення цього явища лише невелика модифікація потрібна: введення змінної, яка відстежує час придбання та його введення в перехід для надсилання повідомлення. Чи можна зафіксувати таку поведінку в СД? Внесок ВВУ у коефіцієнт усиновлення тепер може бути різним для будь-яких двох усиновлювачів, і це також змінюється з часом для усиновлювача. Отже, агрегування усиновлювачів, що приймають одну (або будь-яку розумну кінцеву кількість) акцію, спотворюють результати.

СД вже давно стикалася з подібними проблемами і навіть запропоновано певне часткове рішення, точніше – обхідний шлях: запас, що містить об'єкти з досить різними властивостями, розкладається на масив сегментів, а об'єкти переміщуються між сегментами, коли їхні властивості змінюються. Однак, розглянемо об'єкти, що мають кілька таких властивостей. Масив сегментів зростає як добуток розмірів i , через кілька кроків, кількість секцій у масиві може легко перевищувати кількість окремих об'єктів у реальному світі [26]. Це робить таку модель СД безглуздою і повільною, тоді як АОМ завжди міститиме необхідну кількість об'єктів.

3.1.2 Модель «Хижак-жертва»

Крім того, в деяких випадках СД навіть не може надати рішення, коли мають справу з певними проблемами, як на прикладі іншого відомого випадку: модель хижак-жертва. На основі пари диференціальних рівнянь, розробленої Лоткою і Вольтерром, модель характеризується коливаннями чисельності популяції як хижаків, так і жертв, при цьому пік коливань хижака трохи відстає за піком коливань жертви. Однак лише за допомогою АОМ можна подолати деяке припущення, яке повинно бути в оригінальній СДМ, вводячи таким чином змінні для обліку часу та простору. Щодо першого, використовуючи моделювання на основі агента, можна надати зайцю та рисі тривалість життя, так що вони тепер вмирають через вік, з'їдання (заєць) або голод (рись). Щодо останнього, натомість, заєць і рись тепер знають про простір, так що можна обмежити щільність для народження і пересування зайця, і обмежити відстань і мисливські рухи рисі. На відміну від СД, залежно від параметрів моделі, рись або зайці можуть повністю вимерти, що ніколи не трапляється в моделі СД завдяки її постійному характеру. В АОМ також можна переглянути 2D-зображення та простежити рух одного зайця, рисі, їхніх сімей або поколінь та передову аналітику.

Як щойно продемонстрували, загалом за допомогою агентного моделювання можливо фіксувати більше явищ у реальному житті та корисних знань, ніж за допомогою підходу з СД. Однак це не означає автоматично, що АОМ - остаточна заміна системної динаміки, як і не означає, що завжди є відповідне рішення, як у випадку, показаному в наступному параграфі.

3.2 Напрямки розвитку агенто-орієнтованого моделювання

AnyLogic повністю здатний мати справу з СД та моделлю, заснованою на агентах, і навіть цілісно інтегрувати ці два підходи в одне середовище. Він

також дозволяє автоматично імпортувати та конвертувати модель Vensim®, щоб легко продовжувати розвивати її. Однак інструмент для прямого перетворення моделі від СД до АО не доступний у програмі AnyLogic, а також як заявка третьої сторони. Насправді, перетворення моделі не завжди є простим процесом і передбачає два питання різного роду: одне технічного плану та друге концептуального.

Стосовно технічного питання, оскільки при АОМ процес трапляється з підходом знизу вгору, індивідуальні дані повинні бути надані агентам, щоб змусити глобальну динаміку впливати з окремих взаємодій. Тому, цей процес може працювати в цьому, але не у зворотному напрямку: як системна динаміка має справу лише з сукупностями, витяг окремих даних неможливий, оскільки запас вмісту не відрізняється від одного до іншого. Перехід з СД до відповідної моделі, заснованої на агентах, означає - загалом - підвищення рівня деталізації, що вимагає вищої деталізації доступних даних. За відсутності даних виникає проблема заповнення порожнього місця, і лише усереднення або розрахунковий розподіл ймовірності можуть бути використані, що, ймовірно, призведе до отримання висновків з сумнівною доданою вартістю.

Щодо концептуального питання, слід зазначити, що процедура реконцептуалізація має сенс лише у випадку вдосконалення агентів, що стосується захоплення їх індивідуальності. Якщо елементи, що зберігаються в моделі СД, є природно пасивний і невідмінний, мабуть, користі від цього перетворення їх на агентів буде не багато. Яскравим прикладом можуть бути запаси грошей: зазвичай немає інтересу до історії окремого долара, а долари не виявляють жодної активної поведінки. Отже, з практичної точки зору, процедура реконцептуалізації відбувається відповідно до бачення та досвіду моделіста, що зараз переформулює систему за допомогою підходу знизу вгору. Таким чином, використовуючи свою інтуїцію, розробник також шукає, які аспекти виділити, щоб переслідувати бажану мету із системою та наявними даними.

Перетворення з СД на модель, засновану на агентах є нетривіальним. Насправді модель представляє деякі критичні моменти, які ускладнюють її можливість бути перетвореною прямолінійно.

Перш за все, замість того, щоб складатися з безперервного потоку даних з одного фонду до іншого, вона складається з взаємодії безлічі параметрів, таких як допоміжні та константи. Звичайно, це означає, що у разі переходу на агентну модель, відповідні агенти не мали б відповідних станів або спроможності робити переходи, а скоріше просто включатимуть деякі математичні розрахунки.

Незважаючи на існуючі проблеми, неправильно стверджувати, що модель не може бути переведена у АОМ. Насправді є можливим заново концептуалізувати модель із використанням підходу знизу вгору та зосередитись на ресурсі, що становить обмеження ємності. Таким чином, глобальна динаміка впливатиме з індивідуальної взаємодії, з масштабом відтворення функціонування оригінальної моделі, поки одночасно додаючи деталі на нижчому рівні.

3.3 Агентна модель логістичних послуг

Динаміку розвитку логістики Харкова, методологію агентного моделювання, можливості та проблеми, що виникають у результаті порівняння з системною динамікою та її можливості перетворимо в модель, засновану на агентах. Під час процесу розробки були зроблені деякі спрощення та припущення, щоб отримати ефективну модель складної логістичної системи. Таким чином також можна отримати модель, що може бути застосовано не тільки у справі Харкова, а навіть до інших регіональних логістичних центрів у світі, за умови достатньої кількості даних, які можливо подавати в модель.

Як уже зазначалося, реконцептуалізація попередньої моделі складається із зосередження уваги на ресурсах, що створюють вузьке місце.

Ці ресурси, які тепер моделюються як агенти, будуть взаємодіяти між собою, таким чином генеруючи обмеження пропускну́ї здатності всієї системи. Крім того, вхідні замовлення, виражені як TEU для залізної дороги та тонни для автомобільних перевезень, також будуть змодельовані як агенти. Моделювання на практиці у професіональному середовищі дозволяє розглядати також пасивні об'єкти як агенти, щоб взяти переваги потужних інструментів, що надаються програмним забезпеченням для відстеження індивідуальності та взаємодії з іншими агентами.

Базову основу моделі складає за замовчуванням головне середовище, в якому всі інші агенти живуть. Відповідно до об'єктно-орієнтованого програмування, логіка при створенні агентів наступна:

1. Тип агентів – клас, у якому визначено характеристики.
2. Популяція агентів [..] – список, в якому згруповані агенти.
3. Агент – сам об'єкт.

Оскільки екосистеми залізниць та автотранспорту змодельовано дзеркально, просто для побудови моделі потрібні такі типи агентів: Order, Transport, Handling, Warehouse, Link. Потім ці типи використовуються для створення популяцій агентів для залізниці та автотранспорту, як узагальнено в таблиці нижче (таблиця 3.1).

Таблиця 3.1 – Популяції агентів

Тип агента	Популяція агентів		
	Залізниця	Автотранспорт	Зв'язок
Order	teus[..]	tons[-
Transport	train[..]	truck[..]	-
Handling	handlings_rail_in[..]	handlings_motor_in[..]	-
	handlings_rail_out[..]	handlings_motor_out[..]	-
Warehouse	warehouses_rail[..]	warehouses_motor[..]	-
Link	-	-	links[..]

3.3.1 Модель Main

У середині AnyLogic вищезазначені групи розміщені в Main середовищі і представлені червоним символом (рисунок 3.3).

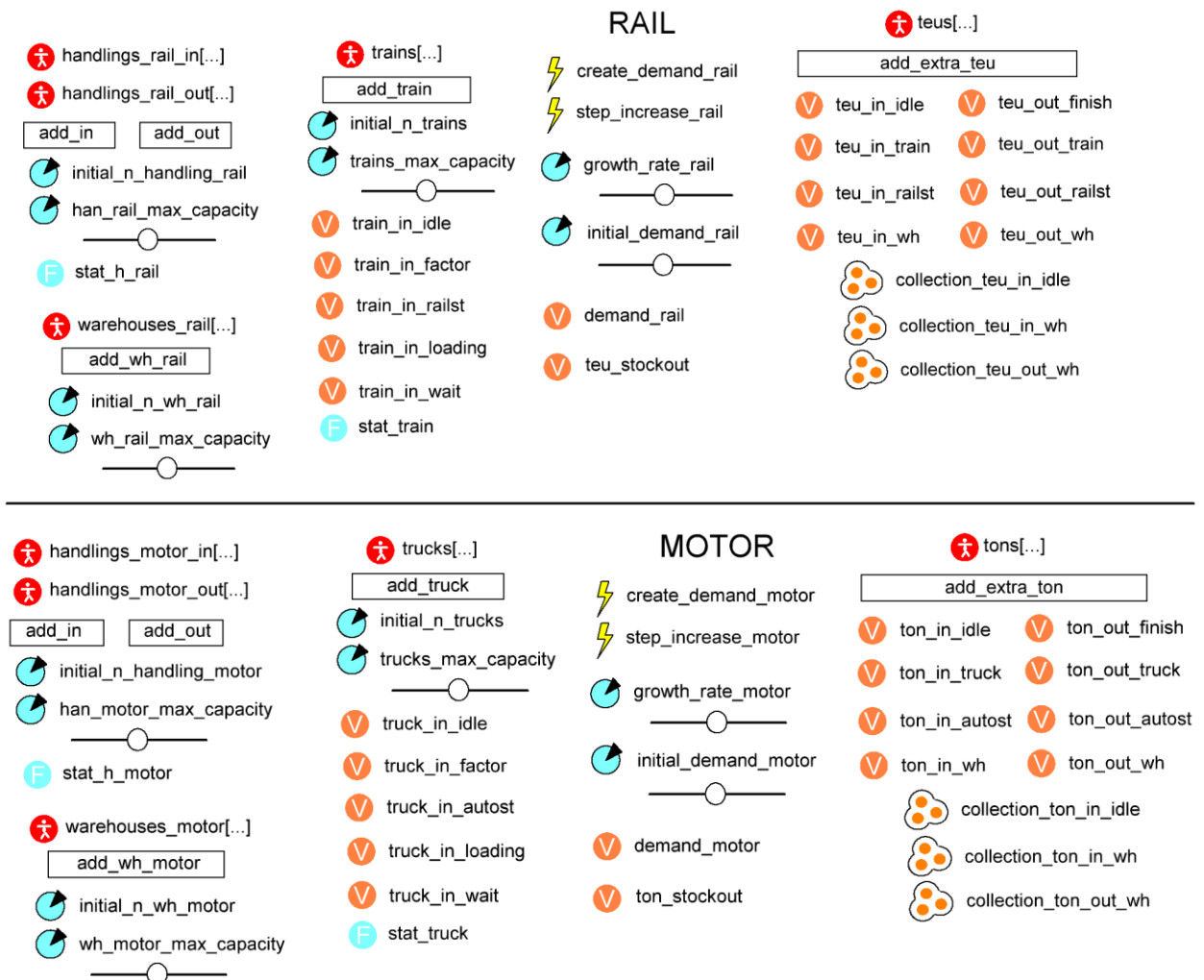


Рисунок 3.3 – Модель: Main

Повзунки та кнопки призначені виконувати варіації параметрів, щоб в реальному часі побачити зміни в системній поведінці. Усі інші елементи, тобто змінні, параметри, колекції, функції, події використовуються для подачі даних до системи, для аналізу статистики та з ціллю відлагодження. Принципова частина системи представлена блоком Rail(та Motor), який відповідає за генерацію замовлень, якими обробляються всі інші ресурси в системі.

Все починається з подій `create_demand_rail` та `create_demand_motor`, котрі генерують `teus` і тонни за допомогою циклічного таймера, відповідно до кількості, що задається за допомогою змінних `demand_rail` та `demand_motor`. Вони йдуть в парі з подією `step_increase_rail` та `step_increase_motor`, які використовуються з метою періодичного збільшення попиту, використовуючи щорічно складені темпи зростання.

3.3.2 Блок замовлення Order

Замовлення (рисунок 3.4) - це основний блок, на якому побудована вся модель. Діаграма стану використовується для представлення стану, в якому замовлення фактично знаходиться. Замовлення, слідуючи стрілкам, що описують процес, мож бути фізично розташовані в транспорті, у вантажно-розвантажувальних роботах, на складі, або у вхідних чи вихідних каналах.

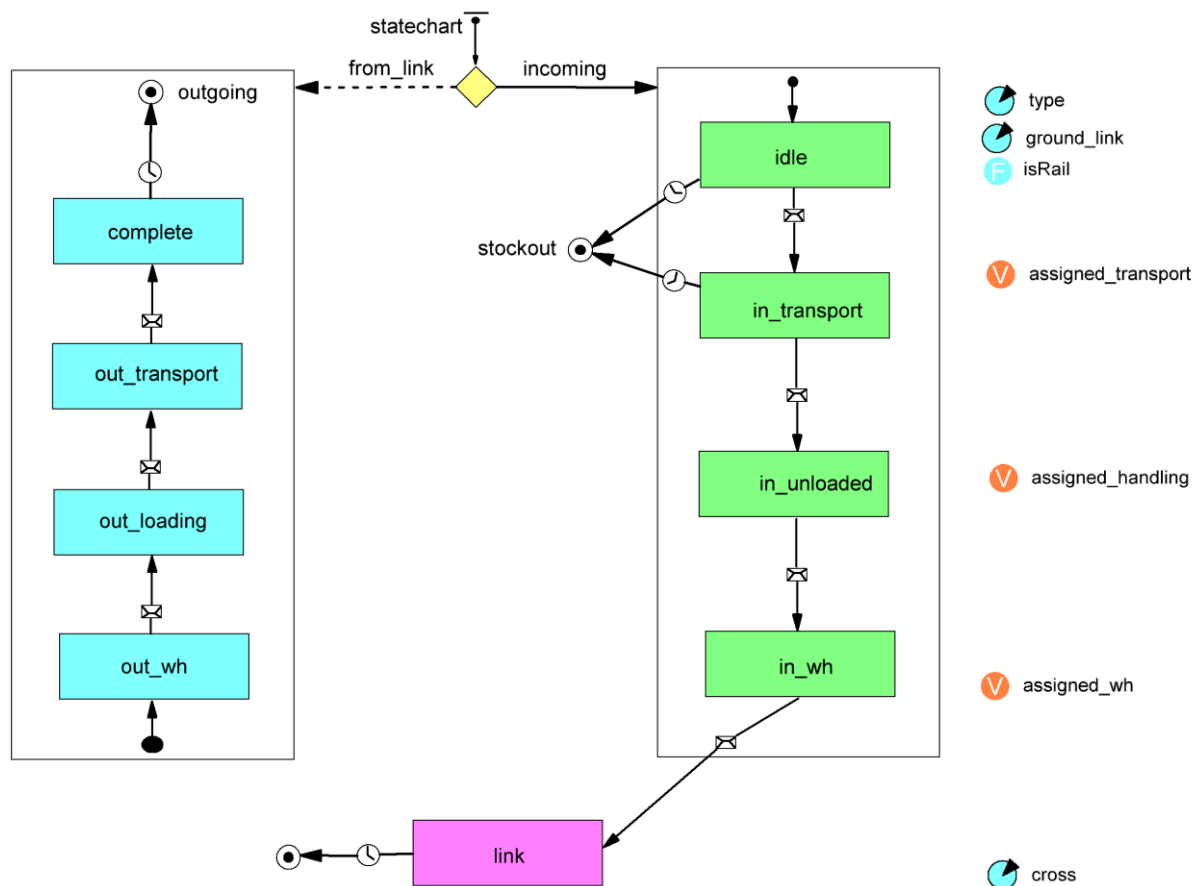


Рисунок 3.4 – Модель: Order

Крім того, є два віртуальних місця розташування, `idle` та `complete`, що використовуються. Враховуючи той факт, що тип `Order` використовується як для `teus`, так і для тонн, `type` (параметр) та `isRail` (функція) використовуються разом для розрізнення обох.

Процес починається зі створення замовлення, яке входить у діаграму стану вище. На даний момент замовлення, що надходять за посиланням, не розглядаються, отже, пунктирна лінія (`from_link`) не дотримується; також переходом з `in_wh` на даний момент `link` нехтують.

Як тільки замовлення створюється, то переходить в режим очікування, воно додається до колекції (приклад 3.1) і чекає, поки транспортний агент зможе його «зловити».

```
if (isRail ())
main.teu_in_idle ++;
main.collection_teu_in_idle.add ( this );
} else {
main.ton_in_idle ++;
main.collection_ton_in_idle.add ( this );
}
```

Приклад 3.1 – Замовлення, стан очікування, дія введення

Якщо замовлення залишається в режимі очікування занадто довго, замовлення втрачається, воно усувається з системи і враховується як запасна одиниця для цілей статистики. Це однозначно означає, що транспортна спроможність поступається попиту на неї і тому потребуватиме оновлення. В іншому випадку, якщо доступний один транспортний агент, повідомлення надсилається з транспортного агента до агента замовлення та відповідного переходу повідомлення, з `idle` в `in_transport`, спрацьовує (символ конверта).

Після прибуття транспортного агента на залізничну станцію або автостанцію настала черга обробки ресурсів для обробки замовлення, що відбувається у два етапи: по-перше вивантаження транспорту, по-друге перенесення замовлення на склад. Через деякий час, проведений на складі, замовлення йде в протилежному напрямку (ліва частина діаграми):

починаючи зі складу, вона переміщується до маніпуляції ресурсами і завантажується в транспорт, який зараз залишає Харків. Отже, порядок завершено, і після таймауту воно виключається із системи.

Протягом усього процесу можна відстежити, який агент є замовленням, призначеним в кожній фазі, яка реалізується зі змінними `assign_transport`, `assign_handlings` і `assign_wh`. Це особливість, яка відрізняє агентне моделювання від системної динаміки, оскільки остання має справу лише з сукупностями і унеможливує розрізнення одиниць і отримання детальної аналітики.

3.3.3 Модель Transport

Транспорт являє собою вантажні потяги та вантажівки в системі. Як було зроблено в випадку для `Order`, `type`(параметр) та `isRail`(функція) використовуються разом для розрізнення обох видів при виконанні команд. Ідея, яка стоїть за цим типом агента, це наступна, згідно з діаграмою стану на зображенні (рисунок 3.5):

- `idle`: транспорт чекає замовлення, щоб його зловити, шукаючи його за допомогою внутрішнього переходу;
- `demand_in`: повторення процесу, поки не досягнута `max_capacity`, потім подорож до залізниці чи автостанції;
- `arrived`: після прибуття транспорт очікує розвантаження агентами `handlings_in`;
- `loading_out`: після повного розвантаження транспорт очікує завантаження агентами `handlings_out`;
- `wait`: нарешті, транспорт залишає залізничну станцію автостанцію і очікує скидання циклічного таймера події.

Останній стан має вирішальне значення, оскільки він використовується для скидання пропускнуої спроможності транспорту. Насправді це робиться для того, щоб транспортний агент міг завантажити лише певну кількість замовлень у визначені користувачем часові рамки, таким чином створюючи

загальну пропускну здатність на сукупному рівні. Таймер події запускається, як тільки перше замовлення вловлюється, дозволяючи не тільки точне відстеження часу, але і виявлення недостатнього використання ресурсів.

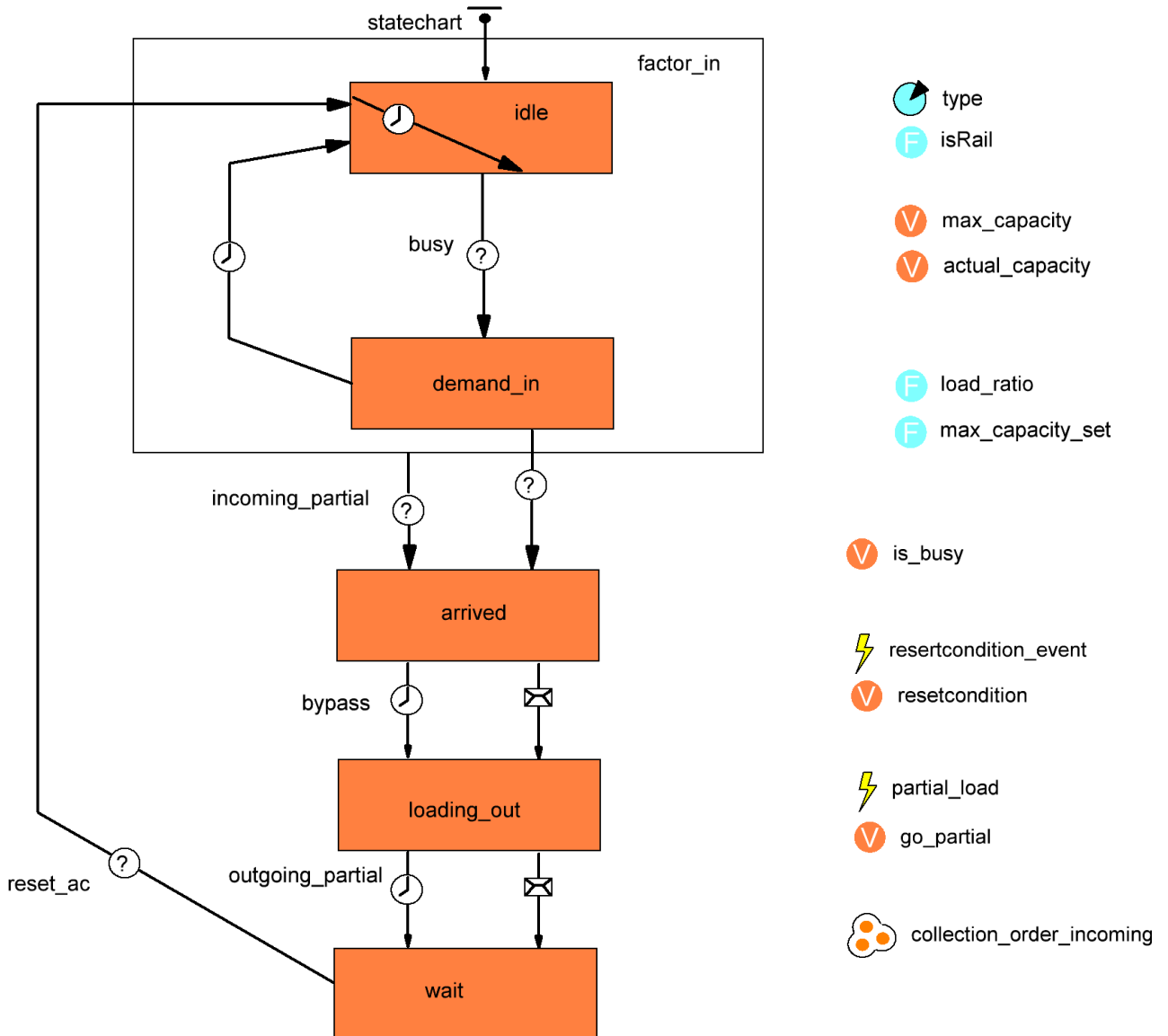


Рисунок 3.5 – Модель: Transport

Кількість визначається змінною `max_capacity`, яка використовується в поєднанні з `actual_capacity` для декількох різних цілей. До них належать оцінка того, чи транспорт повністю завантажений/розвантажений (для спрацьовування переходів) і для відстеження статистики.

Перебуваючи в стані `idle`, транспорт періодично викликає внутрішній

перехід ловити попит. Процес складається з наступних рядків коду (приклад 3.2) і коротко описаний після.

```

if(
  actual_capacity <msx_capacity &&
  (isRail ()
  ? ! main.collection_teu_in_idle.isEmpty ()
  :! Main.collection_ton_in_idle.isEmpty ())
  ) {
  Order t = isRail ()
  ? main.collection_teu_in_idle.pop ()
  : main.collection_ton_in_idle.pop ();

  t.assigned_transport = this ;
  collection_order_incoming.add (t);
  deliver ( "demand", t);

  actual_capacity ++;
  is_busy = true ;
  part_load.restart ();
  event.restart ();
}

```

Приклад 3.2 – Транспорт, стан idle, внутрішній перехід

Оператор `if` перешкоджає виконанню коду у випадку досягнення максимальної ємності (щоб уникнути перевищення вказаної максимальної потужності) або немає замовлення для виловлення (потрібна перевірка, щоб уникнути появи помилки). `isRail` використовується спільно з тернарним оператором «?» і повертає рядок 4, якщо `true`, рядок 5 – в іншому випадку. Отже, з простим виразом, можна виконувати операції з різним набором колекцій, `teus` та тонн, лише з одним типом агента. Якщо умови дотримані, замовлення для виловлювання береться з колекції (рядки 8 або 9), агент поточного транспорту зберігається всередині замовлення (рядок 11), замовлення реєструється на транспорті (рядок 12) і надсилається повідомлення, щоб викликати перехід від стану `idle` до `in_transport` стану всередині вибраного замовлення (рядок 13). Потім фактична потужність збільшується і для змінної встановлено значення `true`, щоб викликати умовний перехід. Крім того, активовано дві відкладені події: перша використовується для відправлення транспорту до місця призначення, навіть якщо він не повністю завантажений, щоб не затримувати замовлення

(`incoming_partial`), друга використовується для скидання циклу (`reset_ac`) не раніше зазначеного періоду, що запобігає перевищенню визначеної максимальної потужності транспорту.

Будучи простим процесом, спочатку може виглядати дивно, що є пара переходів за кожен змінюваний стан, а не тільки один. Причина в тому оскільки перший набір переходів (з лівого боку) пов'язаний із стандартним процесом, тоді як другий (з правого боку) використовується в якості обхідного рішення у випадку, якщо потрібно.

- `incoming_partial` вже описано та використовується для уникнення непотрібних затримок, якщо місткість потяга значно більша, ніж попит.

- `outgoing_partial` слідує тій же логіці, що і вище, але в протилежному напрямку. Більше того, це дозволяє впорядковане виконання `reset_ac`, щоб не зміщувати максимальну визначену ємність

- `bypass`, як нагадує слово, є тимчасовим обхідним шляхом для дуже особливої ситуації, яка може спричинити блокування загального процесу: це може траплятися рідко, залежно від стохастичного характеру системи, коли ємність складу є причиною вузького місця та попит перевищує потужність на тривалий час. Транспорт може закінчитися, застрягнувши в `arrived` стані, очікуючи вивантаження `handlings` агентами, що, на жаль, чекають вільного місця на складі. Отже, транспорт застряг і не може бути заповнений вихідними замовленнями, які можуть в свою чергу звільнити трохи місця на складі. `Bypass` дозволяє розпізнати цю ситуацію і розблокувати його, звільнивши транспорт від замовлень. Обхідний шлях залишається як відкрита точка для майбутнього розвитку, але був правильно перевірена і не спотворює нормальну роботу системи, оскільки замовлення, що розвантажуються таким чином, усуваються із системи, якщо не піднімаються агентами `handlings` швидко.

3.3.4 Модель Handling

Перевантажувальні ресурси, модель яких представлена на зображенні вище (рисунок 3.6), служать посередниками між транспортом і складами, як для вхідних, так і для вихідних замовлень.

Що стосується Transport та Order, type (параметр) та isRail (функція) використовуються разом для розрізнення ресурсів, що належать залізниці та автостанціям. in_out (параметр) та isIn(функція), з тією ж логікою, використовуються для розрізнення ресурсів, що обробляють вхідні та вихідні замовлення.

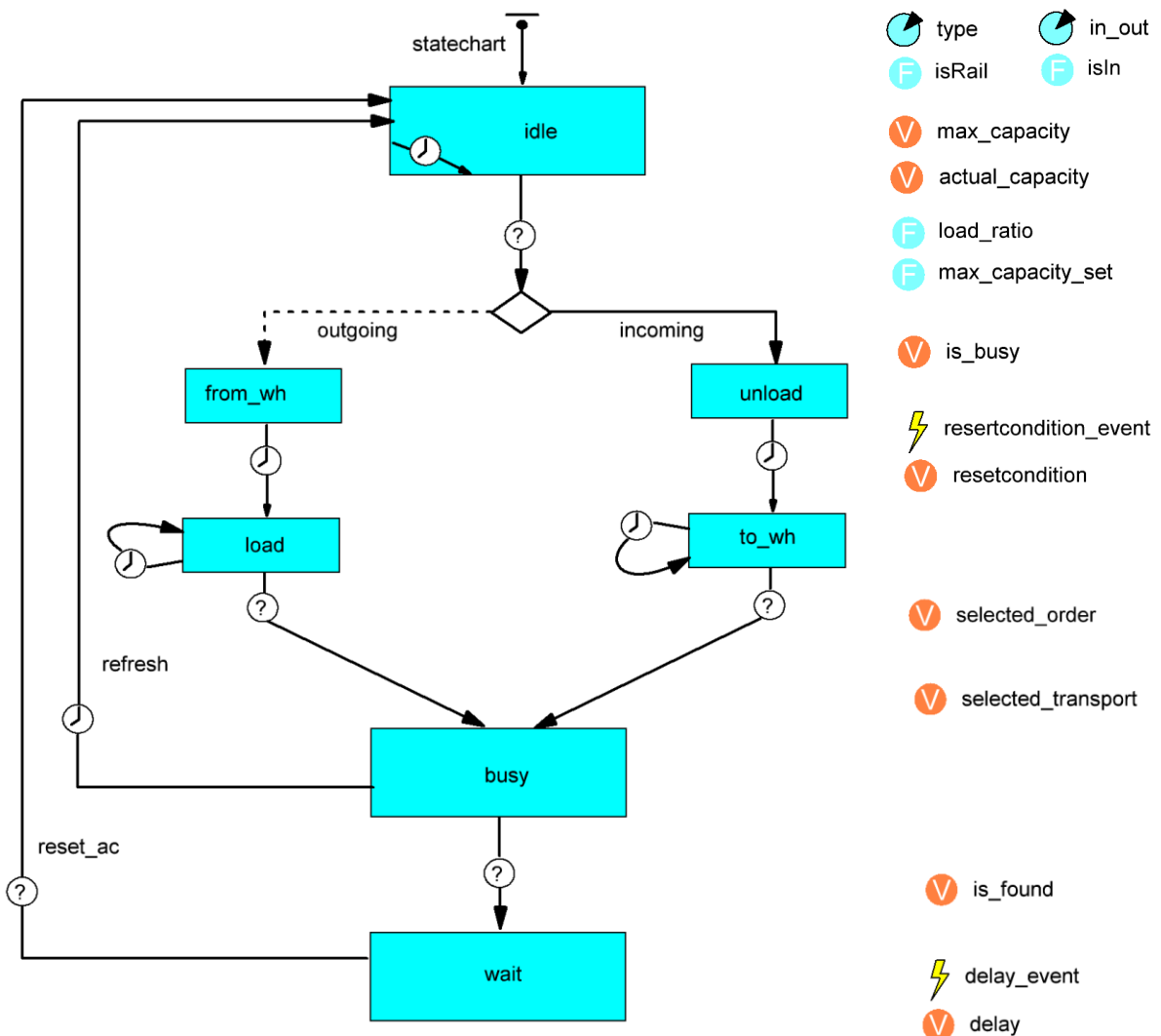


Рисунок 3.6 – Модель: Handling

Таким чином можна отримати 4 популяції агентів використовуючи лише один тип агента: `handlings_rail_in`, `handling_rail_out`, `warehouse_rail`, `handlings_motor_in`, `handling_motor_out`. Як це сталося для `bypass`, різниця між входом і виходом є практично обхідний шлях, оскільки це був найпростіший спосіб уникнути можливих перерв у процесі у разі надлишкового попиту та використання ресурсів, що представляють вузьке місце. Це рішення залишається відкритою точкою для майбутніх удосконалень, але воно вже було успішно протестовано і не змінює загальної продуктивності системи, дозволяючи точно дотримуватися обмеження, заданого через `max_capacity`.

Ітераційний процес, який виконується кожним агентом обробки, описаний у діаграмі стану і включає:

- `Idle`: очікує обробки замовлення, шукаючи його за допомогою внутрішнього переходу;
- (тільки для вхідних) `unload`: вивантажує замовлення з транспорту, надіславши повідомлення йому, таким чином, ініціюючи перехід повідомлення;
- (тільки для вхідних) `to_wh`: шукає перший доступний склад для розміщення обробленого замовлення та надсилає його туди;
- (тільки для вихідних) `from_wh`: переміщує замовлення зі складу;
- (тільки для вихідних) `load`: завантаження замовлення на транспорт такими ж операціями, як і при розвантаженні;
- `busy`: після обробки одного замовлення лічильник `actual_capacity` збільшується, і процес перезапущений. Якщо `actual_capacity` досягла `max_capacity`, ініціюється умовний перехід до наступного стану;
- `wait`: чекає на скидання потужності та початок нового циклу. Таймер подій запускається, як тільки обробляється перше замовлення, слідуючи тієї самої логіки, яка була прийнята раніше для транспорту.

Перебуваючи в стані `idle`, агент обробки періодично запускає внутрішній перехід для обробки замовлень, що показано в наступних рядках коду (приклад 3.3).

```

    if (! Is_busy) {
    if (
    isIn () && isRail ()
? ! main.collection_teu_in_wh.isEmpty ()
:! Main.collection_ton_in_wh.isEmpty ()
)
    selected_order = isRail ()
? main.collection_teu_in_wh.pop ()
: main.collection_ton_in_wh.pop ();

    if (
! inIn () && isRail ()
? ! main.collection_teu_out_wh.isEmpty ()
:! Main.collection_ton_out_wh.isEmpty ()
)
    selected_order = isRail ()
? main.collection_teu_out_wh.pop ()
: main.collection_ton_out_wh.pop ();

    is_busy = true ;
    resetconditionevent.restart ();

```

Приклад 3.3 - Обробка, стан idle, внутрішній перехід

Процес схожий на той, що вже задокументований для транспортного стану `idle`, як, наприклад, попередня перевірка збору, щоб уникнути викидання винятку. Тут, зокрема, якщо оператори використовуються разом із тернарним оператором, щоб вибрати правильні колекції замовлень. Залежно від параметрів агента обробки, агент вибирає одну з чотирьох альтернатив: вхідну `teu` (рядок 8), вхідну тонну (рядок 9), вихідний `teu` (рядок 17), вихідний тон (рядок 18). Після цього він змінює змінну на `true`, щоб викликати умовний перехід до наступного стану. Нарешті, таймер події для ініціювання переходу до скидання починається, що дозволяє точно відповідати ліміту, заданому `max_capacity` у зазначений часовий проміжок.

У стані `unload` (або `load`) після відправлення повідомлення про переміщення замовлення та лічильник транспортного навантаження зменшено, інша дія повинна бути виконана додатково (приклад 3.4).

```

if (isSea ()) {
send ( "unload" , selected_order);
selected_transport.actual_capacity--;
if (selected_transport.actual_capacity == 0)
deliver ( "isempty" , selected_transport);
// selected_transport.onChange ()
}

```

Приклад 3.4 – Обробка, стан unload, вхідна дія

Транспортний агент перевіряє, чи транспорт повністю вивантажений (чи завантажений) і, якщо ця умова виконана, надсилає йому повідомлення, щоб викликати його перехід повідомлення. Цей останній уривок є обов'язковим, оскільки в AnyLogic умовні переходи агента контролюються лише в тому випадку, якщо зміна змінної виконується всередині того ж самого агента. Якщо зміна проводиться зовні, стан не контролюється, тобто handling агент вносить зміни до actual_capacity транспортного агента. Виникають два рішення: або надсилання повідомлення на транспорт (рядок 5), або виклик метод selected_transport.onChange(), щоб повідомити транспортного агента про зміни (рядок 6).

У стані to_wh виконується такий внутрішній перехід (приклад 3.5):

```
for (Warehouse w: isRail ()
? main.warehouses_rail
: main.warehouses_motor
) {
if (w.actual_capacity < w.max_capacity &&! Is_found) {
send ( "movetowh" , selected_order);
selected_order.assigned wh = w;
w.actual_capacity ++;

is_found = true ;
}
}
```

Приклад 3.5 – Обробка, to_wh стан, внутрішній перехід

Транспортні агенти шукають наявний склад, використовуючи цикл for; зокрема, список, через який він повторюється, подається з використанням - як і в попередніх випадках - isRail та тернарного оператора, який повертає рядок 2, якщо істина, рядок 3 – в іншому випадку. Коли знайдено перше позитивне входження, до замовлення надсилається повідомлення про активацію переходу до стану warehouse (рядок 6), склад зберігається в історії замовлення (рядок 7) і збільшується місткість складу (рядок 8). Нарешті, логічна умова змінюється на true, щоб зупинити ітеративний пошук і викликати умовний перехід агента обробки в наступний стан. В стані load виникає подібний процес, знову викликаючи внутрішній

перехід (приклад 3.6).

```

for (Transport t: isRail ()
? filter (main.trains, t -> t.inState (t.loading_out))
: filter (main.motors, t -> t.inState (t.loading_out))
) {
if (t.actual_capacity < t.max_capacity &&! Is_found) {
send ( "load" , selected_order);
t.actual_capacity ++;
is_found = true ;}
if (t.actual_capacity > = t.max_capacity) send ( "go" , t);
}

```

Приклад 3.6 – Обробка, стан завантаження, внутрішній перехід

Агенти обробки шукають доступний транспорт за допомогою циклу `for`; у цьому випадку особливість полягає в тому, що список, через який він повторюється, відфільтрований по порядку щоб вибрати лише транспортні агенти, які перебувають у стані `loading_out`. Залежно від `isRail` результату, вибраний список знаходиться в рядку 2 або рядку 3. Коли перший доступний транспорт знайдено, на замовлення надсилається повідомлення про ініціювання переходу у стан `out_transport` (рядок 6), фактична пропускна спроможність транспорту збільшується (рядок 7) і логічна умова змінюється на `true`, щоб зупинити ітеративний пошук і викликати умовний перехід агента обробки в наступний стан (рядок 8). Нарешті, з тієї самої причини, що стосується навантаження, транспортне повідомлення має надіслати повідомлення на випадок повного завантаження для запуску його переходу; в якості альтернативи, умовний перехід міг би використовуватися для транспортного агента в поєднанні з `t.onChange()`.

3.3.5 Модель Warehouse

Тип складу дуже простий (рисунок 3.7), і його не потрібно моделювати як агента, але до нього ставляться так само для зручності. Незважаючи на те, що `AnyLogic` дозволяє створювати власні об'єкти `Java`, вони не мають графічного інтерфейсу, і тому програмне забезпечення не робить різниці між

об'єктом та агентами на практиці, останні є кращими.



Рисунок 3.7 – Модель: Warehouse

Не потрібно використовувати жодну діаграму стану, оскільки єдина сфера застосування складу - це пасивне розміщення замовлень, річ, яка можлива шляхом перевірки чи `actual_capacity` нижче за `max_capacity`. Як завжди `type` використовується для розрізнення залізниці та авто, тоді як функції використовуються для збору індивідуальної статистики. Подальше розвинення типу Warehouse, наприклад із внутрішньою обробкою, наявністю ресурсів та розподілом фізичного простору залежно від характеристик замовлення, залишається відкритою точкою для подальшого розвитку.

3.3.6 Модель Link

Враховуючи, що залізниця та автотранспорт взаємопов'язані, `link` служить точкою зв'язку між двома екосистемами, піклуючись про замовлення (рисунок 3.8). Це отримується у поєднанні з логічними параметрами `cross` та `ground_link`, які вставляються в кожному порядку.

Перший - якщо `true` - робить можливим, що як тільки замовлення досягне складу, він чекає, поки повідомлення з ресурсу посилення буде переміщено до `link` стану. Перехід часу очікування, який перемістив би його до вихідного сектора, ніколи не виконується для перехресних замовлень завдяки вставленому в перехід логічному виразу "охорона після спрацьовування".

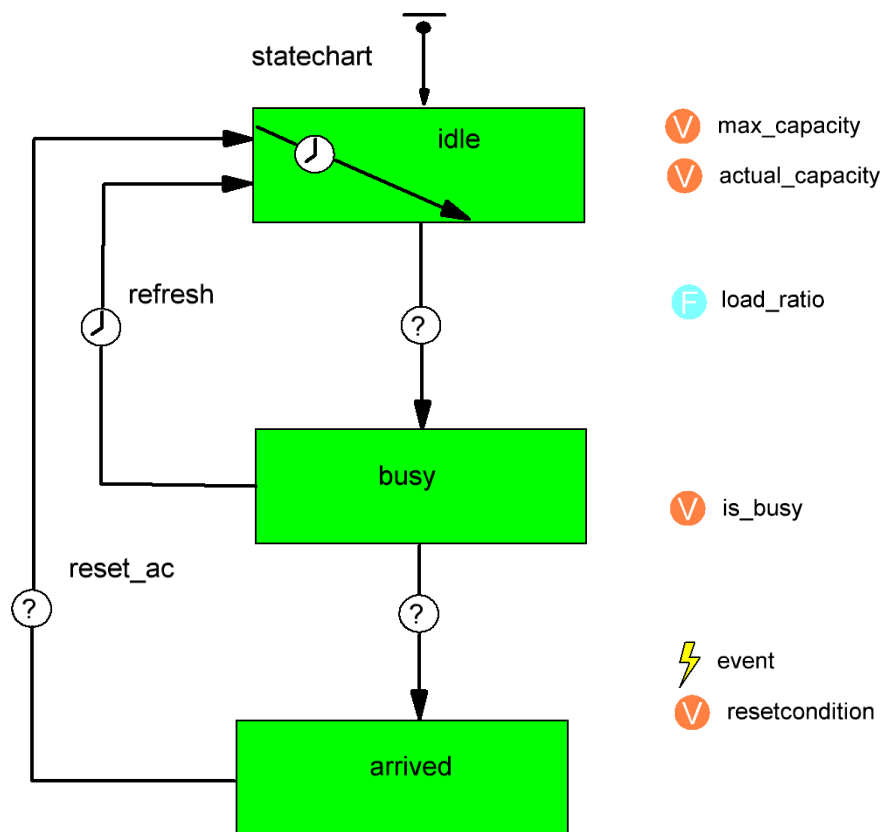


Рисунок 3.8 – Модель: Link

З тією ж логікою, що діє для видів транспортування та обробки, внутрішній перехід в режимі `idle` використовується для вилову замовлень, що очікують на конвертацію (приклад 3.7).

```

if (actual_capacity < max_capacity &&
! main.collection_link.isEmpty ()) {
t = main.collection_link.pop ();
send ( "link" , t);
main.convert (t);

actual_capacity += t.isRail ()? 1: 1,0 /
main.conversion_ton_teu;
is_busy = true ;
event.restart ();
}
  
```

Приклад 3.7 – Зв'язок, стан очікування, внутрішній перехід

Це отримується, як зазвичай, надсилаючи повідомлення на замовлення (рядок 4), таким чином запускаючи його пов'язаний перехід повідомлення в стан `link`. Тоді фактична пропускна здатність каналу збільшується на

одиницю teu або її еквівалент в тоннах, перетворених у teus, з урахуванням визначеного користувачем коефіцієнтом перетворення (рядок 7).

Реальне перетворення (рисунок 3.9) відбувається в рядку 5, викликаючи функцію головного рівня `main.convert(t)`.

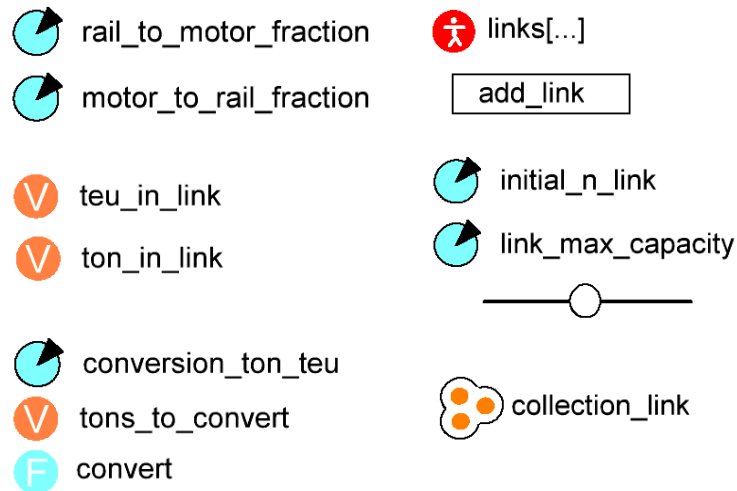


Рисунок 3.9 – Модель: Main(2)

Функція `main.convert(t)` отримує замовлення на перетворення як аргумент і коротко пояснюється далі (приклад 3.8).

```

if (t.isRail ())
for (i = 0; i <conversion_ton_teu; i++) {
new_order = add_tons ( "motor" , true , true );
}
else {
tons_to_convert++;
if (tons_to_convert >= conversion_ton_teu) {
New_order = add_teus ( "rail" , true , true );
Tons_to_convert -= conversion_ton_teu;
}
}
}

```

Приклад 3.8 – Main, функція перетворення

У випадку, якщо наказ про перехід йде з залізниці, teu розбивається і створюється еквівалентна кількість тонн; у рядку 3 посилаються аргументи в дужках до параметрів, переданих при створенні нового замовлення, відповідно `type`, `cross`, `from_link`. Якщо замовлення надходить з автотранспорту, багато тонн змішуються в одній teu, що створюється при досягненні коефіцієнта перерахунку.

4 ЕКСПЕРЕМЕНТАЛЬНЕ МОДЕЛЮВАННЯ

4.1 Моделювання агентної моделі логістичних послуг

З практичної точки зору стає важко зрозуміти поведінку системи на сукупному (агрегатному) рівні. Агентне моделювання значною мірою покладається на властивість спливання, тому опису окремих його компонентів іноді не достатньо для прогнозування результатів у цілій системі. З цієї причини для полегшення цього завдання пропонується практичний інструмент - інформаційна панель, яка представлена далі (рисунок 4.1). На одному екрані можна отримати повний доступ до усіх відповідних КРІ, які використовуються для моніторингу системи. Він має кілька графіків для того, щоб відстежувати ситуацію на кожному часовому кроці та надає реальну інформацію. Він поступово розвивається, слідуючи часу моделювання, пропонуючи можливість внесення змін до параметрів системи за допомогою повзунків та кнопок і можливість відразу бачити наслідки в режимі реального часу.

Макет організований у вигляді сітки: верхня частина стосується ресурсів залізниці, тоді як нижня частина стосується ресурсів автотранспортної системи. Стовпці використовуються відповідно для ідентифікації сукупних замовлень, транспорту, обробки та складів.

Графіки можна згрупувати за трьома категоріями:

- WIP: графік, що описує замовлення, стосується фактичної кількості в обігу в системі на кожному кроці часу, розрізняючи замовлення в режимі idle (або відставання), завантажене у транспорті або кероване агентами обробки. Коли замовлення успішно надійшло на склад, воно більше не враховується. Таким чином, представляючи поточні незавершені замовлення (WIP), графік слідує логіці "нижче - краще".

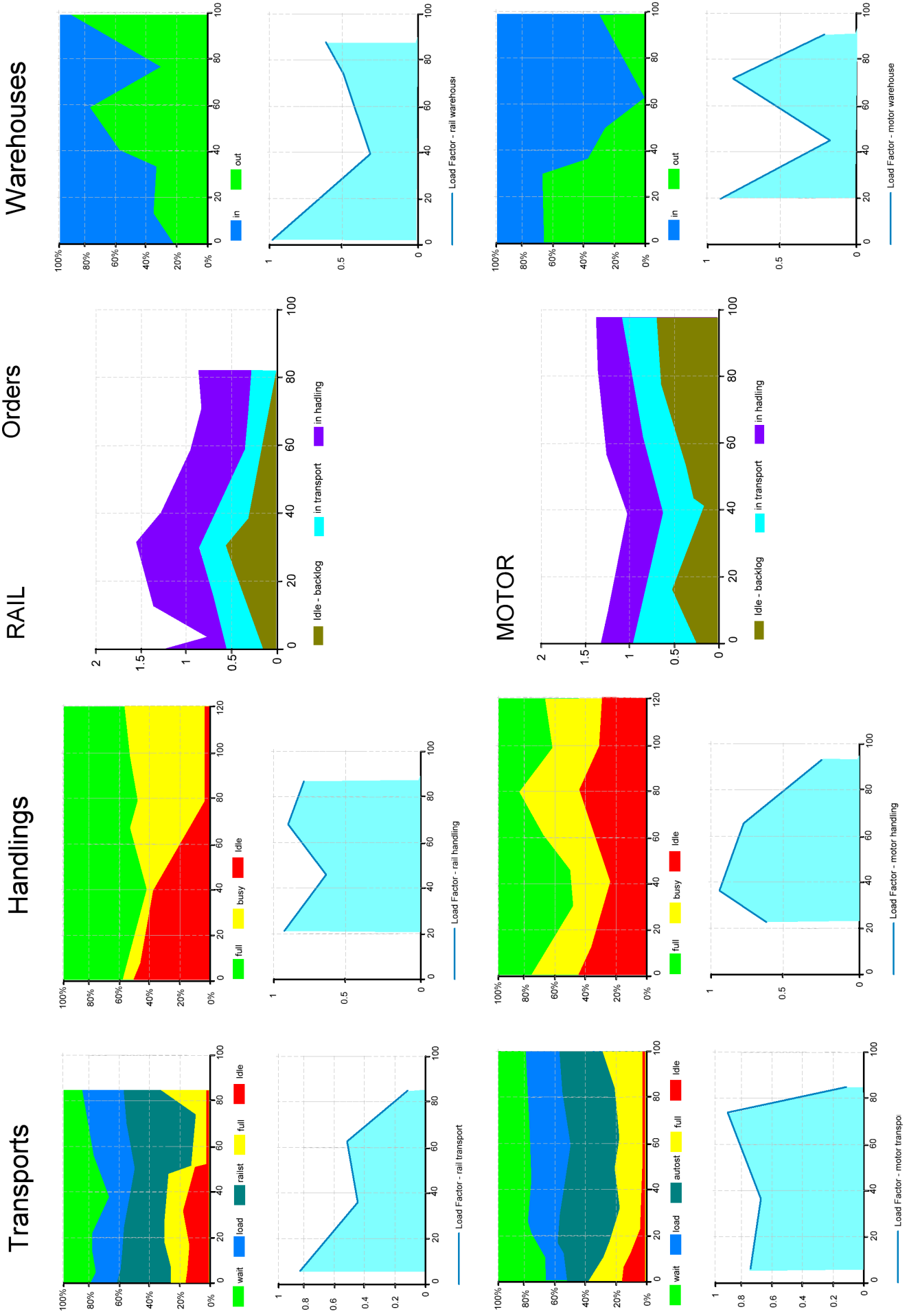


Рисунок 4.1 – Динамічна інформаційна панель

- State: кольорові графіки в першому та третьому рядках використовуються для швидкого подання транспорту та обробляє фактичний стан у діаграмі стану. Зокрема червоний використовується для виділення ресурсів у режимі idle, тобто не працює при надлишковій ємності. Логіка, якої дотримуються, полягає в тому, що "зелений краще", тому що таким чином він означає, що ресурси швидко завершили призначені операції та просто чекають скидання, щоб почати знову.

Навпаки, оскільки склад не має станів, графік представляє набір замовлень у ньому, розрізняючи вхідні та вихідні замовлення. В основному він повинен відображатись зеленим, але можлива проблема може виникнути, якщо синя частка значно збільшується, це означає, що агенти `handling_out` намагаються обробити вихідні замовлення, що ускладнює звільнення місця на складі.

- Load Factor: Світло-сині графіки у другому та четвертому рядках представляють коефіцієнт завантаження кожного ресурсу, тобто відношення фактичної суми попиту замовлення і максимальної потужності. Як наслідок, значення, близьке до 100% слід підтримувати. Якщо коефіцієнт набагато нижчий, ресурс в даний час відчуває надлишкову потужність, тоді як, з іншого боку, якщо коефіцієнт більше ресурс потребує оновлення, інакше замовлення почнуться накопичуватись у відставанні.

Найкращий спосіб зрозуміти як все функціонує – показати, поведінку на прикладі реального випадку та опису поряд із кожною з семи фаз.

Приклад показує (рисунок 4.2, 4.3), як збільшення попиту впливає на всю систему, шляхом погляду на ресурси, задіяних в процесі, з особливим акцентом на коефіцієнт використання. Поки попит зростає, потрібно додати додаткових агентів до системи для компенсації втрачених замовлень через запас. Таким чином, це впливає на пропускну здатність системи, і в свою чергу змінюється коефіцієнт навантаження інших популяцій.

Для простоти враховуються лише замовлення `teu`, але наступний опис можна використовувати також у випадку замовлення тонн, оскільки дві

екосистеми – залізниці та автотранспорт – змодельовано дзеркально і дотримуються тієї ж логіки процесу.

При фазі 1 попит нижчий за фактичну потужність системи, задану мінімумом між місткістю транспорту, навантажень та складів. В цьому випадку вузьке місце визначається транспортною спроможністю, яка має найбільший коефіцієнт навантаження. На даний момент його значення все ще нижче 100%, так що попит може бути повністю задоволений.

Значення графіку WIP дуже близьке до нуля, що вказує на відсутність відставання, і замовлення швидко доставляються до місця призначення.

Що стосується обробки, графік коефіцієнта навантаження показує досить низьке значення, що припускає, що ресурс недостатньо використаний щодо його потужності. Відповідний графік стану вище показує, що в середньому деякі ресурси не працюють (в режимі очікування, червоним).

Склад не демонструє жодної цікавої поведінки, крім нормального стану графічних коливань, заданих дискретною природою використовуваної парадигми.

Коли настає фаза 2, збільшується попит, а графік коефіцієнта транспортного навантаження показує значення близько 100%, а це означає, що попит зараз дорівнює потужності цілої системи. Графік WIP набуває форми прямокутника, що означає, що середня кількість замовлень в режимі очікування залишається незмінною.

Відтоді нічого важливого для ресурсів обробки та складських ресурсів не відбувається, тому обмеження пропускної здатності спричинене транспортом.

На третій фазі попит більший, ніж пропускна здатність, а коефіцієнт транспортного навантаження показує величину більше, ніж одиниця, що означає, що потрібно уникнути оновлення запасів. З цієї причини замовлення починають накопичуватися в режимі очікування, а якщо коригувальних заходів не проводиться, незабаром їх збираються ліквідувати і зараховуються як запас.

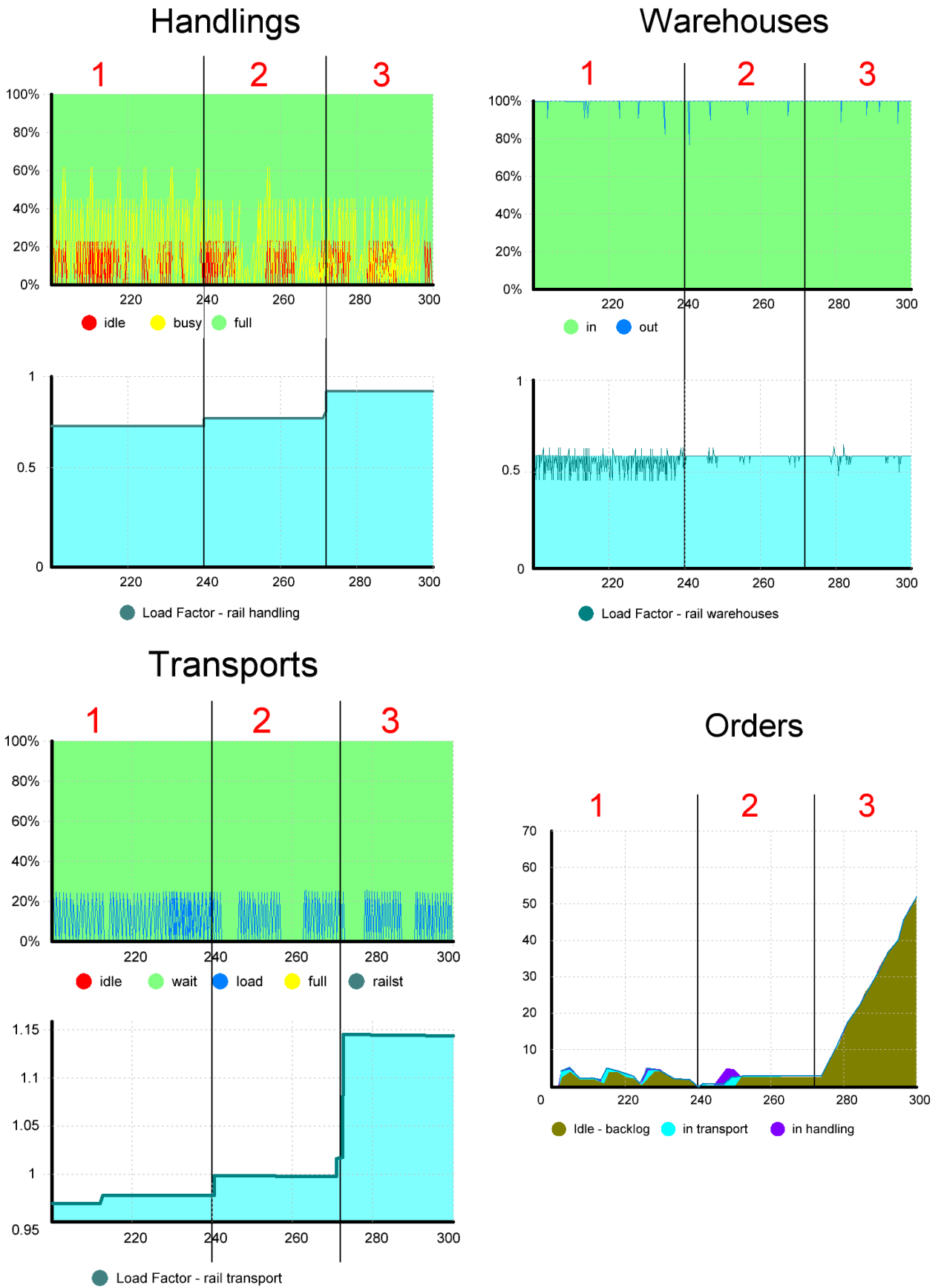


Рисунок 4.2 – Приклад приборної панелі (1)

Наступає фаза 4. Зберігаючи той самий попит, що і на фазі 3, жодна інвестиція не проводилась, щоб збільшити потужність транспортних ресурсів. Резерв замовлень накопичувався протягом тривалого часу у прострочці аж до того, що спрацьовував їхній індивідуальний таймер. WIP більше не росте і тримається на стабільному рівні. Неможливо обробити прострочене замовлення протягом тривалого часу, тому результати замовлень втрачаються і враховуються як резерв після спрацьовування індивідуального таймеру. Можливість підвищення потужності необхідна і не може бути відкладена.

Коли настає фаза 5, відбувається наступний сценарій: після додавання до системи нових транспортних агентів коефіцієнт транспортного навантаження зменшується нижче 100%. Отже, прострочка замовлення починає зменшуватися, і як наслідок резервування зараз припиняється. Зокрема, незначна частина WIP зараз складається із замовлень, які знаходяться у стані "в транспорті", а це означає, що вони чекають на обробку агентами handling. Коефіцієнт навантаження на обробку є близьким до 100%, цей ресурс генерує вузьке місце для всієї системи. Відповідний графік стану показує, що ресурси перейшли з простою в режим роботи. Це пов'язано з графіком стану транспорту, який показує, що значна кількість агентів чекає на вивантаження (темно-зелений) або завантаження (синій) агентами обробки. Склад стикається зі збільшенням середньої кількості замовлень на розміщення як наслідок швидкого відновлення прострочених замовлень, оскільки зараз в системі більше замовлень.

На фазі 6 до системи додаються нові транспортні агенти, щоб поглинути відставання замовлень швидше. Нахил графіка WIP обертається всередину, це означає, що попит на відставання тепер потребує менше часу для обробки, таким чином відновлюється нормальний операційний рівень замовлень. Це може здатися розумним маневром, але на практиці це вимагає інвестицій, які зараз не є важливими для реального моменту і буде означати негативні наслідки.

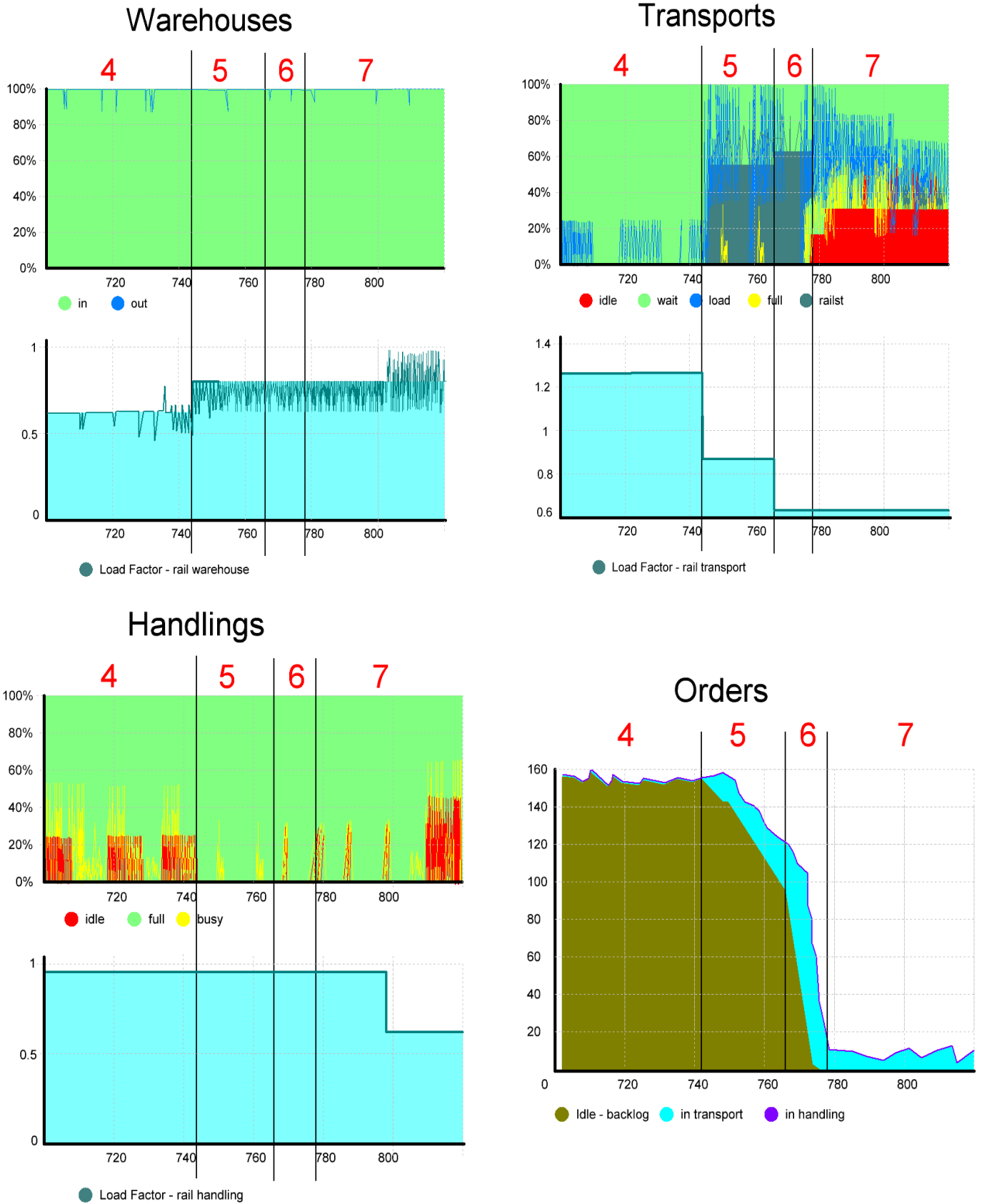


Рисунок 4.3 – Приклад приборної панелі (2)

Як природний наслідок, за фази 7 графік WIP знову знаходиться під контролем, але надмірне інвестування попереднього етапу зараз призводить до надлишкової ємності транспортних агентів. Коефіцієнт навантаження залишається таким же, як і на попередній фазі, оскільки попит однаковий, але графік стану показує, що частка ресурсів у простої, а це означає недостатнє використання. В кінці фази 7 був доданий ще один обробний агент, просто щоб довести, що ця інвестиція у ресурси не вузького місця призводить лише до недостатнього використання, як показано в графіках стану та коефіцієнту навантаження.

4.2 Тестування агентної моделі логістичних послуг

При моделюванні першою метою розробника є побудова системи, яка може бути представником реального модельованого світу. Отже, фундаментальний крок процесу моделювання включає тестування отриманих результатів з даними попередніми або подібні доступні моделі. Сфера оцінки полягає в тому, що модель така якомога ближче до реальності, щоб її можна було безпечно використовувати в якості практичного інструменту для підтримки прийняття рішень.

З метою перевірки моделі, заснованої на агентах, виконується порівняння з оригіналом моделі системної динаміки. Зокрема, для того, щоб мати порівняні результати, ті самі вихідні дані були подані в систему. Це стосується не тільки початкових змінних та таблиці пошуку, але також до структури математичних операцій, що виконуються в оригінальній моделі СД. Якщо бути точним, правильніше стверджувати, що системна динаміка та модель, заснована на агентах, були об'єднані в гібридну модель.

Як спочатку було введено в процедуру реконцептуалізації, місце займають агенти запасів, що описують ресурси, які створюють вузьке місце, тобто місткість потягів, місткість обробки, місткість складів, місткість вантажівок, місткість обробки при автотранспортуванні, складська

потужність при автотранспортуванні. Отже, кожен із них зараз виражається як сума ємності кожного окремого агента, яка задана наступною функцією: $\text{sum}(\text{population}, p \rightarrow p.\text{max_capacity})$. Можна помітити, це лише приклад того, що спеціально зроблено для збереження індивідуальності агента, що дозволяє диференціювати характеристики в них.

Наступні результати (рисунки 4.4-4.9) показують повну відповідність системно-динамічній моделі, таким чином перевіряючи модель об'єкта дослідження.

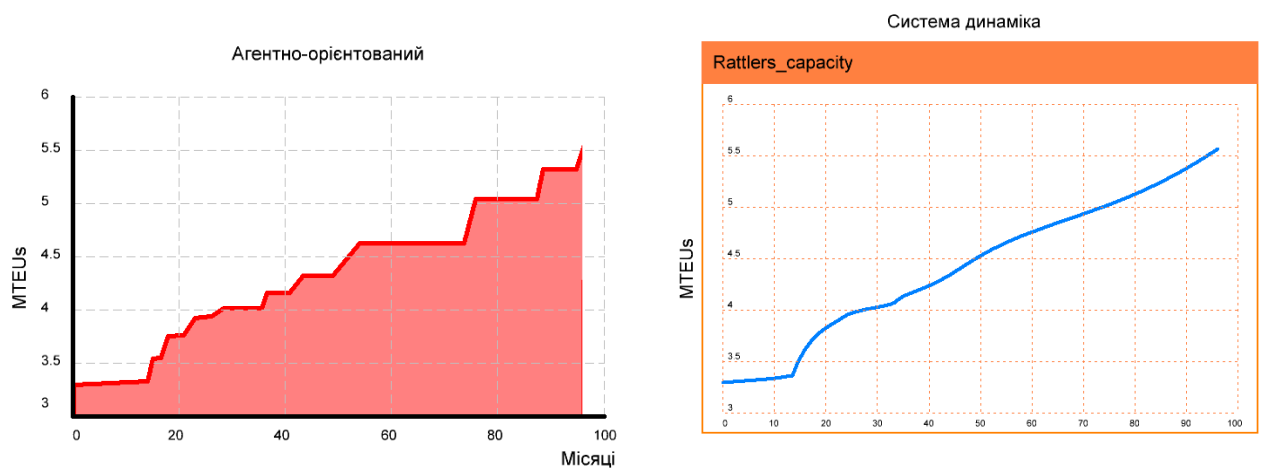


Рисунок 4.4 – Перевірка: залізничний транспорт

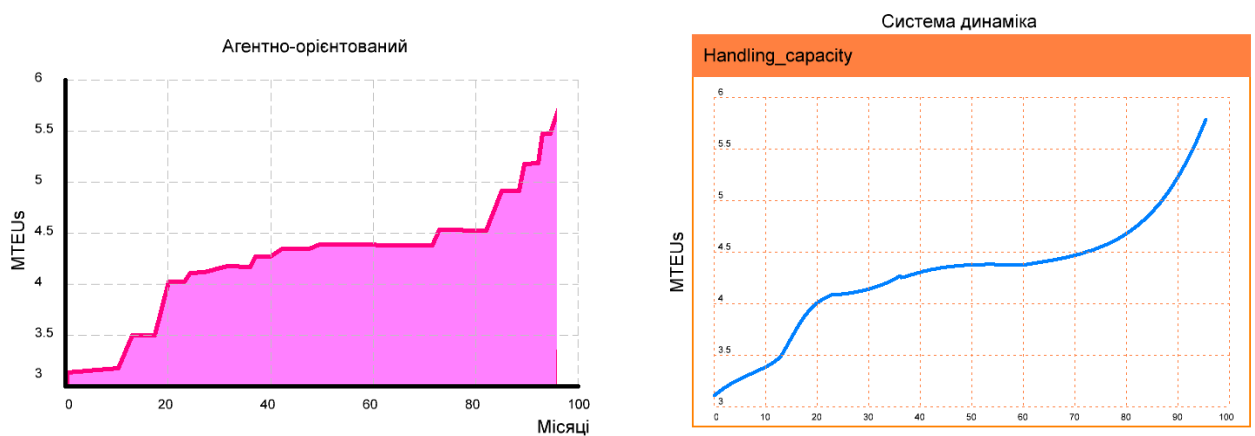


Рисунок 4.5 – Перевірка: залізнична обробка

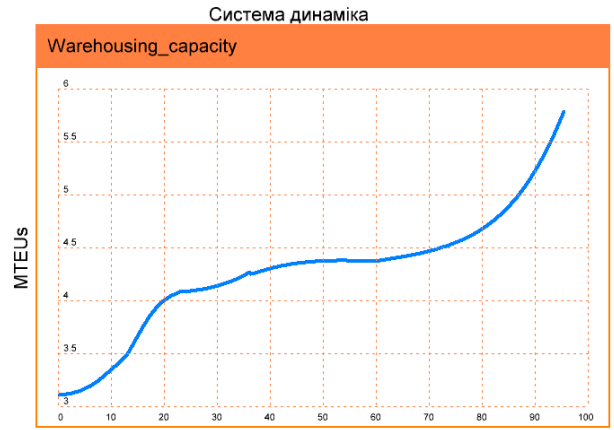


Рисунок 4.6 - Перевірка: залізничні склади

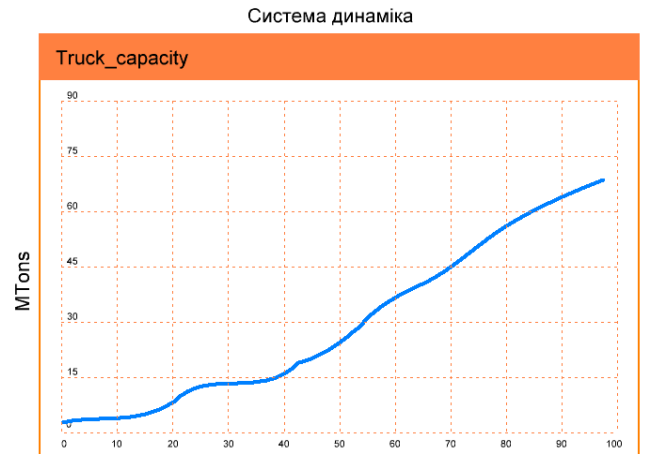
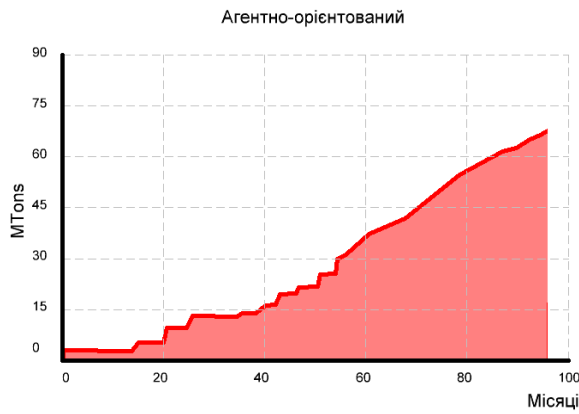


Рисунок 4.7 – Перевірка: автотранспорт

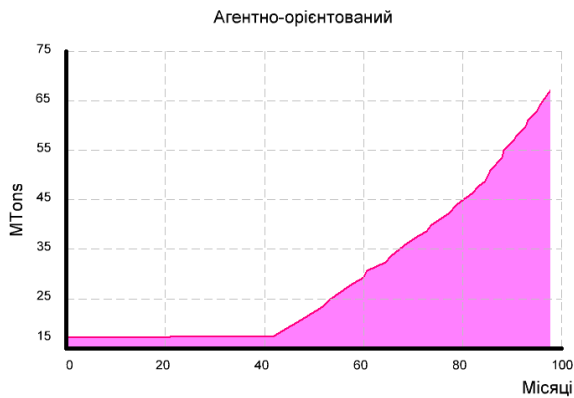


Рисунок 4.8 – Перевірка: автотранспортна обробка

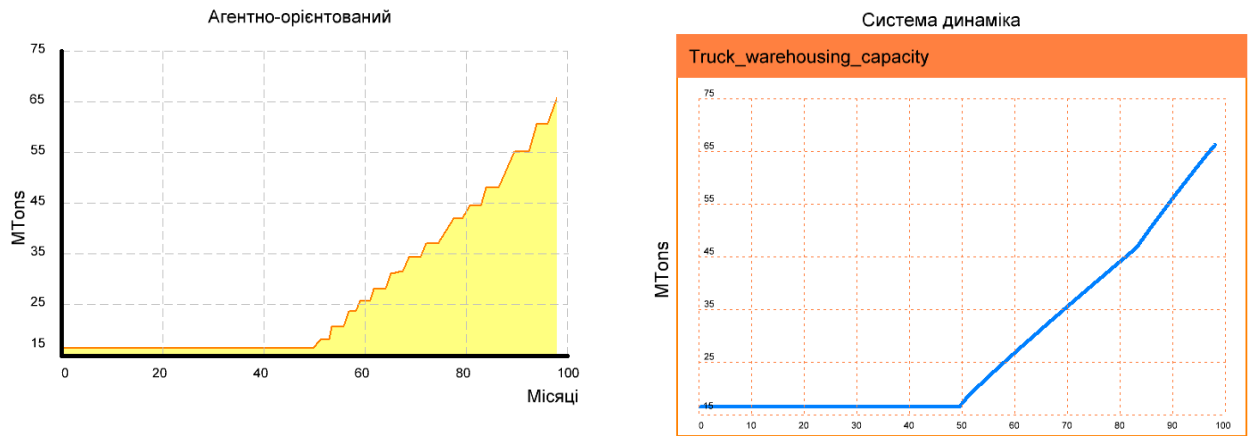


Рисунок 4.9 - Перевірка: автотранспортні склади

Для стислості вище були представлені графіки лише одного сценарію цілої перевірки. Вони посилаються на "найкращий сценарій", який містить наступні характеристики:

- перехресний попит є активним;
- перетворення залізниця-автотранспорт складає 15%;
- перетворення автотранспорт-залізниця складає 15%;
- зацікавлені сторони демонструють поведінку несприятливого ризику.

Тести проводились у кожному сценарії, і дійсність моделі завжди перевірялась, за винятком невеликої різниці, яка відбувається з оглядом на дискретизацію. Насправді, уточнення необхідне: поетапну поведінку можна вважати характеристикою агента моделювання, задану його дискретним характером; навпаки, СД дає безперервні рішення. З цієї причини на графіках системної динаміки, представлених вище, демонструється ідеальна безперервна поведінка, як це природно очікується від моделі системної динаміки.

ВИСНОВКИ

У галузі вивчення складних систем агентне моделювання являє собою значний інноваційний крок, враховуючи його здатність обробляти більше даних та виділяти індивідуальну поведінку та взаємодії. В останні роки АОМ використовується для зростаючої кількості додатків у різних галузях та дисциплінах - як у логістичному секторі - для вирішення проблем, що вимагають глибокого аналізу даних та прийняття складних рішень.

Однак слід також пояснити, що агентне моделювання не завжди надає цінну інформацію для деяких додатків, будучи менш ефективним за інші, менш потужні парадигми. Це пов'язано з тим, що для належної роботи необхідно надавати достатню кількість детальних даних як вхідних даних до моделі, щоб отримати вигоду від її повного потенціалу. Насправді, агентне моделювання призначене для тих, хто не боїться забруднити руки та вийти за межі традиційних підходів; це особливо актуально якщо система, що моделюється, містить активні об'єкти (люди, бізнес-одиниці, тварини, транспортні засоби, проекти, запаси, продукція тощо). Для такого типу систем AnyLogic дозволяє ефективно розробляти агенти з мінімальним кодуванням, а також може підтримувати міграцію з інших парадигм та інтеграцію в гібридну модель.

Отже, в ході виконання магістерської роботи було:

- досліджено методи та моделі побудови агентно-орієнтованих систем;
- досліджено логістичну ситуацію вантажних перевезень в Харкові;
- розроблено агентно-орієнтовану модель для імітації вантажних перевезень;
- проаналізовано отримані результати, та надані деякі поради щодо покращення сучасної ситуації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. McFarlane D. Intelligent logistics: Involving the customer / D. McFarlane, V. Giannikas, W. Lu // *Computers in Industry*. — 2016. — Vol. 81. — Pp. 105–115. DOI: 10.1016/j.compind.2015.10.002.
2. Wooldridge M. An Introduction to Multi-Agent Systems / M. Wooldridge. — John Wiley & Sons, 2009. — 368 p.
3. Leitão P. Recent developments and future trends of industrial agents / P. Leitão, P. Vrba // *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*. — Springer, Berlin, Heidelberg, 2011. — Pp. 15–28. DOI: 10.1007/978-3-642-23181-0_2.
4. Fragapane G. I. et al. An agent-based simulation approach to model hospital logistics // *Int J Simul Model*. — 2019. — T. 18. — №. 4. — С. 654-665.
5. Sakai T. et al. SimMobility Freight: An agent-based urban freight simulator for evaluating logistics solutions // *Transportation Research Part E: Logistics and Transportation Review*. — 2020. — T. 141. — С. 102017.
6. Pal K. Agent-Based Simulation for Supply Chain Transport Corridors // *International Journal of Computer and Information Engineering*. — 2015. — T. 9. — №. 7. — С. 1677-1681.
7. Winikoff M. et al. Agent-based container terminal optimisation // *German Conference on Multiagent System Technologies*. — Springer, Berlin, Heidelberg, 2011. — С. 137-148.
8. Démare T. et al. Modeling logistic systems with an agent-based model and dynamic graphs // *Journal of Transport Geography*. — 2017. — T. 62. — С. 51-65.
9. Vidal J. M., Huynh N. Building agent-based models of seaport container terminals // *International Conference on Autonomous Agents and Multiagent Systems*. — 2010. — С. 49-58.
10. Peng Y. et al. Evaluation of airport capacity through agent based simulation // *International Journal of Grid and Distributed Computing*. — 2014. — T.

7. – №. 6. – C. 165-174.

11. Ribino P. et al. Agent-based simulation study for improving logistic warehouse performance //Journal of Simulation. – 2018. – T. 12. – №. 1. – C. 23-41.

12. Henesey L., Davidsson P., Persson J. Simulation of operational policies for transshipment in a container terminal //Proceedings of the 10th WSEAS international conference on Computers. – World Scientific and Engineering Academy and Society (WSEAS), 2006. – C. 1342-1347.

13. Anand N. et al. Validation of an agent based model using a participatory simulation gaming approach: the case of city logistics //Transportation Research Part C: Emerging Technologies. – 2016. – T. 71. – C. 489-499.

14. Barenji A. V. et al. Intelligent E-commerce logistics platform using hybrid agent based approach //Transportation Research Part E: Logistics and Transportation Review. – 2019. – T. 126. – C. 15-31.

15. Van Heeswijk W. J. A. et al. Evaluating urban logistics schemes using agent-based simulation //Transportation science. – 2020. – T. 54. – №. 3. – C. 651-675.

16. Jabeur N. et al. Toward leveraging smart logistics collaboration with a multi-agent system based solution //Procedia Computer Science. – 2017. – T. 109. – C. 672-679.

17. Serrano-Hernandez A. et al. Agent-based simulation for horizontal cooperation in logistics and transportation: From the individual to the grand coalition //Simulation Modelling Practice and Theory. – 2018. – T. 85. – C. 47-59.

18. Wang Z., Zhang J. Agent-based evaluation of humanitarian relief goods supply capability //International Journal of Disaster Risk Reduction. – 2019. – T. 36. – C. 101105.

19. Gusah L., Cameron-Rogers R., Thompson R. G. A systems analysis of empty container logistics—a case study of Melbourne, Australia //Transportation Research Procedia. – 2019. – T. 39. – C. 92-103.

20. Irannezhad E., Hickman M., Prato C. G. Modeling the efficiency of a port community system as an agent-based process //Procedia Computer Science. – 2017. – T. 109. – C. 917-922.

21. Cagliano A. C. et al. Assessing city logistics projects: A business-oriented approach //Proceedings of the Summer School Francesco Turco. – 2016. – C. 13-15.
22. Jennings N. R., Sycara K., Wooldridge M. A roadmap of agent research and development //Autonomous agents and multi-agent systems. – 1998. – T. 1. – №. 1. – C. 7-38.
23. Franklin S., Graesser A. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents //International Workshop on Agent Theories, Architectures, and Languages. – Springer, Berlin, Heidelberg, 1996. – C. 21-35.
24. Wooldridge M. J., Jennings N. R. Intelligent agents: Theory and practice //The knowledge engineering review. – 1995. – T. 10. – №. 2. – C. 115-152.
25. Bandini S., Manzoni S., Vizzari G. Agent based modeling and simulation: an informatics perspective //Journal of Artificial Societies and Social Simulation. – 2009. – T. 12. – №. 4. – C. 4.
26. Chan W. K. V., Son Y. J., Macal C. M. Agent-based simulation tutorial-simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation //Proceedings of the 2010 winter simulation conference. – IEEE, 2010. – C. 135-150.
27. Taniguchi E., Tamagawa D. Evaluating city logistics measures considering the behavior of several stakeholders //Journal of the Eastern Asia Society for Transportation Studies. – 2005. – T. 6. – C. 3062-3076.
28. Teo J. S. E., Taniguchi E., Qureshi A. G. Evaluation of load factor control and urban freight road pricing joint schemes with multi-agent systems learning models //Procedia-Social and Behavioral Sciences. – 2014. – T. 125. – C. 62-74.
29. Tamagawa D., Taniguchi E., Yamada T. Evaluating city logistics measures using a multi-agent model //Procedia-Social and Behavioral Sciences. – 2010. – T. 2. – №. 3. – C. 6002-6012.
30. Roorda M. J. et al. A conceptual framework for agent-based modelling of logistics services //Transportation Research Part E: Logistics and Transportation

Review. – 2010. – Т. 46. – №. 1. – С. 18-31.

31. Vidal J. M., Huynh N. Building agent-based models of seaport container terminals //International Conference on Autonomous Agents and Multiagent Systems. – 2010. – С. 49-58.

32. Borshchev A., Filippov A. From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools //Proceedings of the 22nd international conference of the system dynamics society. – 2004. – Т. 22.

33. Borshchev A. The big book of simulation modeling: multimethod modeling with AnyLogic 6. – AnyLogic North America, 2013. – С. 115-160.

34. Sterman J. D. System dynamics modeling: tools for learning in a complex world //California management review. – 2001. – Т. 43. – №. 4. – С. 8-25.

35. Borshchev A. et al. Multi-method modelling: AnyLogic //Discrete-event simulation and system dynamics for management decision making. – 2014. – С. 248-279.

36. Дольнев Р. Огляд моделей агентів на штучному фінансовому ринку. *Методи та засоби обчислювального інтелекту: матеріали XXIV міжнар. молод. форуму, м. Харків, 7-9 квіт. 2020 р. Харків, 2020. С. 163-164.*

37. Дольнев Р. Проблема оцифрування паперових документів. *Інформаційні технології та системи управління в економіці, техніці та екології: тези доп. міжнар. наук.-практ. конф., м. Харків, 20-21 лют. 2020 р. Харків, 2020. С. 31.*