

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи розподіленого імітаційного моделювання в
програмних засобах дослідження операцій та управління

(тема)

Виконав:

студент II курсу, групи СПЗм-21-1
Коваленко А.В.
(прізвище, ініціали)

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

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: проф. Волк М.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Коваленку Анатолію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи розподіленого імітаційного моделювання в програмних засобах дослідження операцій та управління

затверджена наказом по університету від “ 24 ” березня 2023 р. № 60 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 17 травня 2023 р.

3. Вхідні дані до роботи _____

Моделі та методи розподіленого імітаційного моделювання

Архітектура високого рівня HLA

Методологічні підходи до вибору фрейворку розподіленого моделювання

4. Перелік питань, що потрібно опрацювати у роботі _____

Аналіз предметної області

Розробка фрейворку розподіленого моделювання

Розробка методу методу розподіленого імітаційного моделювання

в програмних засобах дослідження операцій та управління

Проведення експериментальних досліджень

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	27.03.23 – 06.04.23	
2	Розробка методів	06.04.23 – 12.04.23	
3	Реалізація алгоритмів	13.04.23 – 20.04.23	
4	Розробка структури програмних засобів	21.04.23 – 25.04.23	
5	Розробка програмних модулів	26.04.23 – 30.04.23	
6	Оформлення матеріалів атестаційної роботи	01.05.23 – 10.05.23	
7	Подання атестаційної роботи керівникові та її попередній захист	11.05.23 – 14.05.23	
8	Подання атестаційної роботи на рецензування	14.05.23 – 17.05.23	

Дата видачі завдання 27 березня 2023 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Волк М.О.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 69 с., 8 рис., 2 табл., 1 дод., 52 джерела.

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ, АРХІТЕКТУРА ВИСОКОГО РІВНЯ, РОЗПОДІЛЕНЕ ПРОГРАМУВАННЯ

Метою кваліфікаційної роботи є дослідження та розробка методу розподіленого імітаційного моделювання в програмних засобах дослідження операцій та управління.

Розподілене моделювання дозволяє об'єднувати існуючі моделі для моделювання великомасштабних систем або розділяти великі моделі на моделі, які виконуються на окремих комп'ютерах. Серед переваг – повторне використання моделі, прискорення, конфіденційність і узгодженість даних. DS широко використовується в оборонному секторі. Однак, незважаючи на потенційні переваги, він рідко використовується в додатках дослідження операцій і управління у таких сферах, як виробництво та охорона здоров'я. Основними перешкодами для використання розподіленого моделювання в задачах дослідження операцій та управління є технічна складність впровадження та розрив між підходами. У цій роботі ми пропонуємо новий метод, який намагається поєднати методологічні практики розподіленого моделювання та досліджень операцій та управління. Використовуючи репрезентативне прикладне дослідження, ми показуємо, що запропонована методологічна основа значно спрощує впровадження цієї методології.

ABSTRACT

Master's thesis: 69 pages, 8 figures, 2 tables, 1 appendice, 52 sources.

SIMULATION, HIGH-LEVEL ARCHITECTURE, DISTRIBUTED PROGRAMMING

The purpose of the qualification work is research and development of the method of distributed simulation modeling in the software tools of operations and management research.

Distributed simulation allows you to combine existing models to simulate large-scale systems, or to split large models into models that run on separate computers. Benefits include model reuse, acceleration, data privacy and consistency. DS is widely used in the defense sector. However, despite its potential advantages, it is rarely used in operations research and management applications in areas such as manufacturing and healthcare. The main obstacles to the use of distributed modeling in operations research and management tasks are the technical complexity of implementation and the gap between approaches. In this work, we propose a new method that attempts to combine the methodological practices of distributed modeling and operations and management research. Using a representative case study, we show that the proposed methodological framework significantly simplifies the implementation of this methodology.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Розподілене імітаційне моделювання	11
1.2 Архітектура високого рівня HLA	14
1.3 Дискретне паралельне та розподілене моделювання	18
2 РОЗРОБКА ФРЕЙВОРКА РОЗПОДІЛЕНОГО МОДЕЛЮВАННЯ.....	26
2.1 Методологічні підходи до вибору фрейворку.....	26
2.2 Фази моделювання	27
2.2.1 Фаза планування.....	29
2.2.2 Фаза розвитку	30
2.3 Розробка методу методу розподіленого імітаційного моделювання в програмних засобах дослідження операцій та управління	32
2.3.1 Фаза планування.....	34
2.3.2 Фаза розвитку	34
2.3.3 Фаза експерименту.....	37
3 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ	39
3.1 Планування імітаційного моделювання	39
3.2 Етапи розробки експериментальної розподіленої моделі.....	43
3.2.1 Розподілена концептуалізація.....	43
3.2.2 Розробка об'єднаної моделі	45
3.2.3 Реалізація проміжного ПЗ	47
3.3 Проведення експериментів	49
ВИСНОВКИ.....	53
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	57
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

ABS – Agent-Based Simulation
ALSP – Aggregate Level Simulation Protocol
API – Application Programming Interface
CORBA – Common Object Request Broker Architecture
DCOM – Distributed Component Object Model
DES – Discrete-Event Simulation
DSEEP – Distributed Simulation Engineering and Execution Process
IEEE – Institute of Electrical and Electronics Engineers
HLA – High Level Architecture
MAS – Multi-Agent System
MPI – Message Passing Interface
NASA – National Aeronautics and Space Administration
OR/MS – Operations Research/Management Science
PDES – Parallel Discrete-Event Simulation
PVM – Parallel Virtual Machine
RTI – Run-Time Infrastructure
SD – System Dynamics
SISO – Simulation Interoperability Standards Organization
VIMS – Visualization, Information Modelling, and Simulation

ВСТУП

Протягом багатьох десятиліть дослідники та практики в галузі дослідження операцій та управління (OR/MS) використовували моделювання та симуляцію (M&S) для системного аналізу та прийняття рішень. Сила M&S полягає в здатності представляти реальну систему та вивчати вплив змін, фактично не змінюючи фізичну систему. Він дає уявлення про операційні процеси та може використовуватися для динамічного прогнозування, аналізу «що якщо» та оптимізації на стратегічному, тактичному та оперативному рівнях планування. OR/MS M&S використовується в широкому діапазоні сфер, включаючи бізнес, охорону здоров'я, виробництво та логістику, і охоплює низку методів від системної динаміки (SD) до моделювання дискретних подій (DES) і, нещодавно, моделювання на основі агентів (ABS). Ці методи підтримуються пакетами програмного забезпечення моделювання, що пропонують повне середовище моделювання, наприклад системи візуального інтерактивного моделювання (VIMS), які підтримують усі (або більшість) аспектів процесу досліджень. VIMS зазвичай працює на настільному комп'ютері та забезпечує зручний інтерфейс моделювання, який можна використовувати з мінімальними знаннями програмування. Моделі легко розробляються за допомогою інтерфейсу та підтримують широкий спектр системних елементів і атрибутів. Складні інструменти візуалізації та аналізу результатів підтримують експерименти та звітування.

Однак практики M&S OR/MS мають деякі обмеження, особливо коли моделюються великомасштабні системи. Один запуск моделювання може зайняти багато часу і тому може обмежити кількість експериментів, що виконуються в проекті. Можливо, симуляція більших моделей займає більше часу. Дійсно, в деяких випадках, коли модель має значні розміри і включає в себе багато об'єктів і подій, під час експериментування необхідно вдатися до значних компромісів під час використання одного настільного комп'ютера [1].

Крім того, великомасштабні системи зазвичай включають підсистеми із взаємозалежністю в одній організації або в автономних організаціях. Може бути зручним створити гібридну модель, яка використовує різні методи M&S для представлення різних систем [2]. Типовим підходом до побудови моделей OR/MS є використання єдиного пакету моделювання, що працює на одному комп'ютері. Необхідні певні компроміси, щоб мати справу з розміром і неоднорідністю великих систем. Для того, щоб мати можливість змодельовати велику модель за прийнятний час, розробники моделей OR/MS зазвичай підвищують рівень абстракції за рахунок деталізації, включеної в модель. Це може призвести до втрати точності результатів. Крім того, оскільки більшість пакетів моделювання підтримують лише одну техніку M&S, модель, як правило, розробляється з використанням однієї техніки моделювання (наразі лише один комерційний інструмент моделювання підтримує багаторівневе моделювання, Anylogic, www.anylogic.com). Необхідність використання гібридних підходів стає все більш очевидною, оскільки, наприклад, операції підприємства стають більш сумісними [3]. Сучасні підприємства функціонують у мережах, дедалі більше заглиблюючись у все більш взаємозалежні процеси. Отже, розробникам моделей OR/MS доводиться мати справу з більш складними, неоднорідними та великими процесами. Розробники моделей OR/MS повинні звертатися до все більших і більших систем без шкоди для точності, використовуючи зручні гібридні підходи.

Розподілене моделювання (DS) дозволяє об'єднувати існуючі моделі для моделювання великомасштабних систем або розділяти великі моделі на моделі, які виконуються на окремих комп'ютерах. Також можливо використання суперкомп'ютерів, комп'ютерних мереж, кластерів, приватних, публічних та гібридних обчислювальних хмар. Серед переваг – повторне використання моделі, прискорення, конфіденційність і узгодженість даних. DS широко використовується в оборонному секторі. Однак, незважаючи на потенційні переваги, він рідко використовується в додатках дослідження операцій і управління (OR/MS) у таких сферах, як виробництво та охорона

здоров'я. Основними перешкодами для використання DS в OR/MS є технічна складність впровадження та розрив між підходами спільнот DS та OR/MS. У цій роботі ми пропонуємо новий метод, який намагається поєднати методологічні практики OR/MS та DS. Використовуючи репрезентативне прикладне дослідження, ми показуємо, що наша методологічна основа значно спрощує впровадження DS для змішаних систем імітаційного моделювання складних систем та процесів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Розподілене імітаційне моделювання

Практика M&S в обороні та дослідженні космосу розвивалася іншим шляхом, ніж OR/MS. Застосування M&S у цих сферах, пов'язаних із навчанням, зосереджено на розробці симуляцій у реальному часі (для підготовки до ведення війни чи експлуатації космічних апаратів) [4, 5]. Із-за високій вартості цього моделювання та необхідності повторного використання моделей як компонентів більших систем, з початку 1990-х багато роботи було зосереджено на розподіленому моделюванні (DS) та його взаємодії. Одним із головних результатів цієї роботи була стандартизація способу взаємодії моделей, що призвело до широко відомого стандарту високорівневої архітектури (HLA) [6]. Вплив цього стандарту та його причина широкого застосування у вищезазначених областях полягає в тому, що моделі створюються як взаємодіючі компоненти великомасштабних моделей. Як показано на рисунку 1.1, логічні моделі взаємодіють безпосередньо, як якщо б вони були частиною однієї моделі.

Фізично моделі працюють на різних комп'ютерах (потенційно розташованих по всьому світу) і взаємодіють через комунікаційну мережу через програмне забезпечення (проміжне програмне забезпечення) на основі стандарту HLA. Проміжне програмне забезпечення називається інфраструктурою часу виконання (RTI). HLA має власну термінологію, яка може бути досить непрозорою для «сторонніх» – кожна модель є федератом, а набір моделей є федерацією. Федерати можуть бути перекомпоновані для створення різних федерацій. Кожен федерат виконує дії незалежно та може вибрати, якою інформацією будуть ділитися з іншими федератами у федерації.

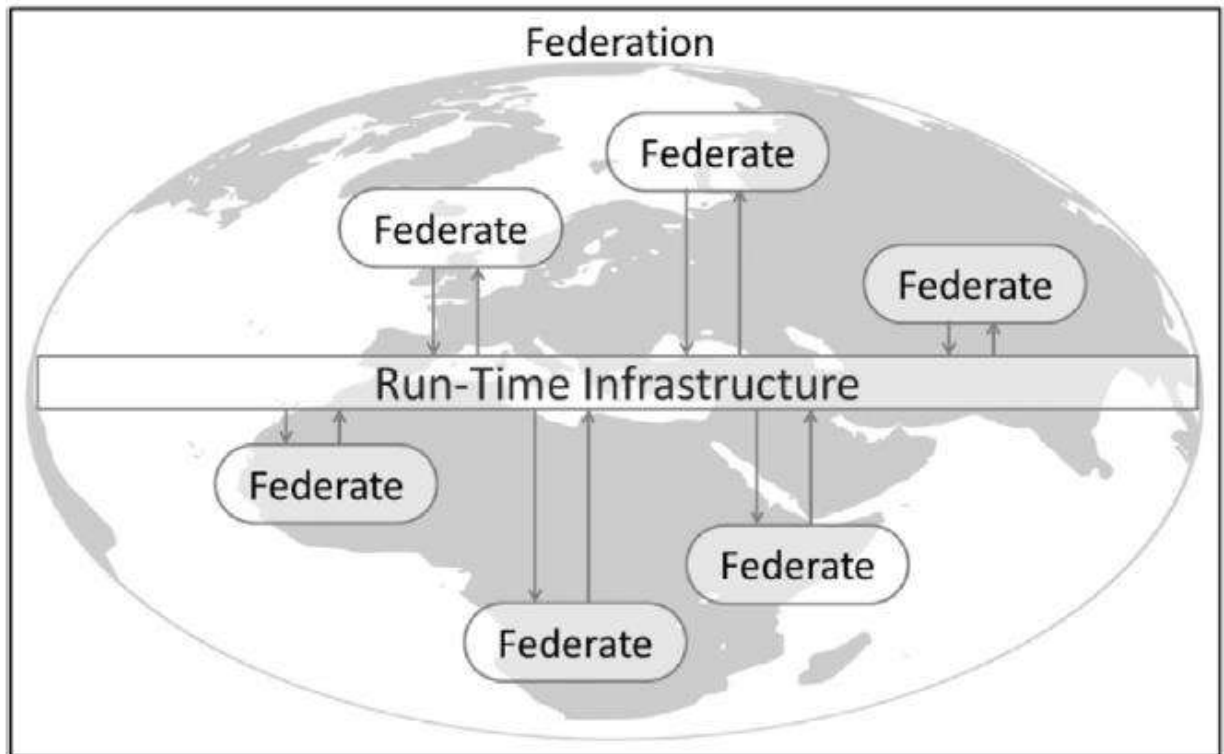


Рисунок 1.1 – Система розподіленого моделювання на базі стандарту HLA

Підходи M&S, які стали можливими завдяки HLA, можуть вирішити деякі проблеми моделі в OR/MS. Моделі можуть складатися з підмоделей, які працюють на ресурсах власних комп'ютерів без необхідності враховувати розподілену природу усій моделі. Подібним чином, моделі можуть бути створені в іншому програмному забезпеченні моделювання, що підтримує різні методи моделювання, якщо це доцільно. Ці моделі можна об'єднати разом як підсистеми більших моделей. Ця концепція використання HLA для підтримки великомасштабного моделювання OR/MS не нова. Очевидно, що розподілене моделювання може використовувати OR/MS як життєздатний підхід до моделювання для великомасштабних сумісних систем. Такі характеристики, як розподіл обчислювального навантаження в мережі та побудова незалежних сумісних симуляцій, можуть полегшити проблеми, згадані вище. Підсистеми всієї системи можуть бути змодельовані як об'єднані за допомогою найбільш прийнятної техніки моделювання. Ці

федерати можуть спілкуватися та обмінюватися даними на рівні інтерфейсу, зберігаючи свою індивідуальність і конфіденційність. Інтероперабельні федерати можна повторно використовувати в інших великомасштабних системах. Нарешті, кожне об'єднання може бути виконано у вузлі мережі локально або віддалено. Таким чином, DS може забезпечити рішення для недостатньої обчислювальної потужності шляхом розподілу навантаження по мережі, для гетерогенності шляхом формування гібридних федерацій і, отже, реалістичного охоплення всієї системи, для обміну інформацією між організаціями та конфіденційності шляхом вибору рівня прозорості та багаторазове використання моделей шляхом розробки сумісних і комбінованих моделей. Ці федерати можуть спілкуватися та обмінюватися даними на рівні інтерфейсу, зберігаючи свою індивідуальність і конфіденційність. Інтероперабельні федерати можна повторно використовувати в інших великомасштабних системах. Нарешті, кожне об'єднання може бути виконано у вузлі мережі локально або віддалено. Таким чином, DS може забезпечити рішення для недостатньої обчислювальної потужності шляхом розподілу навантаження по мережі, для гетерогенності шляхом формування гібридних федерацій і, отже, реалістичного охоплення всієї системи, для обміну інформацією між організаціями та конфіденційності шляхом вибору рівня прозорості та багаторазове використання моделей шляхом розробки сумісних і комбінованих моделей. Ці федерати можуть спілкуватися та обмінюватися даними на рівні інтерфейсу, зберігаючи свою індивідуальність і конфіденційність. Інтероперабельні федерати можна повторно використовувати в інших великомасштабних системах. Нарешті, кожне об'єднання може бути виконано у вузлі мережі локально або віддалено. Таким чином, DS може забезпечити рішення для недостатньої обчислювальної потужності шляхом розподілу навантаження по мережі, для гетерогенності шляхом формування гібридних федерацій і, отже, реалістичного охоплення всієї системи, для обміну інформацією між організаціями та конфіденційності шляхом вибору рівня прозорості та багаторазове використання моделей

шляхом розробки сумісних і комбінованих моделей.

Однак, незважаючи на дослідження підходів до створення цих OR/MS DS, які тривали більше десяти років, задокументовано лише дуже невелику кількість промислових прикладів. Основною причиною цього є не відсутність попиту, а складність і непрозорість HLA для практикуючих інженерів і дослідників OR/MS. Як зазначено у [7] існує недостатня підготовка в галузі програмної інженерії для академічних дослідників і практиків. У вчених немає чіткого розуміння процесів розробки програмного забезпечення. Однак потреба у формальних процесах і подальшій підготовці програмної інженерії є важливою, оскільки програмні інструменти стають більш складними. У цій роботі пропонується новий метод, який намагається поєднати методологічні практики OR/MS та HLA. Наша мета полягає в тому, щоб розширити практику OR/MS за допомогою чітко визначених і, де це доречно, перекладених елементів HLA для створення великомасштабних моделей. Тому ми представляємо методологічну структуру DS для додатків OR/MS, яка намагається подолати прірву між двома поглядами на світ. Шляхом виявлення відповідних практик двох найвідоміших структур життєвого циклу моделювання для світів OR/MS та DS, а саме методу моделювання OR/MS від Banks, стандартом якого є IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)[8], В роботі ми пропонуємо нашу методологічну структуру DS, використовуючи нотацію, звичну та широко прийнятну розробниками моделей OR/MS.

1.2 Архітектура високого рівня HLA

Розвиток архітектури високого рівня для моделювання та імітації (HLA) була ініційована Міністерством оборони США у 1995 році через потребу в загальній архітектурі моделювання високого рівня. Стандарт мав сприяти сумісності та багаторазовому використанню всіх типів моделювання, які використовуються та спонсоруються Міністерством оборони.

Необхідність стандарту впливає зі складності та різноманітності програм моделювання, що використовуються, і різноманітності очікувань щодо програм моделювання. Вони включають різні рівні абстракції, різні рівні інтерактивності, різну поведінку у часі тощо. По суті, жодна монолітна симуляційна програма не може задовольнити всі вимоги всіх користувачів.

Враховуючи різні додатки для моделювання, які використовуються, ніхто не міг заздалегідь передбачити їхнє потенційне використання та комбінації. Таким чином, народилася ідея модульного компонованого підходу для побудови об'єднань симуляцій, що зрештою призвело до розробки HLA.

Основною метою HLA було забезпечення відкритої архітектури, що пропонує послуги для сумісності та повторного використання. Архітектура не має обмежень щодо конкретної парадигми моделювання. Він навіть не обмежується додатками для моделювання, радше він пропонує взаємодію з усіма видами програм. Однак HLA надає спеціальні послуги підтримки взаємодії для задоволення конкретних потреб додатків моделювання.

Таким чином, HLA замінює загальні стандарти взаємодії, такі як CORBA або DCOM. Започаткований як стандарт у спільноті військового моделювання, розробка HLA контролювалася офісом оборонного моделювання та симуляції для Міністерства оборони США, який навмисно застосував дуже відкритий підхід до визначення та доступності HLA та спонсорував загальнодоступні програмні реалізації програмного забезпечення HLA.

Завдяки цій політиці було забезпечено широке залучення спільноти до розробки HLA, що можна розглядати як наріжний камінь для його досить хорошого сприйняття та впровадження. У сфері військового моделювання HLA є обов'язковим стандартом не лише в США, а й у більшості країн НАТО.

Залучення HLA до цивільної спільноти симуляторів здебільшого походить від академічних кіл [9] і було швидше дослідницьким. Значні зусилля були зосереджені на використанні HLA як стандарту для взаємодії між комерційними пакетами моделювання полиць [10].

Важливим спільним дослідницьким проектом у сфері цивільних застосувань HLA був проект IMS Mission. Основна увага зосереджена на прийнятті HLA як стандарту для проектування, планування та функціонування глобально розподілених підприємств. Одним із результатів стала концепція та рішення для моделювання розподіленого ланцюга поставок [11].

Серйозні практичні застосування HLA були досліджені кількома компаніями, серед яких Daimler Chrysler в автомобільному секторі. Особливо для автомобільної промисловості з її великими мережами постачальників і досить передовим використанням цифрових методів планування та моделювання в їх цифрові фабрики.

Докладаючи зусиль, HLA може відігравати значну роль у забезпеченні взаємодії симуляції plug-and-play.

Дослідження в галузі розподіленого моделювання мають давню традицію. Паралельне моделювання дискретних подій (PDES) є однією з важливих галузей розподіленого моделювання, яка в основному ведеться від спільноти цивільного моделювання, яка націлена на проблеми продуктивності та прискорення. Консервативні та оптимістичні протоколи синхронізації були розроблені для обробки можливих порушень причинності між симуляціями.

У спільноті військового симулятора розподілене інтерактивне моделювання (технологія DIS) була розроблена в основному для підключення тренувальних тренажерів у реальному часі.

DIS визначено в стандарті IEEE 1278 з 1993 року. Іншим стандартом для підключення конструктивного військового моделювання був протокол симуляції сукупного рівня (ALSP).

За межами області моделювання, стандарти для розподілених обчислень, як паралельна віртуальна машина (PVM) і інтерфейс передачі повідомлень (MPI), які також вплинули на сферу розподіленого моделювання. HLA поєднує в собі технології-попередники з військового сектору, DIS і ALSP, і є призначеною стандартною архітектурою для всіх видів моделювання та імітаційної діяльності Міністерства оборони США. Розробка HLA почалася в

1995 році з генерального плану моделювання та симуляції Міністерства оборони, який вимагав «створення загальної високорівневої архітектури моделювання для сприяння взаємодії всіх типів моделей і симуляцій, а також для полегшення повторного використання компонентів M&S».

Базове визначення HLA 1.0 Standard у серпні 1996 року можна вважати першим стабільним визначенням HLA. Незабаром після випуску різні версії програмного забезпечення RTI, розроблені за підтримки DMSO, стали загальнодоступними.

Це програмне забезпечення вільно поширювалося серед спільноти та включало службу підтримки RTI як інфраструктуру підтримки для підтримки програмного забезпечення. Наступним великим випуском стандарту HLA був HLA 1.3 у лютому 1998 року. Ця версія стандарту все ще досить часто використовується сьогодні в багатьох програмах моделювання. Після цього випуску стандарту 1.3 було зроблено дві розробки RTI публічно доступний у наступний час, першим був RTI 1.3 у 1998 році, наступним був RTI 1.3 NG у 1999 році.

Останній випуск запропонував покращену продуктивність і все ще доступний сьогодні з віртуальних технологій, як RTI NG Pro. Випуск HLA 1.3 став основою для подальших зусиль зі стандартизації. Серед них було прийняття OMG HLA як засобу для розподіленого моделювання.

Найважливішою діяльністю зі стандартизації став випуск версії IEEE стандарту HLA. Цей випуск здебільшого схожий на HLA 1.3, але містить кілька необхідних покращень, які виявились під час практичного використання HLA 1.3 [7]. Крім того, було внесено деякі зміни, необхідні для відповідності вимогам IEEE.

2002 рік ознаменував кінець перехідної фази, під час якої DMSO очолював (і спонсорував) зусилля з розробки HLA. Ставши стандартом IEEE, подальший розвиток HLA було передано в руки Організації стандартів сумісності моделювання (SISO). SISO виникла більше десяти років тому з Розподілене інтерактивне моделювання (DIS) Workshops і з того часу був

зосереджений на створенні стандартів для взаємодії моделювання. SISO — це волонтерська організація, членами якої є промислові, військові та наукові кола. Окрім розміщення Семінару з моделювання сумісності (SIW), які організуються тричі на рік (два в США, один в Європі). SISO проводить Комітет з питань стандартизації (SAC), яка контролює роботу кількох груп розробки продуктів (PDG). PDG – це фактичні групи людей, які розробляють стандарти взаємодії симуляції. Більшість їхньої роботи базується на HLA, включаючи його майбутнє вдосконалення та розробку стандартів.

Найважливішою PDG є HLA-Evolved Initiative оскільки він контролює перевірку специфікації IEEE 1516. Багато нових потенційних вимог до HLA було визначено на основі відгуків з різних областей і областей застосування. PDG прагне задовольнити ці вимоги за допомогою офіційного відкритий огляд IEEE 1516 серія специфікацій.

У рамках цього процесу PDG включатиме аспекти, підняті в Документі інтерпретацій Міністерства оборони для IEEE 1516 [8] і API HLA, сумісного з динамічним посиланням, для IEEE 1516.1.

1.3 Дискретне паралельне та розподілене моделювання

Дискретне паралельне та розподілене моделювання (DS) широко використовується в оборонному секторі та деяких сферах космічної промисловості (на чолі з NASA). Як зазначено нижче, є кілька прикладів успішного застосування DS в інших сферах. Однак занепокоєння, висловлене Strassburger et al. [12] та опитування Boer та ін. за 2004 рік [13], де експерти в галузі промислових M&S прокоментували труднощі DS, які відчуває галузь, звернули увагу на проблеми, які можуть обмежити широке впровадження DS. До них належать вартість програмного забезпечення RTI, складність DS, відсутність експертів з DS у промисловості та відсутність вказівок щодо включення функцій стандартів DS до промислових застосувань. Зовсім недавно Фудзімото [14] відзначає складність створення DS більш доступною

та складність створення цих програм. Це обмежене впровадження додатково підтверджується опитуванням, проведеним Фахімі та Пробертом [15], який проаналізував академічну літературу щодо методів OR/MS, що застосовуються в секторі охорони здоров'я Великобританії. Вони знайшли лише три видання, які застосовували DS. Вони з'явилися після 2006 року і включають одне дослідження. Слід зазначити, що в промисловості можуть бути й інші історії успіху DS, не пов'язані з обороною, про які не повідомлялося в первинній чи вторинній літературі.

Потенційно промисловість може багато чого отримати від DS, особливо для великомасштабних проектів моделювання. З появою хмарних обчислень M&S може отримати додаткові переваги від «необмеженої» розподіленої обчислювальної потужності на вимогу. Як можна заохотити розробників моделей OR/MS подолати вищезазначені перешкоди для впровадження? Один з перших кроків — почати говорити їхньою «мовою». Замість того, щоб очікувати, що розробники моделей OR/MS схилитимуться до практик програмної інженерії DS, ми можемо перевести ці практики на зручні та звичні для розробників моделей OR/MS.

В академічній літературі є кілька досліджень, які використовували DS для моделювання додатків в області OR/MS. Наприклад, наприкінці 1990-х Кляйн та ін.[25] розробив DS на основі HLA для симуляції трафіку. Вони повторно використали існуючі моделі, які були розширені для включення інтерфейсів HLA. Основна увага була зосереджена на аспектах сумісності та багаторазового використання HLA, тому вони не коментували продуктивність RTI або зусилля щодо розробки. Вони досягли сумісності платформи, мови та керування часом, використовуючи різні інструменти моделювання та анімації, поєднуючи масштабовані симуляції в реальному часі та орієнтовані на події. Функціональність симуляції трафіку DS може значно розширитися за допомогою сумісних модулів Географічних інформаційних систем (ГІС), програми DS, розробленої Бернардом та іншими [16]. У контексті управління надзвичайними ситуаціями вони розробили компонент ГІС на основі HLA,

щоб представити динамічний характер системи. Очевидно, що такі сумісні компоненти моделювання, які використовують загальні стандарти, підтримують повторне використання. Наприклад, одна і та ж HLA GIS може використовуватися в різних федераціях DS. Хан та інші [17] розробив пакет DS на основі HLA для морських систем. Їхня система, HAGGIS, включає різноманітні моделі та симуляції, такі як семантичні моделі, моделювання морського руху та моделювання датчиків, які можуть взаємодіяти через реалізацію RTI.

Klein та інші [18] і Bernard et al.[19] створили DS моделі DES та їх периферійних служб. В іншій галузі OR/MS, що стосується управління великими будівельними проектами, Taghaddos et al.[20] розробив додаток для моделювання HLA ABS і Multi-Agent System (MAS). Вони також включали планування та оптимізацію. Це незалежні федерації, які можна рекомпонувати для створення різних федерацій.

Знову ж таки, вони наголошують на необхідності розбивати великомасштабні системи на підсистеми та моделювати їх незалежно для більш ефективних експериментів і переробки моделей.

Однією з областей OR/MS, яка наполегливо розглядається для DS, є управління ланцюгом поставок (SCM). Ланцюжок постачання складається з великої мережі незалежних організацій, які повинні тісно співпрацювати, але бути слабо пов'язаними, кожна з яких може бути окремою моделлю. Більшість статей OR/MS про DS повідомляють про певну форму ланцюжка поставок. Бруццоне та ін. [21] розробив федерацію HLA ABS для загальнонаціонального ланцюга постачання в Італії, де федерати можуть вести переговори, реагувати та динамічно переносити події. Транспортна мережа також становить федерацію. Крім того, вони підняли питання про відсутність безпеки зв'язку в HLA та застосували протокол безпеки для автентифікації користувачів. DES DS був застосований Jain et al. [22] і Кацалиаки та ін.[24]. Вони змоделювали складні операції ланцюга постачання у виробництві та охороні здоров'я відповідно. Jain та ін.[23] запропонував DS для тестування

стандартів сумісності та відповідності додатків у постачанні ланцюгові інформаційні системи. На прикладі автомобільної промисловості вони створили віртуальне виробниче середовище за допомогою адаптера моделювання розподіленого виробництва (DMS), розробленого Національним інститутом стандартів і технологій (NIST). Адаптер DMS є спрощеною реалізацією стандарту HLA. DS включає федератив від різнорідного ланцюга постачання до процесів на виробництві з використанням Arena (www.arenasimulation.com) і Квест (quest.ucdavis.edu) програмне забезпечення моделювання відповідно. Програмне забезпечення Simul8 DES було використано Katsaliaki et al.[23] створити HLA DS для складного ланцюга кровопостачання у Великобританії. Вони вирішили звернутися до DS після того, як не змогли провести експерименти з симуляцією OR/MS усього ланцюжка поставок на настільному комп'ютері. Вони повідомили результати щодо продуктивності та дійшли висновку, що для дуже великомасштабного моделювання DS є життєздатним рішенням для ефективних експериментів. Медіна та ін. [24] розробив DES DS для ланцюга постачання залізної руди. Вони використовували програмне забезпечення Arena для розробки DES федератив імпорту та експорту морських портів. Зв'язок між компонентами моделювання та реалізацією RTI досягався за допомогою функцій DLL, розроблених на C++. Вони дійшли висновку, що досягнення сумісності та синхронізації годинника є найскладнішими завданнями. Розподілений агентний підхід був використаний Лонгом [25] для моделювання ланцюга поставок. Інфраструктура багатоагентних систем JADE (jade.tilab.com) використовувався для моделювання агентських компонентів мережі ланцюга поставок. Система має власну реалізацію координації та синхронізації з використанням агентів для регламентації завдань та координації часу. Автор визнає, що деякі компоненти запропонованого фреймворку все ще є складними, і необхідні подальші експерименти з реальними додатками. Ту та ін. [26] використовували сценарій ланцюга постачання вгору та вниз за виробництвом автомобілів DS, щоб продемонструвати свій підхід до взаємодії

підприємства. Вони використовували веб-сервіси HLA для виявлення моделі та створення об'єднання.

Значні зусилля зі стандартизації в області DES DS призвели до створення міжнародного стандарту Організацією взаємодії стандартів моделювання (SISO) SISO-STD-006-2010. Тейлор та ін. [27] розроблені еталонні моделі сумісності (IRM) для зв'язування моделей, розроблених у комерційних стандартних (COTS) пакетах моделювання. Намагаючись наблизити DS до практик OR/MS, цей стандарт визначив чотири типи IRM (Тип А. Передача сутності, Тип В. Спільні ресурси, Тип С. Спільна подія та Тип D. Спільна структура даних), що описують проблеми сумісності під час взаємодії DES моделі. Тип А описує способи, якими суб'єкт переходить від однієї федеративної моделі до іншої. Для того, щоб відобразити складність передачі сутностей у DES, які в основному складаються з черг і серверів, визначено три підтипи (Тип А1. Передача загальних сутностей, Тип А2. Обмежений приймаючий елемент і Тип А3. Пріоритизація кількох вхідних даних). Автори відзначили, що IRM можуть допомогти у розробці DS, якщо об'єднання було створено з використанням комерційних пакетів DES, оскільки вони роз'яснюють проблеми зв'язку між розробниками моделей OR/MS та експертами DS. Крім того, IRM допомагають скоротити час, необхідний для розуміння та перевірки способу взаємодії моделей. Стандарт не пішов далі, пропонуючи методологію DS OR/MS.

Впровадження вищезазначених стандартів IRM, а точніше типу А2 IRM, представлено Strassburger та ін. [28]. Автори розробили проміжне програмне забезпечення DS для існуючого моделювання тракторного заводу в Південній Америці. Було шість існуючих симуляцій для різних компонентів виробничого процесу. Відповідно до процесу деякі моделі компонентів живили наступні моделі компонентів. Таким чином, існує передача сутності між моделями. Проект мав на меті дослідити розміри вхідних буферів приймальних компонентів і потенційні наслідки блокування виробничих секцій через повні буфери. Для цього вимоги до підключення належать до специфікацій типу А2

IRM. Автори повідомили, що реалізація інтерфейсу DS була прямою після того, як вони мали справу з деякими проблемами представлення часу в моделях компонентів. Крім того, багато роботи було зроблено в дослідженні загальної проблеми для систем DS DES передачі об'єктів між двома або більше федераціями в промислових умовах Raab et al. [29] та Pedrielli та ін. [30], які створили середовища DS з використанням комерційних пакетів моделювання та впровадження з відкритим вихідним кодом і комерційної реалізації RTI відповідно. У цьому контексті вони дослідили один IRM, тип A.2 IRM SISO-STD-006-2010[31]. Тестовий приклад у Raab et al.[32] не представляє справжнього прикладу, однак це приклад типової виробничої лінії, яка використовується для отримання цікавих висновків для застосування, таких як складність і технічний досвід, необхідні для впровадження DS. Pedrielli та ін.[33], з іншого боку, використовував приклад виробництва листового металу. Проблема в останньому випадку полягала не в розмірі симуляції, а в доступі до інформації для всіх сумісних компонентів. DS дозволяє моделям підсистеми взаємодіяти, використовуючи лише інформацію, необхідну для інтерфейсу, тоді як конфіденційні дані можуть залишатися локально для організації-власника. У цьому випадку одна з фабрик-учасниць не буде ділитися ключовими даними, і тому відсутність спільної інформації унеможливило розробку моделювання OR/MS. Тому DS надав зручне рішення для моделювання інформації, якою обмінюються лише на рівні інтерфейсу.

З точки зору гібридного ABS-DES, де метою є взаємодія моделювання ABS і DES (для ознайомлення з ABS див. Macal і North [34], а для DES див. Robinson [35]), Мустафі та ін.[36] визначив основні підходи підключення, які існують на даний момент. Тобто вручну виконувати моделі та передавати відповідні змінні між ними (несинхронізоване ручне виконання), автоматизувати передачу змінних, але несинхронізованим способом (несинхронізоване автоматичне виконання), і використовувати пакети моделювання, які підтримують численні методи симуляції (синхронізоване виконання, кероване CSP), наприклад AnyLogic (

www.anylogic.com). У своїй роботі Mustafee et al. [37] звіт про моделювання офшорних вітрових електростанцій, що поєднують ABS Netlogo (ccl.northwestern.edu/netlogo) модель деградації турбіни та DES Simul8 (simul8.com) модель моделювання стратегій обслуговування, ремонту та експлуатації. Гібридна модель вимагала автоматизованого синхронного обміну даними з підтриманням порядку причинності. Оскільки моделі були розроблені в різних пакетах моделювання, неможливо було прийняти синхронізований підхід до виконання, керований CSP. Автори взяли на озброєння розробку DS як засіб автоматизованого та синхронізованого підходу для виконання гібридних моделей ABS-DES. Одночасно з цим підходом можливо використання графічних процесорів, як одної з складових частин середовища моделювання. Вони визначили взаємодію між моделі ABS і DES і розробили DS за допомогою HLA RTI. Середовище DS у контексті верифікації безпеки будівництва морських вітрових турбін було розроблено Laesche та ін.[38]. Автори запропонували симуляцію машин і людей як агентів і фізичного середовища як симуляцію фізичного світу за допомогою відкритого коду GameKit (github.com/gamekit-developers/gamekit).

З перерахованих вище робіт більшість, якщо не всі, представили докази концепції або техніко-економічні обґрунтування. Продукти не виходять за межі стадії прототипу. Більшість досліджень зосереджуються на технічних питаннях, таких як сумісність моделей, крос-платформна сумісність і продуктивність RTI, а не на реальних програмах. Ці дискусії виходять за рамки обмежень і занепокоєння розробників моделей OR/MS, і ця мова звучить чужорідно для їхнього сектора. Усі, більш-менш, описують шлях, яким вони розробляли свій DS, але жоден не згадує про систематичний шлях розробки моделей DS. У літературі немає жодної рекомендації, яку можна було б використовувати як керівництво для проведення проектів DS у світі OR/MS. Незважаючи на обіцянки DS щодо забезпечення неоднорідності, конфіденційності даних, автономності моделі, можливості повторного використання, і обчислювальних ресурсів було зроблено мало роботи для

визначення того, як вони будуть реалізовані в OR/MS. У наступному розділі ми намагаємося подолати цю «прірву» між поглядами на світ OR/MS і DS, відображаючи відповідні дії в обох практиках OR/MS і DS для проведення проекту моделювання. Це формує основу для запропонованої методологічної основи OR/MS DS, впливає на вибір моделей та методів імітаційного розподіленого моделювання та обумовлює структуру середовища моделювання. У наступному розділі буде розроблено архітектуру системи розподіленого сумісного моделювання.

2 РОЗРОБКА ФРЕЙВОРКА РОЗПОДІЛЕНОГО МОДЕЛЮВАННЯ

2.1 Методологічні підходи до вибору фрейворку

Запропонована методологічна основа OR/MS DS отримана шляхом поєднання добре встановлених методів, які використовуються як керівні принципи для проведення проектів OR/MS та DS відповідно. Існує багато опублікованих методів проведення симуляційного дослідження OR/MS. Їхня мета – надати вказівки моделювачам щодо управління різними видами діяльності проектів моделювання. Це включає в себе етапи та окремі кроки, яких слід дотримуватися, як правило, ітераційно. Роботи Ulgen et al., Бенкс та ін. [39], Ло і Келтон [40] і Робінсон [41] є серед них. У методах симуляції OR/MS етапи життєвого циклу проекту визначаються або описово, або схематично у формі блок-схем. Незважаючи на різні способи представлення процесів, загалом усі погоджуються з тим, що кілька етапів виконуються паралельно, вся справа є високоітеративною та що необхідно отримати три основні результати. Це план проекту, модель в тому числі концептуальна і комп'ютерна моделі, і результати експериментів. Вибір методу моделювання OR/MS, який буде використовуватися як основа для методологічної основи OR/MS DS, ґрунтувався на критеріях популярності, адекватного описового аналізу етапів моделювання та чіткого графічного представлення, яке також показує паралельні дії. Той, який відповідає нашим критеріям, запропонований Бенксом та ін. [39]. Представлення використовує позначення блок-схеми, а детальний опис кожного кроку супроводжує процес. Крім того, це один із найбільш цитованих методів і чітко демонструє паралельні дії концептуалізації моделі та збору даних.

Зусилля щодо надання вказівок для DS з використанням HLA призвели до розробки стандарту IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP) (IEEE 1516.3-

2003). FEDEP, розроблений SISO та стандартизований IEEE [42]. Він рекомендує загальні методи розробки DS за допомогою HLA. Потім їх можна пристосувати до потреб окремих програм. FEDEP — це замінений стандарт, замінений на DSEEP, який є структурою високого рівня, спрямованою на надання вказівок для розробки трьох архітектур DS і, як FEDEP, розроблений SISO та стандартизований IEEE [43]. Це сприяє HLA, розподіленому інтерактивному моделюванню (DIS) [44] та Архітектура, що сприяє тестуванню та навчанню (TENA) (www.tena-sda.org/display). DSEEP — це узагальнена структура, яку можна адаптувати для задоволення індивідуальних потреб додатків DS. Прикладом адаптації, адаптованої до вимог конкретних організаційних потреб, є структура VEVA, розроблена збройними силами Німеччини, як згадується у Siegfried et al. [45]. VEVA — це детальний інженерний процес, спрямований на полегшення конкретної організації. Обидва методи розроблені оборонним сектором і, хоча їх можна використовувати в додатках OR/MS, вони включають практики, знайомі в основному інженерам військового моделювання. Підхід до системної інженерії, який дотримуються, і мова, що використовується, роблять їх непривабливими для розробників моделей OR/MS. Оскільки DSEEP є єдиним загальним і активним стандартизованим процесом, він обраний як основа для частини DS нашої методологічної бази OR/MS DS.

2.2 Фази моделювання

Як було встановлено вище, два методи, вибрані для позначення світогляду OR/MS та DS у нашій методологічній структурі OR/MS DS, описані у Banks et al.[5] та DSEEP відповідно. Початкове порівняння двох методів призвело до трифазного процесу. Як і в OR/MS, проекти DS мають ті самі три основні результати. Таким чином, ідентифіковані три фази відповідають цим результатам і є етап планування, де план проекту виробляється, фаза розвитку, де модель розробляється, і фаза експериментування, де формуються

результати експериментів. У поданні верхнього рівня зіставлення діяльності OR/MS і DS у межах трьох фаз призвело до списку, показаного в Таблиці 2.1. Етап планування включає дії, пов'язані з розумінням і формуванням проблеми, а також встановлення цілей і визначення ресурсів для проекту моделювання. Фаза розвитку включає діяльність, пов'язану з розробкою моделі(ей), інтерфейсом і тестуванням. Нарешті, фаза експерименту включає дії, пов'язані з виконанням моделювання та отриманням результатів і аналізом.

Таблиця 2.1 – Діяльність OR/MS та DS на етапах планування, розробки та експериментування

Фази	Діяльність OR/MS	Діяльність DS
Планування	Формулювання проблеми Постановка цілей і загального плану проекту	Визначення цілей середовища моделювання
Реалізація	Концептуалізація моделі Збір даних Програмування моделі Перевірка	Виконання концептуального аналізу Проектування середовища моделювання Розробка середовища моделювання Інтеграція та тестування
Експерименти	Планування експерименту Проведення експерименту Документація та звітність	Виконайте моделювання Аналіз даних та оцінка результатів

Кожна з дій верхнього рівня включає детальні дії, які виконуються на кожній фазі. З метою розуміння точок конвергенції та розходжень між OR/MS і DS практиками, наступні підрозділи описуватимуть відповідні детальні дії на кожній фазі для обох підходів і висвітлять функціональні можливості для кожної. Відображення цих дій між методами OR/MS і DS і в межах трьох

визначених фаз показано на рисунку 2.1, де вказівні стрілки показують асоціацію. Деякі дії пов'язані як з OR/MS, так і з DS, тому існує конвергенція. Там, де є розбіжність, дії пов'язані лише з відповідним методом.

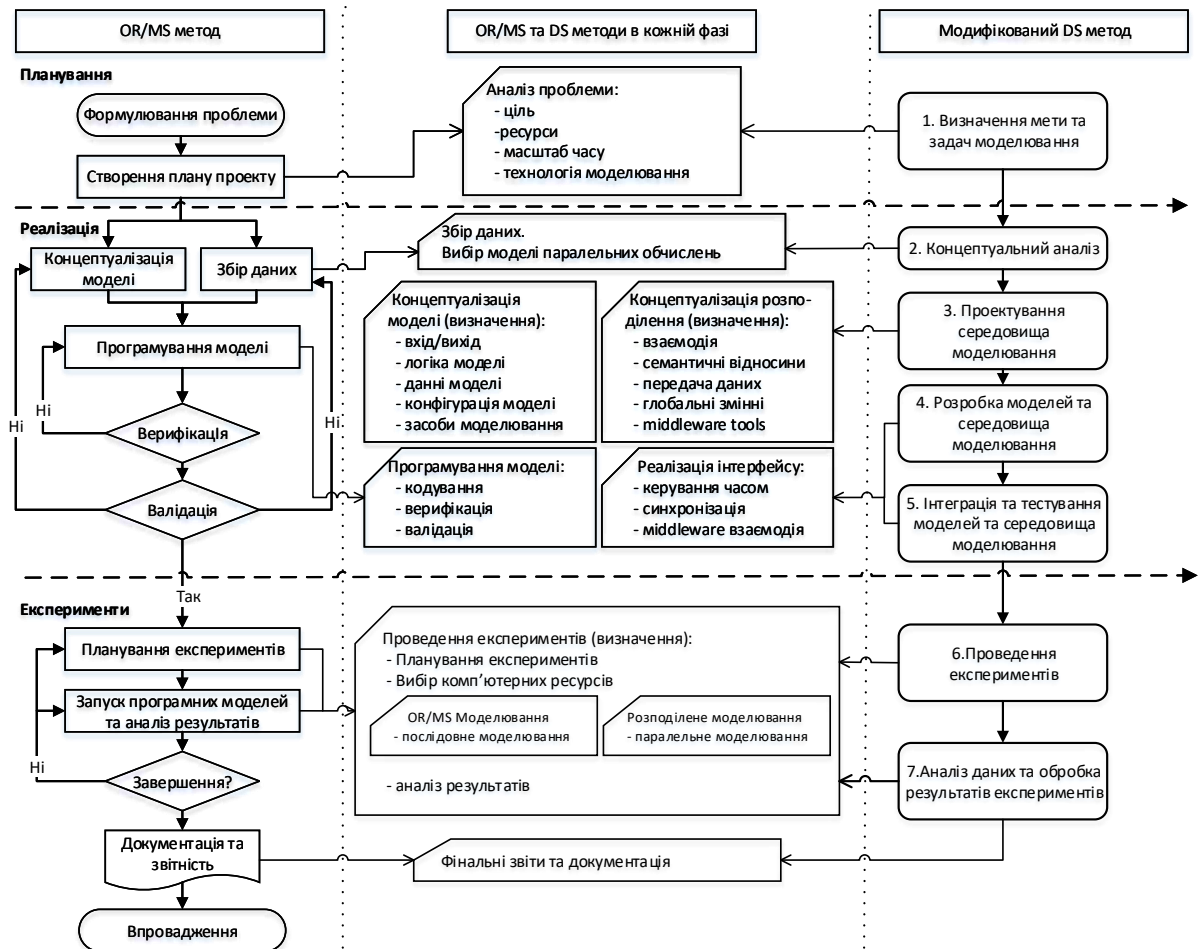


Рисунок 2.1 – Відображення активності методів OR/MS та DS у три фази

2.2.1 Фаза планування

Фаза планування моделювання DS по суті така ж, як і в проектах OR/MS. На цьому етапі відбувається формулювання проблеми. Тобто, щоб зрозуміти проблему реального світу та визначити цілі дослідження, також формується команда моделювання та узгоджується часовий масштаб і можлива вартість проекту. На цьому етапі також вирішується найбільш відповідна техніка моделювання. По суті, основною діяльністю на цьому етапі

є виконання планування проекту моделювання.

2.2.2 Фаза розвитку

На етапі розробки можна виділити три основні види діяльності, тобто збір даних, концептуалізацію та реалізацію. Найбільш фундаментальні відмінності нашого підходу в цій другій фазі.

Збір даних — це діяльність, яка відбувається подібним чином у проектах OR/MS і DS. Він включає контекстні дані високого рівня для концептуалізації моделі та детальні дані для розробки комп'ютерної моделі [46].

Концептуалізація та реалізація повинні розглядатися по-різному при розробці проектів OR/MS та DS. Концептуалізація моделі OR/MS включає такі дії, як визначення вхідних і вихідних даних, логіка моделі, припущення та спрощення. Однак у DS є ще одне міркування, яке необхідно вирішити; якою інформацією будуть обмінюватися моделі компонентів. Крім того, слід визначити прозорість об'єднань, глобальні змінні (наприклад, атрибути/параметри спільних (опублікованих) об'єктів/взаємодій) і право власності на змінні в об'єднанні. Крім того, у гібридній системі, де існує більше ніж одна техніка моделювання, має бути розуміння та документування семантичних зв'язків між різними техніками. Наприклад, фундаментальним елементом у DES є «сутність», яка є об'єктом, який протікає через змодельований процес і його стан змінюється під час дії. Якщо ця сутність переходить до ABS у DS, вона стає агентом – об'єктом, який може приймати рішення та змінювати свою поведінку відповідно до деяких внутрішніх правил. Дуже важливо розуміти ці зв'язки та відображати пов'язані дані для кожного елемента. Нарешті, програмне забезпечення для моделювання, яке буде використовуватися, має бути вирішено як у проектах OR/MS, так і в проектах DS. У проектах DS слід також вирішити, яка реалізація RTI буде використана. Дуже важливо розуміти ці зв'язки та відображати пов'язані дані для кожного елемента. Нарешті, програмне забезпечення для моделювання,

яке буде використовуватися, має бути вирішено як у проектах OR/MS, так і в проектах DS. У проектах DS слід також вирішити, яка реалізація RTI буде використана. Дуже важливо розуміти ці зв'язки та відображати пов'язані дані для кожного елемента. Нарешті, програмне забезпечення для моделювання, яке буде використовуватися, має бути вирішено як у проектах OR/MS, так і в проектах DS. У проектах DS слід також вирішити, яка реалізація RTI буде використана.

При реалізації моделювання слід розробити програмне забезпечення, яке реалізує концептуалізацію. Це може передбачати написання програми або створення моделі за допомогою пакетів COTS. В обох випадках необхідно також розробити компоненти DS. У DS на основі HLA, крім імітаційних моделей, це фактично означає налаштування компонентів RTI. Однак перед впровадженням RTI слід визначити стратегію управління часом. У DS на основі HLA перехід у час контролюється централізовано RTI; є дві служби керування часом, які наразі підтримуються HLA: Запит на завчасний час (TAR) і Запит на наступну подію (NER). Кожен федерат запитує від RTI збільшення часу, тоді RTI відповідає наданням Time Advance Grant (TAG), що дає федерації логічний час для прогресу. У TAR, федерат запитує перенесення часу на наступний часовий крок. Однак, коли федерація є DES, де підтримується список майбутніх подій, служба керування часом може бути реалізована як NER, де федерація запитує логічне просування часу на час, коли запланована наступна подія. Крім того, слід визначити протокол синхронізації. Консервативний протокол синхронізації гарантує, що RTI надсилатиме повідомлення з міткою часу, меншою за поточний час моделювання об'єднання. Це супроводжується витратами часу на синхронізацію. Оптимістичний протокол синхронізації працює краще з точки зору затримки. Оптимістичне об'єднання керування часом допускає порушення локальних обмежень причинно-наслідкового зв'язку, проте оптимістичні алгоритми забезпечують механізми відновлення помилок у випадку, коли сталася помилка причинності.

В проектах OR/MS, і в проектах DS код має бути перевірений, а симуляції перевірені. У симуляції OR/MS перевірка моделі зазвичай передбачає порівняння результатів із реальною системою або очікуваними результатами. Однак у DS причинно-наслідковий зв'язок і поведінку всієї системи також слід перевіряти. Необхідно провести перевірку на відповідність вимогам федеративної синхронізації та належного впровадження IRM.

2.2.3 Фаза експерименту

DS зазвичай виконується через мережу. Тому проект експерименту повинен передбачати цю функцію. Необхідно забезпечити доступність мережевих ресурсів розподілених обчислювальних інфраструктур і визначити вимоги до з'єднання, такі як пропускна здатність і безпека. Крім комп'ютерних ресурсів, план експерименту та аналіз результатів однакові в проектах OR/MS і DS M&S.

2.3 Розробка методу методу розподіленого імітаційного моделювання в програмних засобах дослідження операцій та управління

Кінцева мета полягає в тому, щоб об'єднати дії DSEEP у життєвий цикл моделювання OR/MS таким чином, щоб це було прийнятно для практик OR/MS. Запропонована методологічна основа отримана шляхом об'єднання аспектів методів OR/MS та DS після наведеного вище обговорення. Беручи до уваги необхідні дії з DSEEP і дещо модифікувавши метод моделювання OR/MS, розроблений Banks et al.[39], наша методологічна основа OR/MS DS проілюстрована на рисунку 2.2.

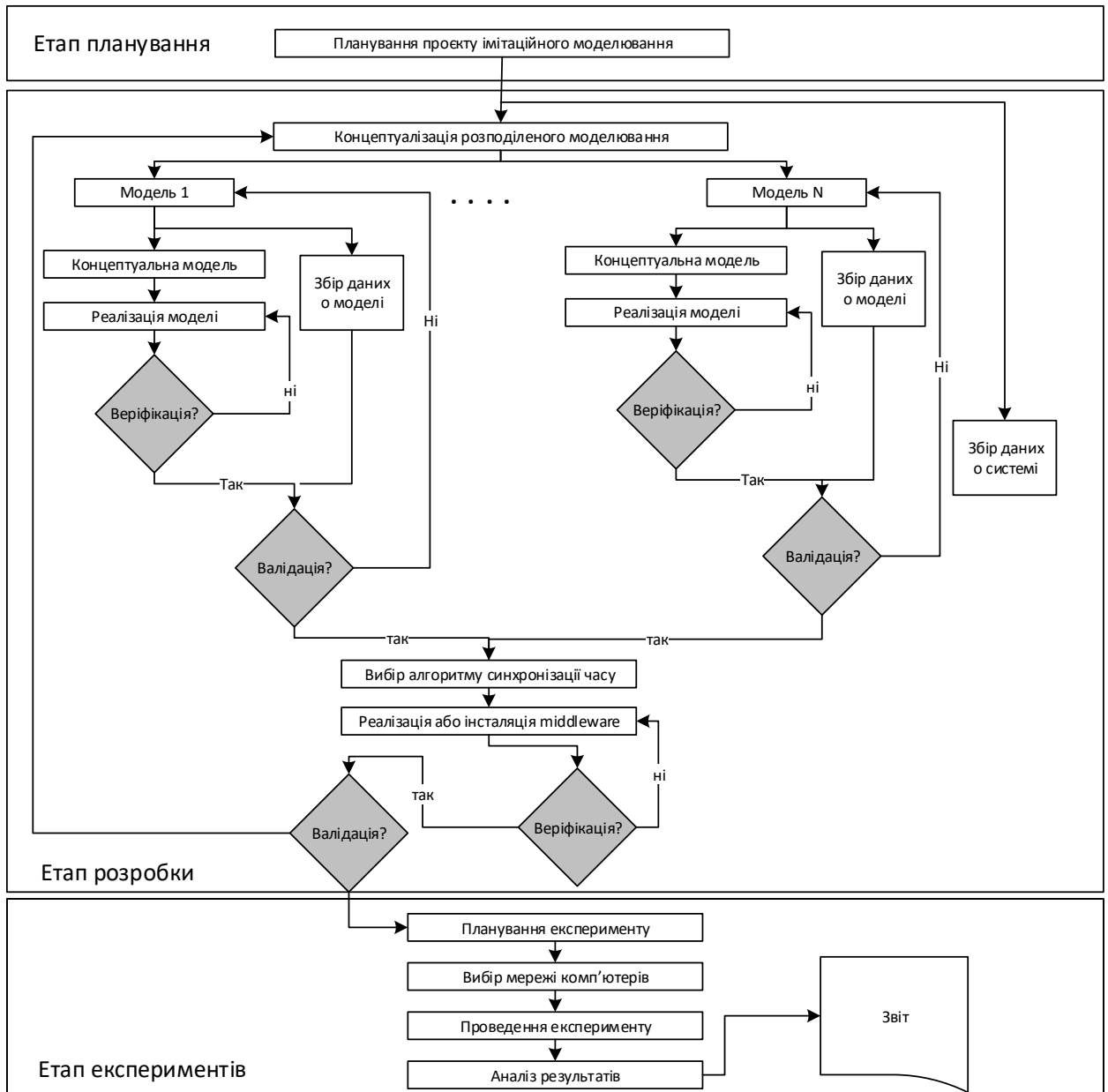


Рисунок 2.2 – Розроблений метод

Слід підкреслити, що процес документування починається з першого кроку і триває протягом усього проекту моделювання. Ця документація веде до остаточного звіту про проект. Невеликі зміни в методі OR/MS від Banks et al.[5]це дві паралельні види діяльності. А саме збір даних і процеси документування. Як показує досвід, програму комп'ютерного моделювання можна перевірити, навіть якщо модель не заповнена фактичними даними. Кодування моделі, а також процес збору даних може зайняти дуже багато часу. Виконуючи дві дії паралельно, можна заощадити багато часу. Крім того,

як згадували Онго та Хілл [47], ці дві діяльності тісно переплітаються. Наприклад, наявність, точніше, недоступність і якість даних можуть призвести до змін моделі. Подібним чином, паралельне документування кожного кроку може допомогати уникнути пропусків у остаточному звіті. Діяльність проекту OR/MS DS на кожній фазі описана нижче.

2.3.1 Фаза планування

На цьому першому етапі формується проект моделювання. Основним результатом є план моделювання проекту. Враховуючи дії в обох методах OR/MS і DS, основною діяльністю є планування проекту моделювання та включає завдання, пояснені нижче.

На цьому кроці спочатку слід визначити проблему, яку потрібно проаналізувати. У цьому беруть участь усі зацікавлені сторони, і необхідно переконатися, що всі залучені сторони, наприклад, менеджери/розробники політики та моделісти, досягають спільного розуміння та погоджуються щодо проблеми, яка буде змодельована, і чи справді моделювання підходить для аналізу проблеми. Потім слід встановити цілі проекту та визначити питання, на які має відповісти моделювальне дослідження, а також визначити ключові показники ефективності (KPI), тобто показники ефективності, які будуть виведені в результаті моделювання. Нарешті, визначаються необхідні ресурси, часові рамки та план дій і основні етапи. На цьому етапі вибирається відповідний метод(и) моделювання. Тут команда проекту повинна вирішити, чи можна використовувати DS.

2.3.2 Фаза розвитку

На етапі розподіленої концептуалізації має бути представлена концептуальна модель усієї системи. Це стосується компонентних моделей (федератів) та їх взаємодії. На цьому етапі немає необхідності включати деталі

кожної моделі. Важливими елементами є підсистеми, які представлятиме кожна модель, і зв'язок між ними. Тому окремі моделі можна представити як чорні ящики. Однак інтерфейс між кожною з взаємодіючих моделей має бути визначений. Іншими словами, IRM мають бути визначені, щоб описати, що і яким чином передають моделі. IRM включають обмінювану інформацію, але не обов'язково містять інформацію про синхронізацію часу. Часовий вимір згадується лише тоді, коли може виникнути конфлікт і порядок подій має відповідати вказаному відношенню. Тут слід зазначити, що стандартні IRM (SISO-STD-006-2010), які були розроблені з урахуванням вимог сумісності для DES, можуть бути легко перенесені в інше моделювання дискретного часу, наприклад, ABS. Однак, якщо DS є гібридним, тобто використовується більше ніж одна техніка моделювання, необхідно визначити семантичні зв'язки між різними техніками. Це необхідно для загального розуміння обміну інформацією. Зазвичай модель DS включає більше одного стандартного IRM. необхідно визначити семантичні зв'язки між різними техніками. Це необхідно для загального розуміння обміну інформацією. Зазвичай модель DS включає більше одного стандартного IRM. необхідно визначити семантичні зв'язки між різними техніками. Це необхідно для загального розуміння обміну інформацією. Зазвичай модель DS включає більше одного стандартного IRM.

Після цього виконується збір даних усієї системи. Ці дані містять детальну інформацію про всю систему. Діяльність зі збору даних починається паралельно з діяльністю розподіленої концептуалізації. Ці дані будуть використані для розробки та перевірки федерації.

Побудова моделей, які можна використовувати як об'єднані в HLA, не відрізняється радикально від процесу побудови моделі моделювання OR/MS. Тому ми пояснимо лише додаткові дії, необхідні для моделі DS. Крім того, якщо федеративні моделі існують, їх можна повторно використовувати з деякими модифікаціями для включення компонентів HLA.

Далі виконуємо концептуалізацію моделі. Додатковою вимогою в діяльності з концептуалізації моделі OR/MS під час розробки моделі як

федеративної є визначення рівня прозорості. Тобто, щоб позначити, яку внутрішню інформацію може бачити вся федерація. Ця діяльність керуватиметься розподіленою концептуалізацією та IRM. Потім слід визначити глобальні змінні. Значення глобальних змінних міститимуть значення, які потрібно оновити для надсилання або отримання від інших федеративів через RTI. Нарешті, слід визначити право власності на змінні. Тобто федерація може оновити значення.

Наступний крок - збір даних моделі. Ця діяльність передбачає збір даних для моделювання кожної підсистеми. Вона починається паралельно з концептуалізації моделі підсистеми. Якщо модель повторно використовується, ця дія може бути пропущена, якщо дані не оновлювалися з часу створення попередньої моделі.

Далі йде реалізація моделі. Ця діяльність передбачає моделювання кодування, перевірку створеного коду та валідацію моделі підсистеми, точно так само, як у моделюванні OR/MS. Визначаємо стратегію випередження часу. На основі HLA є дві служби керування часом. Залежно від дизайну кожного об'єднання та методів моделювання, які моделюються, можна вибрати TAR або NER. Крім того, тут слід визначити протокол синхронізації. Протокол синхронізації повинен відображати вимоги підтримки причинності системи DS. Як було сказано вище, протокол синхронізації може бути консервативним або оптимістичним.

Після цього реалізуємо проміжне ПЗ. Фактично, це кодування реалізації RTI. Абсолютно необхідними компонентами, які необхідно розробити, є протокол RTI, протокол взаємодії федеративів, об'єктна модель федерації (FOM) і імітаційна об'єктна модель (SOM). Протокол RTI відповідає за отримання повідомлень від RTI, протокол федеративів відповідає за надсилання повідомлень іншим федератам через RTI. Інформація, якою обмінюються під час виконання, визначається у FOM. FOM є загальним для всіх федеративів у федерації та визначає, якими об'єктами та їхніми атрибутами, якими взаємодіями та їхніми параметрами обмінюється через RTI. У HLA об'єкти є

фундаментальними елементами, характеристики (атрибути) яких можуть бути опубліковані суб'єктом федерації та на які інші федерати можуть підписатися. Взаємодії – це події, які можуть вплинути на інших федератів, тому їхні параметри можуть бути опубліковані федератом, а інші федерати можуть на них підписатися. Кожна федерація має власний SOM, який визначає обмін інформацією на рівні федерації. Якщо SOM сумісний із FOM, федерат може приєднатися до федерації. Перевірка передбачає тестування коду проміжного ПЗ. Перевірка DS відбувається після завершення всіх моделей компонентів і проміжного програмного забезпечення для перевірки всієї системи. Це передбачає перевірку синхронізації часу та причинності всієї моделі DS OR/MS. Перевірка передбачає тестування коду проміжного ПЗ. Перевірка DS відбувається після завершення всіх моделей компонентів і проміжного програмного забезпечення для перевірки всієї системи. Це передбачає перевірку синхронізації часу та причинності всієї моделі DS OR/MS. Перевірка передбачає тестування коду проміжного ПЗ. Перевірка DS відбувається після завершення всіх моделей компонентів і проміжного програмного забезпечення для перевірки всієї системи. Це передбачає перевірку синхронізації часу та причинності всієї моделі DS OR/MS.

2.3.3 Фаза експерименту

Експериментальна фаза включає дії з проектування та проведення експериментів з імітаційною моделлю, а також аналіз результатів і отримання результатів.

На початку виконується планування експерименту. Експериментальний дизайн для OR/MS DS включає рішення, подібні до будь-якого експерименту моделювання OR/MS. Модельєр тут вирішує, чи потрібен період ініціалізації чи розігріву, і якщо так, то скільки цей період триватиме. Крім того, визначається тривалість і кількість прогонів моделювання з використанням різних випадкових чисел. Крім того, план експерименту передбачає

визначення сценаріїв для експерименту, включаючи ідентифікацію вхідних параметрів і діапазону їх значень у визначених сценаріях. Додаткові відомості про дизайн експерименту OR/MS можна знайти в Bank et al.[39] і Робінзон [47].

Далі обираємо комп'ютерну мережу. У той час, як моделювання OR/MS зазвичай виконується на одному настільному комп'ютері, DS працює в комп'ютерній мережі. Тип мережі необхідно визначити відповідно до вимог проекту. Серед розглянутих вимог – місцезнаходження, конфіденційність і доступність обчислювальних ресурсів.

І наприкінці виконуємо аналіз результатів моделювання. Аналіз результатів у симуляції OR/MS включає числовий і графічний аналіз KPI, аналіз чутливості та рекомендацію рішення[42]. Коли система розроблена як DS, можна отримати уявлення про взаємозалежності підсистем. Аналіз результатів може включати вплив продуктивності однієї підсистеми на іншу підсистему або на всю систему.

3 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

3.1 Планування імітаційного моделювання

Як приклад моделювання складної системи ми обрали моделювання системи охорони здоров'я в Україні. Загалом, організації охорони здоров'я є складними, великими системами, найчастіше з неоднорідною динамікою. Багато ендогенних факторів у багатьох організаціях охорони здоров'я впливають на роботу всієї системи. Наприклад, на роботу служби швидкої допомоги серйозно впливає діяльність регіональних відділів екстреної допомоги, а на них, у свою чергу, впливає затримка виписки з лікарні тощо. Крім того, екзогенні фактори, такі як політико-економічний клімат, зростання та старіння населення, впливають на ресурси та тиск на систему охорони здоров'я.

У нинішньому економічному стані та постійному зростанні та старінні населення сектор охорони здоров'я стикається з проблемою надання високоякісних послуг з меншими ресурсами більшій кількості населення. Ефективне управління та перспективне планування, не лише на місцевому рівні, а радше в масштабах усієї системи, може підтримати сектор охорони здоров'я для подолання викликів.

Служби швидкої допомоги та відділення невідкладної допомоги (A&E) тісно пов'язані між собою в системі EMS. Комплексний аналіз повинен включати обидва. Тим не менш, аналіз таких великих і складних систем надзвичайно складний без належних інструментів прийняття рішень і системного аналізу.

Встановлення того, чи M&S підходить для аналізу проблеми. Одним із таких інструментів є M&S, який завдяки прогресу в обчислювальній техніці забезпечив значний теоретичний і технологічний прогрес. M&S використовується в ряді наукових дисциплін для вивчення нових систем або

змін до існуючих і відповідей на запитання «що, якщо», коли фізичну реалізацію системи важко або навіть неможливо досягти.

Природно, симуляція використовує історичні дані як вхідні дані та виводить оцінку поведінки системи з урахуванням випадковості системних змінних. Можливо, добре розроблене моделювання може досягти достатньої точності. Якість і достовірність результатів сильно залежать від якості вхідних даних і повторюваності експериментів. Іноді розмір експерименту залежить від доступних обчислювальних ресурсів. Сучасні обчислювальні інфраструктури, які пропонують велику ємність для зберігання та пам'яті, а також кілька процесорів за доступною ціною, відкривають нову перспективу в області M&S.

Таким чином, M&S, які можуть бути виконані в розподіленій інфраструктурі, вважаються бажаними інструментами для більш обґрунтованого прийняття рішень у масштабі систем EMS.

Визначивши, що M&S є підходящим підходом для аналізу такої складної системи, як EMS, метою дослідження було дослідити нові підходи до аналізу EMS з використанням M&S з дискретним часом, а головною метою було продемонструвати здійсненність цих підходів. KPI для всієї системи були показниками ефективності, такими як час виконання та масштабованість. Таким чином, був проведений аналіз досліджень M&S EMS з дискретним часом, нещодавно опублікованих у науковій літературі, щоб виявити прогалину в цій області.

Є багато статей, які представляють симуляційні дослідження служб швидкої допомоги. Ці дослідження зосереджені на різноманітних проблемах, з якими стикаються керівники служб швидкої допомоги, таких як стратегії відправлення, планування, покриття тощо. Наприклад, сценарії вдосконалення французької служби швидкої допомоги вивчали Aboueljlane та ін.[48] за допомогою DES. Вони розробили свою модель за допомогою комерційного програмного забезпечення для моделювання Arena. Ібрі та ін.[49] змоделювали служби екстреної медичної допомоги в Швейцарії за допомогою

MAS. Послуги швидкої медичної допомоги були змодельовані Лі та ін. [50] використання корейських даних про випадки травм.

Мы використовували DES з моделюванням математичної оптимізації, розробленим на C#. Ван Бюрен та ін. [51] розробив симулятор швидкої медичної допомоги для нідерландської служби швидкої допомоги в регіоні Амстердама. З метою переносимості вони використали мову програмування C++ для реалізації симулятора. Це симуляція з дискретним часом, однак вона може працювати у швидкісному режимі як DES, де підтримується впорядкований список поточних і майбутніх подій.

У той же час, багато симуляційних досліджень зосереджені на роботі відділів невідкладної допомоги. Наприклад, Ванга [52] представив прототип моделі ABS, розробленої в Netlogo, для відділення невідкладної допомоги в університетській лікарні, штат Вірджинія. Автор стверджує, що ABS може бути легше прийнято менеджерами охорони здоров'я, ніж складні математичні моделі. DES з використанням комерційного програмного забезпечення для моделювання FlexSim Healthcare (www.flexsim.com/flexsim-healthcare) був використаний для моделювання відділення невідкладної допомоги університетської лікарні Акерсхус Норвегія. Модель розроблено для вивчення ефекту очікуваного збільшення обсягу пацієнтів.

З літератури було зроблено два очевидні висновки; по-перше, у багатьох дослідженнях обговорюються моделі, розроблені для служб швидкої допомоги та відділень невідкладної медичної допомоги з певною метою, і немає плану повторного використання моделі; по-друге, бракує моделей, які об'єднують усі компоненти EMS як цілісної системи.

Більшість симуляцій швидкої медичної допомоги лише визнають переваження швидкої медичної допомоги. Однак мало свідчень з літератури вказує на те, що вся система EMS і взаємодія між різними підсистемами були належним чином вивчені. Це пов'язано з кількома причинами.

По-перше, розробка всіх залучених моделей вимагає величезних зусиль

від розробників моделей, по-друге, необхідні дані можуть бути недоступними, по-третє, виконання таких великих і складних симуляцій є дуже дорогим з точки зору обчислень. Таким чином, DS може бути дійсно корисним для моделювання EMS, визначення відповідних методів M&S.

Два та основні методи M&S, які зазвичай використовуються для аналізу охорони здоров'я в організаційному контексті: DES і ABS. DES і ABS — це методи мікроскопічного моделювання, які характеризуються високим рівнем деталізації.

Зазвичай у DES часові кроки не є фіксованими, але є часові випередження до часу моделювання, коли запланована наступна подія. Двигуни DES зберігають список майбутніх подій для цієї мети.

Час ABS прогресує з фіксованими кроками. І DES, і ABS можуть моделювати систему на низькому рівні окремих об'єктів. Об'єкти DES – це об'єкти, які проходять процес, і їхній стан змінюється в міру розвитку часу моделювання.

Агенти ABS – це об'єкти, які «живуть» у середовищі та взаємодіють з ним і один з одним. Агенти мають здатність навчатися та змінювати свою поведінку. Через процесно-орієнтований характер невідкладної допомоги, DES було обрано для моделювання лікарні.

З іншого боку, модель служби швидкої допомоги може виграти від автономії агентів та їх здатності взаємодіяти з оточенням і коригувати свою поведінку; тому для служби швидкої допомоги було обрано ABS.

На високому рівні показано схематичне зображення федерації (рисунок 3.1) DES і ABS є високостохастичними методами, де виконується кілька реплікацій для підвищення достовірності результатів моделювання. Через недоступність існуючих моделей всі федерати були розроблені заново.



Рисунок 3.1 – Федерати розподіленої системи моделювання

3.2 Етапи розробки експериментальної розподіленої моделі

3.2.1 Розподілена концептуалізація

У гібридній моделі EMS використовуються два методи моделювання, це ABS і DES. Між ABS і DES існують семантичні відмінності, тому було визначено семантичне відношення. Наприклад, агент в ABS може бути сутністю або ресурсом у DES. Черга в DES не моделюється явно в ABS. Важливо, що DES керується подіями, тоді як ABS є моделюванням, керованим часом. Щодо IRM, які можуть визначати взаємодію між моделями компонентів, у EMS можна ідентифікувати три типи обмінюваної інформації. По-перше, усі моделі відділень швидкої допомоги повинні мати можливість повідомляти про свою доступність моделі швидкої допомоги. Для функціональності системи важливо, щоб коли модель швидкої допомоги

шукала найбільш підходящу лікарню для транспортування пацієнта, найновіша інформація про всі наявні відділення була відома службі швидкої допомоги. По-друге, об'єкт/агент пацієнта слід перенести з моделі служби швидкої допомоги до точки входу об'єкта/сутності пацієнта швидкої допомоги моделі відділення швидкої допомоги. По-третє, модель служби швидкої допомоги повинна мати можливість повідомити модель відділення швидкої медичної допомоги про те, що пацієнта переводять, щоб зарезервувати ресурси та уникнути конфлікту щодо рівня доступності лікарні. Крім того, об'єкт пацієнта та всі його властивості повинні бути відомі моделі екстреної медичної допомоги. Таким чином, у цьому проекті взаємодія між двома частинами гібридної моделі може бути представлена кортежем Type(A.1, C, D) або Type(загальна передача сутності, спільна подія, спільна структура даних) SISO- 006-2010, як показано на рисунку 3.2.

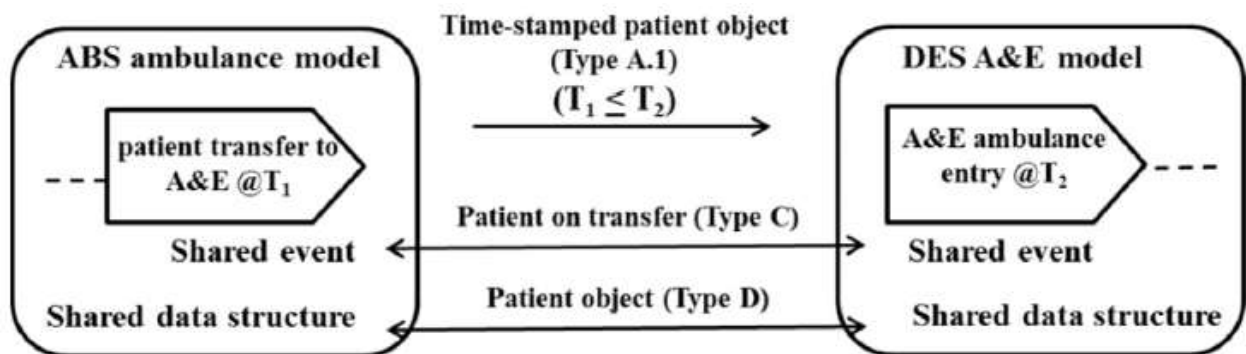


Рисунок 3.2 – Взаємодія федератив

Існує часове обмеження, яке має бути визначено в A.1 IRM. Тобто час, протягом якого пацієнт залишає модель ABS LAS, завжди повинен бути меншим або дорівнювати часу, коли той самий пацієнт прибуває до однієї з моделей DES A&E. Програмні засоби, вибрані для цього проекту, були з відкритим вихідним кодом як для розробки моделі, так і для реалізації RTI. Як симулятор використовувався інструментарій Repast Symphony (RepastS). RepastS — це, по суті, набір інструментів ABS, однак його можна легко

перетворити на симулятор DES, оскільки і ABS, і DES — це симуляції дискретного часу. Для цього фундаментальні компоненти DES, а саме черги, робочі станції та ресурси, були жорстко закодовані. Для інтерфейсу, реалізація Portico v2.0 RTI IEEE-1516-2000 (www.porticoproject.org).

Усі дані в цьому проєкті було зібрано з загальнодоступних онлайн-архівів за 2020–2021 роки. Процес збору даних можна розділити на дві частини. По-перше, потрібні дані для всієї федерації. У нашому прикладі нам потрібні дані про основні станції швидкої допомоги. Це кількість пунктів швидкої та невідкладної допомоги в регіоні та їх точне розташування, а також тип послуг, які вони надають, наприклад, чи є вони загальними чи спеціалізованими пунктами швидкої допомоги та чи мають вони педіатричні відділення. У період збору даних у зоні охоплення швидкої допомоги (LAS), було 62 пункти невідкладної допомоги. Друга частина містить дані для кожної лікарні: загалом 182 лікарні.

3.2.2 Розробка об'єднаної моделі

У системі швидкої медичної допомоги ми маємо дві організації: службу швидкої допомоги та лікарні. Що стосується цілей проєкту, тут визначається рівень деталізації для федеративних моделей. У цьому прикладі та оскільки метою проєкту було дослідити нові підходи до аналізу EMS з метою продемонструвати їх реалізованість, обидві системи були змодельовані з низьким рівнем деталей. Було зроблено припущення, що швидка допомога завжди має доступні ресурси, коли прибуває пацієнт швидкої допомоги, тому час передачі не моделювався.

Як згадувалося раніше, дані про прибуття пацієнтів були зібрані з онлайн-ресурсів, тоді як дані про час активності та ресурси були оцінені з використанням попереднього досвіду моделювання невідкладної допомоги. Ключові показники ефективності були часом очікування, оскільки це міра ефективності для A&E. Різниця в дизайні для підтримки індивідуального

виконання, а також для взаємодії в DS полягала в прибутті швидкої допомоги. Під час виконання симуляції операційної/МС пацієнти надходять із використанням відповідного розподілу, тоді як під час роботи як федерації прибуття швидкої допомоги здійснюється з моделі ABS. Кожен A&E обмінюється інформацією лише з моделлю LAS. Лікарні не знають одна про одну. Усі внутрішні процеси швидкої допомоги приховані від моделі лікарні, окрім її доступності, ресурсів і прибуття швидкої допомоги. Вхідні дані для моделей DES відділів швидкої допомоги показано в таблиці 3.1.

Таблиця 3.1 – Вхідні дані моделі A&E

Нормальний розподіл часу між прибуттями		Стан хворого	
Середній	10.9	Не потрібна лікарня	55%
Відхилення	4.2	Потрібна лікарня	45%
Нормальний розподіл часу на транспортування		Потрібне лікування	
Середній	15.6	Так	35%
Відхилення	5.3	Ні	65%
Нормальний розподіл часу на прийом		Ресурс прийомного пункту	
Середній	20.1	Чисельність персон.	5-15
Відхилення	11.7	Ємність прийому	3-5
Нормальний розподіл часу на спеціаліста		Наявність місці у лікарні	
Середній	34.67	Основні	5-60
Відхилення	15.3	Спеціальні умови	2-10

Для моделювання ABS ми OR/MS дотримувалася стандартної процедури для моделювання ABS. Середовище та топологія були визначені, агенти та їх взаємодія були визначені. Лікарні присутні в моделі швидкої допомоги як частина середовища. Після того, як пацієнта перевели до лікарні, при виконанні як симуляції OR/MS час лікування вибирається з розподілу, потім пацієнта виписують і доступність лікарні обчислюється локально; після

розподілу пацієнт переходить до моделі A&E DES і знімається з моделі ABS. Так само, як і в моделях швидкої медичної допомоги, керівники лікарень не знають внутрішньої логіки моделі швидкої допомоги, окрім пацієнта, якого передають, і його властивостей. LAS, у період збору даних, мають 70 станцій швидкої медичної допомоги та 375 автомобілів швидкої медичної допомоги. Інші типи транспортних засобів, такі як двоколісні транспортні засоби та автомобілі швидкого реагування, не включені в модель. Середня швидкість відносно низька: від менше ніж 25 км. на годину в центрі до 40 км. на годину в більшій частині міст. У моделі враховується середня швидкість 30 км/год. Крім того, відстань у міських умовах можна розрахувати з високим ступенем достовірності за допомогою евклідової відстані з коригуючим коефіцієнтом [23]. Експерименти проводилися з використанням об'єднання розміром приблизно в половину LAS.

Як згадувалося вище, це гібридна федерація, яка складається з моделей DES і ABS. У ABS, як у техніці, що керується часом, не підтримується список подій. Виходячи з цього, розширена стратегія часу, яка підходить конкретній федерації, полягає в запровадженні служби керування часом TAR. У цьому дослідженні був реалізований консервативний протокол синхронізації.

3.2.3 Реалізація проміжного ПЗ

У цьому прикладі кожна федерація A&E має унікальний ідентифікатор, який відповідає A&E у середовищі ABS. Опублікованими атрибутами моделі швидкої допомоги є ідентифікатор вибраного A&E, стан пацієнта під час передачі та час, коли швидка допомога з пацієнтом прибуває до вибраного A&E. Моделі A&E підписуються на наведені вище атрибути, щоб отримувати інформацію. У свою чергу, моделі швидкої допомоги публікують свою доступність, а модель швидкої допомоги підписується на цей атрибут. Уся ця інформація визначена в документі FOM XML. У цій конкретній реалізації SOM не були визначені. Portico є повністю

децентралізованою реалізацією. Єдина вимога полягає в тому, щоб кожен федерат мав змогу отримати доступ до бібліотеки Portico та реалізувати модуль RTI, який включає класи RTI Ambassador і Federate Ambassador. Протокол RTI відповідає за надсилання оновлень (опублікованих атрибутів/параметрів), а протокол федератов за отримання оновлень (підписаних атрибутів/параметрів) через RTI. Щоб досягти інтеграції, екземпляри RTIAmbassador і протокол федерата додаються в контекст RepastS під час ініціалізації об'єднання (за допомогою RepastSContextBuilder Інтерфейс). Для публікації оновлень у раніше визначених атрибутах, updateAttributeValues() метод RTIAmbassador використовує клас. Перш ніж робити фактичні оновлення, ми оголосили змінні дескриптора, які зберігають значення атрибутів, які будуть передані, і додали їх до атрибута Колекція. Метод оновлення в RTIAmbassador для федерації швидкої допомоги показано в лістингу 3.1.

Лістинг 3.1 – Метод оновлення федерації

```

1. public void updateAttributeValues(int hID, double arrivalTime,
2. int pCondition) throws RTIException
3. {
4. AttributeHandleValueMap attributes =
   rtiamb.getAttributeHandleValueMapFactory(). create(3);
5. HLAinteger32BE aaValue =
   encoderFactory.createHLAinteger32BE(hID);
6. HLAinteger32BE abValue =
   encoderFactory.createHLAinteger32BE(pCondition);
7. HLAfloat64BE aeValue =
   encoderFactory.createHLAfloat64BE(arrivalTime);
8. attributes.put(aaHandle, aaValue.toByteArray());
9. attributes.put(abHandle, abValue.toByteArray());
10. attributes.put(aeHandle, aeValue.toByteArray());
11.
12. HLAfloat64Time time =
   timeFactory.makeTime(fedamb.federateTime + fedamb.
   federateLookahead);
13. rtiamb.updateAttributeValues(objectHandle, attributes,
   generateTag(), time);
14. }

```

Крім того, інтерфейс федерата несе відповідальність за отримання

оновлень змінних, на які ми підписалися, та їх декодування. Це реалізовано в `reflectAttributeValues()` методі. Наприклад, федерація швидкої допомоги підписується на доступність лікарні. Федерація швидкої допомоги також надсилає свою ідентифікацію. Це реалізовано у федераті (лістінг 3.2)

Лістінг 3.2 – Метод оновлення федерації

```

1. for(AttributeHandle attributeHandle: theAttributes.keySet())
2. {
3.   if(attributeHandle.equals(federate.miHandle))
4.   {minorHosAvailability =
decodeInt(theAttributes.get(attributeHandle));}
5.   if(attributeHandle.equals(federate.maHandle))
6.   {majorHosAvailability =
decodeInt(theAttributes.get(attributeHandle));}
7.   if(attributeHandle.equals(federate.idHandle))
8.   {HospitalID = decodeInt(theAttributes.get(attributeHandle));}
9. }

```

Запит на випередження часу реалізовано в класі `RTIAmbassador` з використанням `advanceTime()` метода. Цей метод анотовано як `@ScheduleMethod()` і тому додається до планувальника `RepastS`.

Процес перевірки проміжного програмного забезпечення виконувався ітеративно за допомогою програмування модуля RTI. Перевірка проміжного програмного забезпечення була здійснена шляхом відповідності унікальних ідентифікаторів швидкої медичної допомоги в середовищі ABS і обраної федерації лікарні, атрибутів пацієнта швидкої допомоги та часу прибуття до швидкої медичної допомоги як у моделях ABS, так і в обраних моделях DES.

3.3 Проведення експериментів

Як було визначено на етапі планування, основною метою цього дослідження було продемонструвати можливість використання DS для підходу M&S EMS. Таким чином, експериментальний дизайн включав тестування продуктивності та масштабованості. Для тестування

продуктивності проведені експерименти включали двовимірне збільшення розміру моделювання; по-перше, за рахунок збільшення кількості федеративів у федерації, по-друге, за рахунок збільшення часу моделювання. Для тестування масштабованості експерименти включали збільшення кількості об'єктів (екстрені виклики та прибуття швидкої допомоги) і, як наслідок, подій у федерації.

Виконання об'єднання виконується в однорідній невиділеній мережі, з'єднаній через локальну мережу, з'єднану з мережевою картою 1 Гбіт/с. Кожен вузол мав процесор i5-2500 із частотою 3,30 ГГц і 4,00 ГБ оперативної пам'яті під керуванням Microsoft Windows 10 із встановленою Java 1.7 JRE та Portico v2.0.

Аналіз результатів включав графічне представлення результатів продуктивності та масштабованості та розрахунок прискорення, які проілюстровано на рисунках 3.3-3.5 відповідно. Рисунок 3.3 показує час виконання в порівнянні з кількістю федеративів.

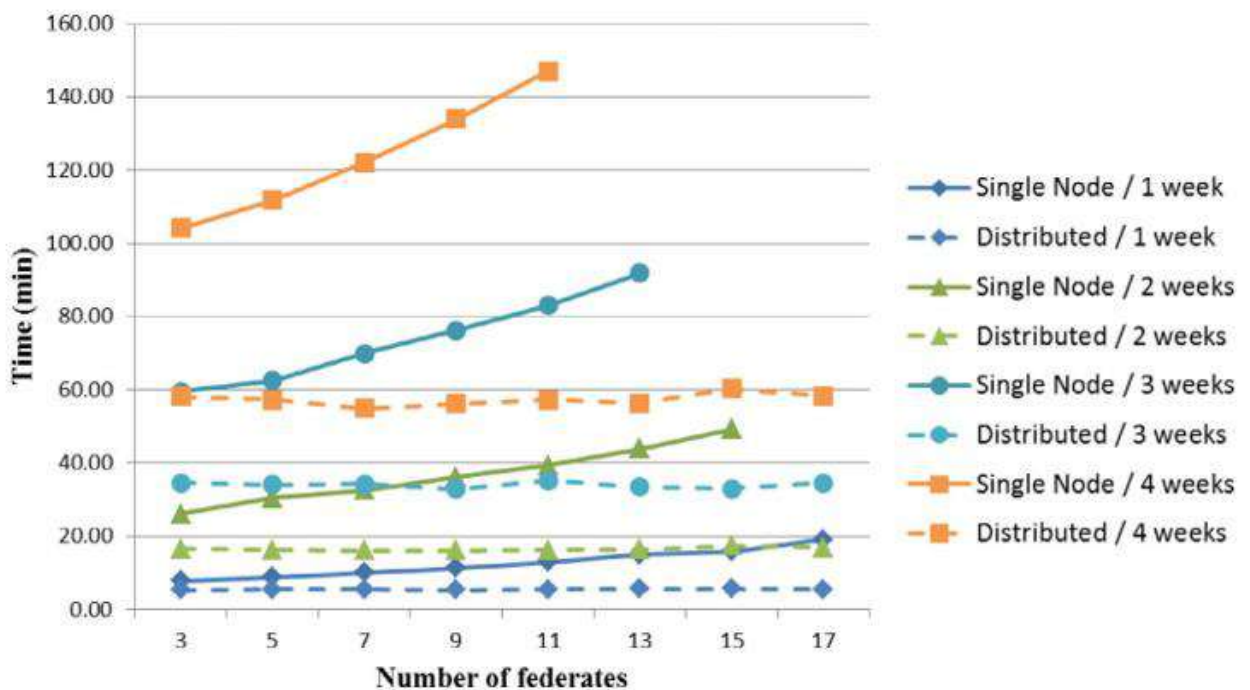


Рисунок 3.3 – Час виконання (кількість тижнів виконання) в залежності від кількості федеративів

Як видно, еквівалентна одиночна симуляція обмежена розміром/часом виконання (чотири тижні обмежують одну симуляцію еквівалентом 11 федеративів). Розповсюджена версія завжди працює краще, при цьому найкраща різниця у виконанні – це 4-тижневий запуск. Час виконання окремої симуляції поступово збільшується за допомогою додаткових симуляцій (поки не буде досягнуто можливості окремого вузла). Час виконання розподіленого моделювання залишається приблизно таким же. Це відображає взаємодію між симуляціями екстреної медичної допомоги та швидкою допомогою – федерація швидкої допомоги жодного разу не стає вузьким місцем, а федерати екстреної медичної допомоги можуть обробляти своє робоче навантаження паралельно.

Рисунок 3.4 демонструє масштабованість робочого навантаження. Реалістично досліджувати чутливість систем невідкладної допомоги, досліджуючи вплив додаткових пацієнтів.

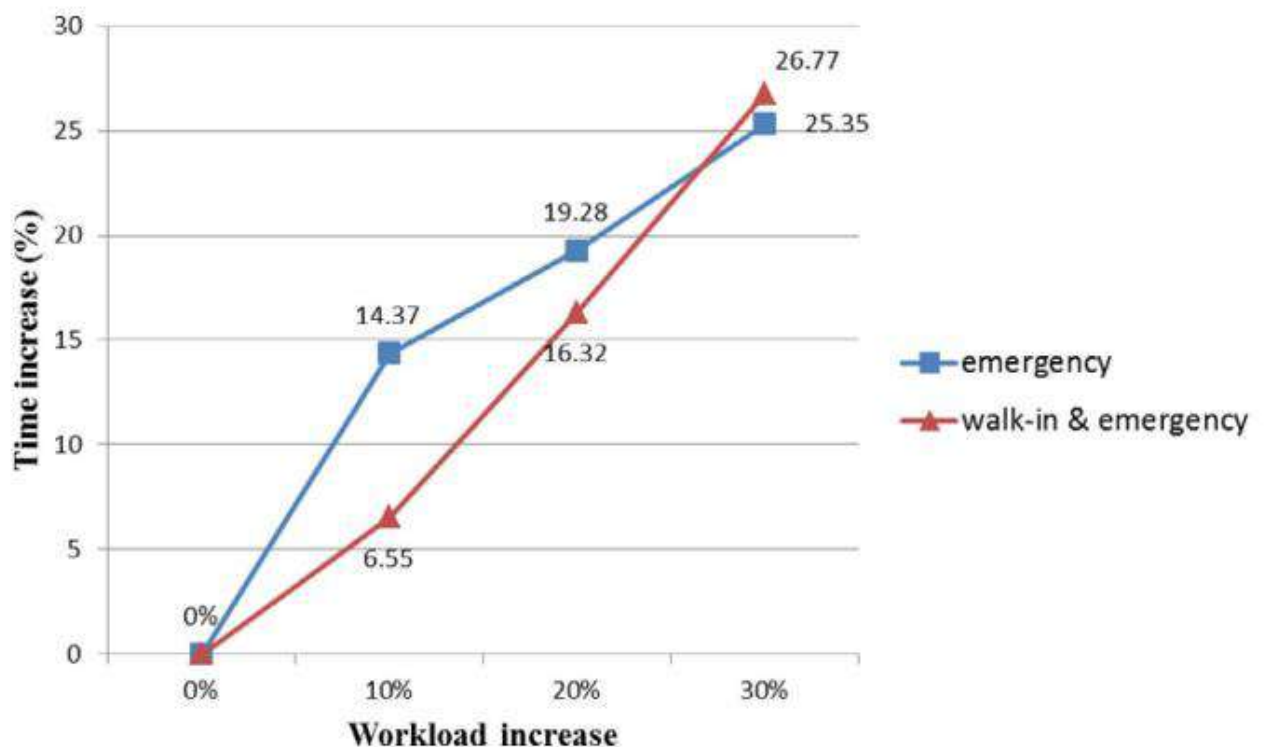


Рисунок 3.4 – Порівняльна масштабованість при збільшенні навантажень

Цей графік показує вплив на час виконання від додавання 10–30% додаткового навантаження від додаткових пацієнтів екстреної допомоги та пацієнтів, які прибули. Як видно, розподілений час виконання моделювання приблизно відображає додаткову роботу моделювання через додаткові події, які необхідно обробити в кожній із симуляцій через загальне збільшення пацієнтів (тобто кожному A&E потрібно більше часу для обробки свого робочого навантаження).

Рисунок 3.5 показує прискорення розподіленого моделювання порівняно з одним моделюванням протягом одного тижня. Тенденція відображає збільшення часу виконання для однієї симуляції порівняно з відносно постійним часом виконання розподіленої симуляції.

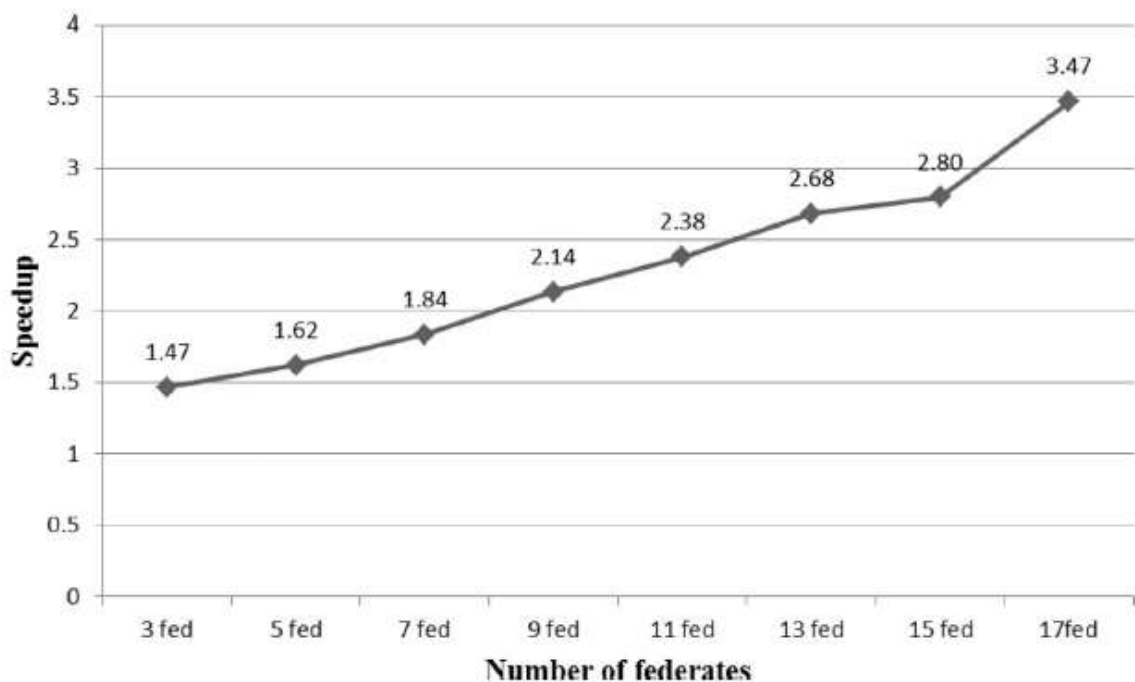


Рисунок 3.5 – Прискорення розподіленого моделювання порівняно з одиничним моделюванням (1 тиждень)

Таким чином, показана працездатність розподіленого моделювання з різними принципами організацій федеративів з значним прискоренням.

ВИСНОВКИ

Запропонована методологічна структура OR/MS DS надала чіткі вказівки для побудови гібридної моделі DS EMS. Організація необхідних дій у рамках звичайних практик OR/MS, а також пояснення їх за допомогою термінології, що не стосується програмної інженерії, значно допомогли в усьому управлінні проектом. Проте ми визнаємо складність розробки DS порівняно з моделями OR/MS, особливо коли моделісти використовують для створення своїх моделей за допомогою COTS, які не вимагають знань програмування. Для розробки DS, безумовно, існує потреба в програмуванні проміжного програмного забезпечення, а оволодіння технічними деталями HLA представляє собою круту криву навчання. Однак методологічна основа OR/MS DS спростила процес і допомогла зрозуміти, що відмінності між моделюванням OR/MS і DS не є драматичними. Важливо те, що це внесло значний внесок у концептуалізацію моделі DS.

Можна стверджувати, що перехід від моделювання OR/MS до DS є найскладнішим завданням. У нашій реалізації розподілена концептуалізація була найбільш вимогливою та тривалою діяльністю. У цьому прикладі виявлення підмоделей було досить очевидним, оскільки кожна підмодель представляла окрему організацію. Але є системи, в яких це складніше, особливо коли вся система знаходиться в одній організації і навіть в одному відділі. Однак виявлення взаємодії та визначення IRM було досить складним завданням.

Вищезазначені рішення є критично важливими для реалізації DS, і їх слід уважно розглянути при розподіленій концептуалізації. Зазвичай рішення таких проблем у симуляції OR/MS реалізуються пакетами моделювання COTS. Для OR/MS моделістів це зміна в тому, як вони розробляють свою модель. Крім того, у цьому прикладі ми розробили DS, використовуючи лише мінімально необхідні функції HLA. HLA – це складний стандарт, який

повністю охоплює всі (або майже всі) можливі служби для DS. Розробникам моделей OR/MS зазвичай потрібно реалізувати лише частину з них. Можливо, представлення всього процесу за допомогою знайомого та спрощеного методу для керівництва розробкою проекту є значним кроком у об'єднанні підходів OR/MS та DS.

Метод, який поєднує методи OR/MS і DS, відсутній в літературі з моделювання. Наша методологічна основа OR/MS DS є додатковим кроком до створення теоретичних інструментів для DS для додатків OR/MS. Підкріплений практичним прикладом, він містить основу для проектів DS OR/MS, які поєднують практики DS, описані в DSEEP, і практики OR/MS, представлені в популярному методі OR/MS Бенксом. Можна стверджувати, що добре розроблені теоретичні основи для DS в контексті OR/MS могли б об'єднати два підходи.

У цьому прикладі ми продемонстрували, що DS можна розробити відносно простим способом для додатків. Тим не менш, застосування запропонованої методологічної основи наразі обмежено сферою охорони здоров'я. Він ще не перевірений в інших сферах OR/MS, таких як ланцюг постачання, виробництво тощо, кожна з яких представляє різні проблеми. Крім того, ця реалізація передбачала створення федеративів заново. Дуже корисним доповненням було б докладне завдання щодо того, як модифікувати існуючі симуляції OR/MS, щоб стати частиною об'єднаної моделі DS.

Визнається, що існують концептуальні та технічні проблеми при розробці DS. Особливо для розробників моделей OR/MS, які зазвичай використовують пакети моделювання для створення симуляцій, і вони рідко є інженерами програмного забезпечення. Практики та дослідники OR/MS не розробляють системи моделювання з використанням мов програмування низького рівня, тому налаштування проміжного програмного забезпечення не є прийнятним завданням. Зокрема, незважаючи на те, що звичні практики здаються достатніми, навіть з їх обмеженнями, розробники моделей OR/MS не будуть докладати зусиль для впровадження практик DS. Основна причина в

тому, що DS виглядає технічно складним. У той же час, переваги DS в додатках OR/MS ще не очевидні.

Тим не менш, програми промислового моделювання можуть багато виграти від використання DS. Нижче наведено деякі з переваг:

- моделі для великомасштабних складних систем можна розбити на менші моделі підсистем, які взаємодіють у об'єднаному середовищі, а не розробляти як дуже великі симуляції OR/MS;
- моделі можуть залишатися локальними та працювати як незалежні симуляції;
- послідовні та актуальні дані; дані можна оновлювати для окремих організацій в одному місці;
- дані залишаються конфіденційними; ділитися лише необхідною інформацією для взаємодії з іншими федератами;
- моделі підсистем реалізуються з використанням найкращої методики моделювання;
- підключення моделей автоматизоване та синхронізоване та підтримує моделювання з однією технікою та гібридне моделювання;
- міжплатформне підключення, коли моделі компонентів можна розробляти за допомогою різного програмного забезпечення для моделювання;
- повторне використання моделей компонентів для складання різних схем;
- ефективніше експериментування шляхом розподілу обчислювального навантаження на розподілену комп'ютерну інфраструктуру.

Крім того, у симуляції OR/MS великомасштабної системи, яка складається з багатьох підсистем, ці підсистеми тісно пов'язані між собою. Однак організації в мережевих операціях у реальному світі співпрацюють у слабо зв'язаний спосіб. Це можна спостерігати в областях всередині організації або за її межами, де підрозділи можуть обмінюватися вибраною інформацією, спілкуватися і розуміти один одного, але в той же час діяти

незалежно. Симуляція, яка здатна відобразити цей рівень реальності та має розподілений контроль над усією системою, але підтримує незалежність окремих частин, таких як DS, має перевагу і, отже, є природним вибором для таких систем.

Можливо, спільнота OR/MS може отримати користь від DS. Як згадувалося раніше, можливість багаторазового використання моделі, сховище складених моделей, які можна використовувати для створення різних великомасштабних розподілених симуляцій, може призвести до скорочення часу розробки моделі та більш ефективного використання ресурсів. Наприклад, завдяки скороченню часу виконання можна проводити більше експериментів і аналізу результатів. Крім того, з нинішнім стрімким розвитком у сфері великих даних і хмарних обчислень компоновані та сумісні симуляції можна пов'язати з іншими службами, такими як моделі даних, компоненти оптимізації та інструменти візуалізації та аналітики. Останнім часом зусилля були спрямовані на розробку хмарних платформ для пакетів моделювання OR/MS.

У цій роботі ми представили методологічну основу, яку можна використовувати як керівництво для розробки DS з точки зору практики OR/MS. Цю роботу можна буде продовжено шляхом застосування методологічної основи до інших регіональних EMS різного розміру та інших сфер OR/MS, таких як ланцюг постачання та виробництво.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. L. Aboueljinane, Z. Jemai, E. Sahin, Скорочення часу реагування швидкої допомоги за допомогою моделювання: випадок екстреної медичної служби департаменту Валь- Де-Марн, у: Матеріали 44-ї зимової конференції з моделювання (WSC12), Берлін, DE , 2012, стор. 943–954, doi:10.1109/WSC.2012.6465018.
2. S. Brailsford, Modeling human behavior –an (ID)Entity crisis? in: Proceedings of the 46th Winter Simulation Conference (WSC14), Savannah, GA, 2014, pp. 1539–1548, doi: 10.1109/WSC.2014.7020 0 06 .
3. Z. Tu , G. Zacharewicz , D. Chen , Building a High-level architecture federated interoperable framework from legacy information systems, Int. J. Computer Integr. Manuf. 27 (4) (2014) 313–332 .
4. L. Rabelo , S. Sala-Diakanda , J. Pastrana , M. Marin , S. Bhide , O. Joledo , J. Bardina , Simulation modeling of space missions using the high level architec- ture, Model. Simul. Eng. 2013 (2013) 12 967483 .
5. NASA. NASA Technology Roadmaps: TA 11: Modeling, Simulation, Information Technology, and Processing. Technology Area 11, NASA, Draft, May 2015 online: http://www.nasa.gov/sites/default/files/atoms/files/2015_nasa_technology_roadmaps_ta_11_modeling_simulation.pdf .
6. IEEE , Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules, 2010 IEEE Standard 1516-2010.
7. J.E. Hannay, H.P. Langtangen, C. MacLeod, D. Pfahl, J. Singer, G. Wilson, How do scientists develop and use scientific software? in: Proceedings of the 2nd International Workshop on Software Engineering for Computational Science and Engineering (ICSE 20 09), Vancouver, CA, 20 09, pp. 1–8, doi: 10.1109/SECSE.2009.5069155
8. IEEE , Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP), 2010 IEEE Standard 1730-2010.

9. T. Schulze, S. Straßburger, U. Klein: Migration of HLA into Civil Domains: Solutions and Prototypes for Transportation Applications. In: SIMULATION, Vol. 73, No. 5, pp. 296-303, November 1999.

10. S. Taylor, B. Gan, S. Straßburger, A. Verbraeck: HLA-CSPIF Panel on Commercial Off-the-Shelf Distributed Simulation. In: Proceedings of the 2003 Winter Simulation Conference, eds. S. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice, pp. 881-887. December 7-10, 2003. New Orleans, USA.

11. M. Rabe, F.-W. Jaekel: Non Military use of HLA within Distributed Manufacturing Scenarios. In: Proceedings Simulation und Visualisierung '01, (Eds.) T. Schulze, V. Hinz, S. Schlechtweg. Magdeburg, 22.03-23.03.

12. S. Strassburger, T. Schulze, R. Fujimoto, Future trends in distributed simulation and distributed virtual environments: results of a peer study, in: Proceedings of the 40th Winter Simulation Conference (WSC08), Miami, FL, 2008, pp. 777–785, doi: 10.1109/WSC.2008.4736140 .

13. C.A. Boer, A. de Bruin, A. Verbraeck, Distributed simulation in industry –a survey Part 3 –The HLA standard in industry, in: Proceedings of the 40th Winter Simulation Conference (WSC08), Miami, FL, 2008, pp. 1094–1102, doi: 10.1109/WSC.2008.4736178 .

14. R.M. Fujimoto , Research challenges in parallel and distributed simulation, ACM Trans. Model. Comput. Simul. 24 (4) (2016).

15. J. Eatock , M. Clarke , C. Picton , T. Young , Meeting the four-hour deadline in an A&E department, J. Health Organ. Manag. 25 (6) (2011) 606–624 .

16. O.M. Ashour , G.E. Okudan-Kremer , A simulation analysis of the impact of FAHP-MAUT triage algorithm on the emergency department performance measures, Expert Syst. Appl. 40 (1) (2013) 177–187.

17. A. Hahn , V. Gollücke , C. Buschmann , S. Schweigert , Virtual test bed for maritime safety assessment, Sci. J. Maritime Univ. Szczecin 44 (116) (2015) 116–122.

18. U. Klein, T. Schulze, S. Strassburger, Traffic simulation based on the high level architecture, in: Proceedings of the 30th Winter Simulation Conference

- (WSC98), Washington, DC, 1998, pp. 1095–1103, doi: 10.1109/WSC.1998.745965.
19. L. Bernard , A. Wytzisk , U. Streit , Dynamic interoperable geoprocessing and geosimulation –scenarios, frameworks, and prototypes, in: Proceedings of the 4th AGILE Conference on Geographic Information Science, Brno, CZ, 2001.
20. H. Taghaddos, S.M. AbouRizk, Y. Mohamed, I. Ourdev, Distributed agent-based simulation of construction projects with HLA, in: Proceedings of the 40th Winter Simulation Conference (WSC08), Miami, FL, 2008, pp. 2413–2420, doi: 10.1109/WSC.2008.4736349.
21. G. Bruzzone , R. Mosca , R. Revetria , E. Bocca , E. Briano , Agent directed HLA simulation for complex supply chain modeling, *Simulation* 81 (9) (2005) 647–655.
22. S. Jain, F. Riddick, A. Craens, D. Kibira, Distributed simulation for interoperability testing along the supply chain, in: Proceedings of the 39th Winter Simulation Conference (WSC07), Washington, DC, 2007, pp. 1044–1052, doi: 10.1109/WSC.2007.4419703.
23. K. Katsaliaki , N. Mustafee , S.J.E. Taylor , S. Brailsford , Comparing conventional and distributed approaches to simulation in a complex supply-chain health system, *J. Oper. Res. Soc.* 60 (1) (2009) 43–51.
24. A.C. Medina, L.G. Nardin, N.N. Pereira, R.C. Botter, J.S. Sichman, A distributed simulation model of the maritime logistics in an iron ore supply chain management, in: Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH-2013), Reykjavík, IS, 2013, pp. 453–460, doi: 10.5220/0 0 04488504530460.
25. Q. Long. An agent-based distributed computational experiment framework for virtual supply chain network development, *Expert Syst. Appl.* 41 (9) (2014) 4094–4112.
26. Z. Tu , G. Zacharewicz , D.Chen , Afederated approach to develop enterprise interoperability, *J. Intell. Manuf.* 27 (1) (2016) 11–31.
27. S.J.E. Taylor , S.J. Turner , S. Strassburger , N. Mustafee , Bridging the gap: a standards-based approach to OR/MS distributed simulation, *ACM Trans.*

Model. Comput. Simul. 22 (4) (2012) 18.

28. S. Strassburger, T. Schulze, M. Lemessi, Applying CSPI reference models for factory planning, in: Proceedings of the 39th Winter Simulation Conference (WSC07), Washington, DC, 2007, pp. 603–609, doi: 10.1109/WSC.2007.4419653.

29. M. Raab, S. Masik, T. Schulze, Support system for distributed HLA simulations in industrial applications, in: Proceedings of the 2011 IEEE Workshop on Principles of Advanced and Distributed Simulation (PADS'11), Nice, FR, 2011, pp. 1–7, doi: 10.1109/PADS.2011.5936773.

30. G. Pedrielli , M. Sacco , W. Terkaj , T. Tolio , An HLA-based distributed simulation for networked manufacturing systems analysis, J. Simul. 6 (4) (2012) 237–252.

31. SISO, Standard for Commercial-off-the-shelf Simulation Package interoperability reference Models, 2010 SISO-STD-006-2010.

32. M. Raab, S. Masik, T. Schulze, Support system for distributed HLA simulations in industrial applications, in: Proceedings of the 2011 IEEE Workshop on Principles of Advanced and Distributed Simulation (PADS'11), Nice, FR, 2011, pp. 1–7, doi: 10.1109/PADS.2011.5936773.

33. G. Pedrielli , M. Sacco , W. Terkaj , T. Tolio , An HLA-based distributed simulation for networked manufacturing systems analysis, J. Simul. 6 (4) (2012) 237–252.

34. C.M. Macal and M.J. North. Tutorial on agent-based modelling and simulation. J. Simul. , 4(3):151–162.

35. S Robinson , Simulation –The Practice of Model Development and Use, John Wiley & Sons Ltd, Chichester, UK, 2004.

36. N. Mustafee , M. Sahnoun , A. Smart , P. Godsiff, D. Baudry , A. Louis , Investigating execution strategies for hybrid models developed using multiple M&S methodologies, in: Proceedings of the 2015 Spring Simulation MultiConference (ANSS 2015), Alexandria, VA, 2015.

37. N. Mustafee, M. Sahnoun, A. Smart, P. Godsiff, An application of distributed simulation for hybrid modeling of offshore wind farms, in: Proceedings

of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS'15), London, UK, 2015, pp. 171–172, doi: 10.1145/2769458.2769492.

38. C. Laesche, V. Golluecke, A. Hahn, Using an HLA simulation environment for safety concept verification of offshore operations, in: Proceedings of the 27th European Conference on Modelling and Simulation (ECMS 2013), Aalesund, NO, 2013, pp. 156–162, doi: 10.7148/2013-0156.

39. J. Banks , J.S. Carson , B.L. Nelson , D.M. Nicol , Discrete-Event System Simulation, third ed, Prentice Hall Inc., USA, 2020.

40. A.M. Law , W.D. Kelton , Simulation Modeling and Analysis, third ed., McGraw-Hill Book Co, Singapore, 2021, ISBN: 0-07-116537-1.

41. S Robinson , Simulation –The Practice of Model Development and Use, John Wiley & Sons Ltd, Chichester, UK, 2014.

42. IEEE, Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP), 2003 IEEE Standard 1516.3-2003.

43. IEEE , Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP), 2010 IEEE Standard 1730-2010.

44. IEEE 1278.1-2012 - Standard for Distributed Interactive Simulation – Application protocols.

45. R. Siegfried , J. Luethi , G. Herrmann , M. Hahn , A comparison of DSEEP with the German approach VEVA, in: Proceedings of the 2021 Spring Simulation Interoperability Workshop (SISO), Boston, US, 2021.

46. S Robinson, Simulation –The Practice of Model Development and Use, John Wiley & Sons Ltd, Chichester, UK, 2019.

47. B.S.S. Onggo , J. Hill , Data identification and data collection methods in simulation: a case study at the ORH Ltd, J. Simul. 8 (3) (2014) 195–205.

48. L. Aboueljineane, Z Jemai, E. Sahin, Reducing ambulance response time using simulation: the case of Val-De-Marne department emergency medical service, in: Proceedings of the 44th Winter Simulation Conference (WSC12), Berlin, DE,

2012, pp. 943–954, doi: 10.1109/WSC.2012.6465018.

49. S. Ibri , M. Nourelfath , H. Drias , A multi-agent approach for integrated emergency vehicle dispatching and covering problem, *Eng. Appl. Artif. Intell.* 25 (3) (2012) 554–565.

50. T. Lee, H. Jang, S.H. Cho, J.G. Turner, A simulation-based iterative method for a Trauma center –air ambulance location problem, in: *Proceedings of the 44th Winter Simulation Conference (WSC12)*, Berlin, DE, 2012, pp. 955–966, doi: 10.1109/WSC.2012.6465042.

51. M. van Buuren, K. Aardal, R. van der Mei, H. Post, Evaluating dynamic dispatch strategies for emergency medical services: TIFAR simulation tool, in: *Proceedings of the 44th Winter Simulation Conference*, Berlin, DE, 2012, pp. 509–519, doi: 10.1109/WSC.2012.6465214.

52. L. Wang, An agent-based simulation for workflow in emergency department, in: *Proceedings of the 2009 IEEE Systems and Information Engineering Design Symposium (SIEDS'09)*, Charlottesville, VA, 2009, pp. 19–23, doi: 10.1109/SIEDS.2009.5166148 .