Національна академія наук України
Інститут кібернетики імені В. М. Глушкова

Кваліфікаційна наукова
праця на правах рукопису

ГОРБАТЮК СЕРГІЙ ОЛЕКСАНДРОВИЧ

УДК 004.05

ДИСЕРТАЦІЯ

МЕТОДИ МОДЕЛЬНОГО РОЗРОБЛЕННЯ ТА АНАЛІЗУВАННЯ КІБЕРБЕЗПЕКИ ДЛЯ ІНФОРМАЦІЙНИХ СИСТЕМ ЛОГІСТИКИ

123 Комп'ютерна інженерія
12 Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____ С.О. Горбатюк

Науковий керівник – Летичевський О.О., доктор фізико-математичних наук, завідувач відділу теорії цифрових автоматів Інституту кібернетики імені В.М. Глушкова Національної Академії Наук України

Київ – 2023

АНОТАЦІЯ

Горбатюк С. О. Методи модельного розроблення та аналізування кібербезпеки для інформаційних систем логістики. - Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 12.123 Комп'ютерна інженерія. – Інститут кібернетики імені В.М. Глушкова, Київ, 2023.

Загальна проблема дослідження полягає у перевірці властивостей безпеки розподілених систем, що використовуються для вирішення відповідних задач у системах міжнародної логістики, за допомогою перевірки моделей (model checking) та алгебраїчного підходу з інсерційним моделюванням. Іншою актуальною проблемою є задача виявлення та протидії випадкам шахрайства, особлива увага приділяється даній проблемі ще на стадії розробки високонадійних систем за допомогою модельного способу розробки логістичних систем.

Метою роботи є вирішення науково-прикладної проблеми створення методології для технології модельного способу розробки для високонадійних розподілених систем, зокрема для систем логістики.

Для досягнення вказаної мети необхідно розв'язати наступні взаємопов'язані задачі:

– провести аналіз існуючих моделей, методів і технологічних засобів, які дозволяють створити інструментарій та методологію для використання розподілених систем, зокрема технології блокчейн в логістичних системах;

– провести аналіз властивостей безпеки та можливих атак в логістичних системах

– визначити теоретичні основи базових понять інсерційного моделювання для логістичних агентів в певному діючому середовищі;

– запропонувати архітектурно-структурні рішення, алгоритми та програмно-апаратні засоби формування і відображення взаємодії логістичних

агентів та безперервного контролю за їх розміщенням та поведінками з метою дотримання безпеки функціонування;

– запропонувати рішення, методи кібербезпеки та алгоритми верифікації для перевірки властивостей цілісних логістичних систем, таких як безпека та життєдіяльність;

- визначити використання алгебраїчного та інсерційного підходу на стадіях модельного способу розробки для процедур верифікації, тестування;

- розглянути методи безпеки в логістичних системах, захисту від атак зловмисника та шахрайства та методи аналізу вразливостей;

- розглянути методи аналізу безпеки в системах на основі блокчейн платформ на основі інсерційного підходу;

- побудувати приклад модельної розробки логістичної системи.

Перший розділ дисертації «*Сучасні системи логістики та модельний метод розробки*» присвячено виявленню та вивченню глобальних проблем, які накопичились в сучасній системі міжнародної логістики, та які можуть бути вирішені за допомогою застосування технології блокчейн та методів модельної розробки систем. Проаналізовано приклади на напрямки прогресивних шляхів використання розподілених систем та смарт-контрактів на їх основі в міжнародній логістиці передовими логістичними операторами, науковими та державними установами. Розглянуто методологію проектування логістичних систем з врахуванням необхідності дотримання властивостей безпеки та надійності протидії шахрайству та атакам зловмисників. При розробці високонадійних систем, критичних до безпеки, є доцільним використання модельного способу розробки, тому проаналізовано процедуру валідації та верифікації артефактів на всіх етапах розробки таких систем.

У другому розділі «*Алгебраїчний підхід у формалізації, верифікації та модельному тестуванні в системах міжнародної та місцевої обслуговуючої логістики*» розкрито основи використання алгебраїчного підходу та інсерційного моделювання при формальній верифікації та формалізації за допомогою тестування на основі моделей. Наведено приклад застосування на практиці алгебри

поведінок на прикладі діючої закритої логістичної системи фермерського господарства, наведено приклади запису моделей поведінок агентів. Розглянуто основні властивості безпеки, що перевіряються в логістичній системі фермерського господарства. Приклад застосування у фермерському господарстві розширено до більш широкого застосування у відкритій системі міжнародної логістики.

У третьому розділі «*Перевірка властивостей кібербезпеки в проектуванні логістичних систем на блокчейн платформах*» проаналізовано проблеми безпеки в проектуванні логістичної блокчейн платформи. Описано інсерційну семантику блокчейн-системи та в розумних контрактах (Smart Contracts,) змодельовано дії зловмисників у цих системах, формалізовано атаки DAO для розумних контрактів. Запропоновано принципи дій для запобігання атакам при проектуванні блокчейн систем.

Четвертий розділ «*Використання модельного способу в розробці індустріальних логістичних систем*» описує два приклади використання модельного методу розробки: приклад побудови логістичної системи для фермерського господарства, яка використовує моделювання на рівні моделей вимог та другий - приклад формалізації вимог та подальше розроблення для логістичної системи, що визначає та супроводжує функціонування ланцюжка постачання. Розглянуто схему документообігу при мультимодальному міжнародному перевезенні, запропоновано використання модельного способу розробки контракту для мономодального перевезення на блокчейн платформі з метою мінімізації ризиків для учасників та запобіганню дій шахраїв.

Практичне значення отриманих результатів полягає у доведені наукових результатів до практичного використання, які підтверджені актами впровадження ФГ «Надія», ТОВ «Смарт Трейдинг», ПрАТ «Ксібекс», приватного підприємства ЛітСофт.

Автор дисертації є співрозробником спеціальної системи логістики для фермерського господарства, розробленої та впровадженої в фермерському господарстві «Надія», що знаходиться в с. Марійка, Жашківського району, Черкаської області, на основі досліджень в дисертації. Описані в дисертаційному

досліджені підходи до побудови логістичних систем є універсальними, та можуть використовуватись, за умови адаптації, до логістичних систем будь-якої складності та масштабу, як для закритих, так і для відкритих систем національного та міжнародного рівня.

Одержані результати також використовуються з навчальною метою в Національному університеті біоресурсів і природокористування України під час викладання дисциплін для ІТ-спеціальностей.

Ключові слова: *верифікація, валідація, математичне моделювання, алгебраїчний підхід, інсерційне моделювання, розподілені системи, міжнародна логістика, логістичні системи.*

ANNOTATION

S.O. Horbatiuk. Cybersecurity problems and model method of distributed systems development in international logistics. - Qualifying scientific work on the rights of the manuscript.

The dissertation on competition of a scientific degree of the doctor of philosophy on a specialty 12.123 Computer Engineering. – V.M. Glushkov Institute of Cybernetics of NAS of Ukraine, Kyiv, 2021.

The general problem of the research is to check the security properties of distributed systems used to solve problems in international logistics systems, using model checking and algebraic approach with insertion modeling. Another urgent problem is the task of detecting and counteracting cases of fraud, special attention is paid to this problem at the stage of development of highly reliable systems using a model method of developing logistics systems.

The goal of the work is to solve the scientific and applied problem of creating a technology model of development for highly reliable distributed systems, in particular for logistics systems.

To achieve this goal it is necessary to solve the following interrelated problems:

– to analyze the existing models, methods and technological means that allow to create tools and methodology for the use of distributed systems, in particular blockchain technology in logistics systems;

– to analyze security properties and possible attacks in logistics systems;

– to determine the theoretical foundations of the basic concepts of insertion modeling for logistics agents in a particular operating environment;

– to offer architectural and structural solutions, algorithms and software and hardware for the formation and display of the interaction of logistics agents and continuous control over their location and behavior in order to ensure the safety of operation;

– to propose solutions, cybersecurity methods and verification algorithms to verify the properties of integrated logistics systems, such as security and life;

- to determine the use of algebraic and insertional approach at the stages of the model method of development for verification, testing procedures;
- to consider methods of security in logistics systems, protection against attackers and fraud and methods of vulnerability analysis;
- to consider methods of security analysis in systems based on blockchain platforms based on the insertion approach;
- to build an example of model development of a logistics system.

The first section of the dissertation *"Modern logistics systems and model development method"* is devoted to identifying and studying global problems that have accumulated in the modern system of international logistics, and which can be solved using blockchain technology and methods of model development. Examples on the directions of progressive ways of using distributed systems and smart contracts based on them in international logistics by advanced logistics operators, research and government agencies are analyzed. The methodology of designing logistics systems is considered, taking into account the need to comply with the properties of security and reliability to combat fraud and attacks by attackers. When developing highly reliable systems critical to security, it is advisable to use a model method of development, so the procedure for

validation and verification of artifacts at all stages of development of such systems is analyzed.

The second section *"Algebraic approach in formalization, verification and model testing in international and local service logistics systems"* reveals the basics of using the algebraic approach and insertion modeling in formal verification and formalization through model-based testing. An example of application in practice of algebra of behaviors on an example of the operating closed logistic system of a farm is given, examples of record of models of behavior of agents are resulted. The main safety properties tested in the logistics system of the farm are considered. The example of application in farming has been extended to a wider application in the open system of international logistics.

The third section *"Testing the properties of cybersecurity in the design of logistics systems on blockchain platforms"* analyzes security issues in the design of logistics blockchain platform. The insertion semantics of a blockchain system are described and in Smart Contracts, the actions of attackers in these systems are modeled, DAO attacks for smart contracts are formalized. The principles of actions for prevention of attacks at designing of blockchain systems are offered.

The fourth section *"Using the model method in the development of industrial logistics systems"* describes two examples of using the model development method: an example of building a logistics system for a farm that uses modeling at the level of requirements models and the second - an example of formalization requirements and further development for the logistics system. and accompanies the functioning of the supply chain. The scheme of document circulation in multimodal international transportation is considered, the use of a model method of contract development for monomodal transportation on a blockchain platform is proposed in order to minimize risks for participants and prevent fraud.

The practical significance of the obtained results lies in bringing the scientific results to practical use, which are confirmed by the acts of implementation of farming household "Nadiya", LLC "Smart Trading", PJSC "Xebec’s", private enterprise LitSoft.

The author of the dissertation is a co-developer of a special logistics system for the farm, developed and implemented in the farm "Nadiya" located in the village. Mariyka, Zhashkiv district, Cherkasy region, based on research in the dissertation. The approaches to the construction of logistics systems described in the dissertation research are universal and can be used, subject to adaptation, to logistics systems of any complexity and scale, both for closed and open systems of national and international level.

The obtained results are also used for educational purposes at the National University of Life and Environmental Sciences of Ukraine in teaching disciplines for IT specialties.

Keywords: *verification, validation, mathematical modeling, algebraic approach, insertion modeling, distributed systems, international logistics, logistics systems.*

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Основні наукові результати дисертації

1. О.О. Летичевський, С.О. Горбатюк. Децентралізовані системи в логістиці: огляд використання та проблеми безпеки // Проблеми програмування – 2020, №1. – С. 55-73. DOI: 10.15407/pp2020.01.055

2. О.О. Летичевський, С.О. Горбатюк, В.О. Горбатюк. Алгебраїчне моделювання в системах міжнародної та місцевої обслуговуючої логістики // Проблеми програмування – 2020, №4. – С. 88-97. DOI: 10.15407/pp2020.04.088

3. В.О. Горбатюк, С.О. Горбатюк. Methods of detection of HTTP attacks on a smart home using the algebraic methods // Проблеми програмування. 2022. № 3-4. Спеціальний випуск – С. 396-402. DOI: 10.15407/pp2022.03-04.396

4. В.О. Горбатюк, С.О. Горбатюк. Методи перевірки алгебраїчним співставленням спротиву HTTP-атакам на розумний будинок // Control systems and computers, 2022, № 4 – С. 13-23. DOI: 10.15407/csc.2022.04.013

5. С. Горбатюк. Блокчейн в логістиці та формальна верифікація властивостей безпеки // Ідеї академіка В.М. Глушкова і сучасні проблеми штучного інтелекту. 8-ма Всеукраїнська науково-практична конференція: “Глушковські читання” (29 жовтня 2019, Київ). – Київ, Видавничий дім Ліра-К, 2019. – С. 57–60.

6. Васюхін М., Касім А., Долинний В., Касім М., Шелестовський В., Горбатюк С. Метод створення класифікатора картографічної інформації для агрономічних автоматизованих систем // Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні ‘2017: збірник матеріалів V Міжнародної науково-практичної конференції (22 – 23 червня 2017 року, Київ). – К.: Компринт, 2017. – С. 47–49.

7. Васюхін М.І., Касім А.М., Долинний В.В., Касім М.М., Шелестовський В.Г., Горбатюк С.О. Геоінформаційна система для малих і середніх фермерських господарств // Безпека життєдіяльності на транспорті і виробництві – освіта, наука,

практика: тези доповідей IV Міжнародної науково-практичної конференції (14-16 вересня 2017 року, Херсон). – Херсон: ХДМА, 2017. – С. 324–330.

8. Васюхін М.І., Горбатюк С.О., Касім М.М., Шелестовський В.Г. комп'ютерні системи. Навчальний посібник. – К.: Компринт, 2017. – 270 сторінок. ISBN 978-966-929-552-1

9. Гусєв Б.С., Горбатюк С.О., Савицька Я.А., Смолій В.В., Шелестовський В.Г. Інформаційна технологія системи управління фермерським господарством. Монографія. – НУБіП України, 2018.- 220 сторінок.

10. Oleksandr Letychevskyi, Serhii Horbatiuk, Viktor Horbatiuk. Algebraic modelling of logistical systems equipped by wireless monitoring devices // The 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (17-18 September 2020, Dortmund, Germany). DOI: 10.1109/IDAACS-SWS50031.2020.9297092

11. Oleksandr Letychevskyi, Volodymyr Peschanenko, Maksym Poltoratskyi, Serhii Horbatiuk, Horbatiuk Viktor and Yuliia Tarasich. One Approach to Formal Verification of Distributed Ledger Technologies on the Blockchain Technologies Example // Conference on Mathematical Foundations of Informatics: Proceedings MFOI-2020 (12–16 January 2021) – Taras Shevchenko National University of Kyiv - pages 227-242.

12. Oleksandr Letychevskyi, Volodymyr Peschanenko, Sergiy Horbatiuk. Consensus Protocol Security Analysis Using an Algebraic Virtual Machine // IntelITSIS'2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security (23–25 March, 2022) – Khmelnytskyi, Ukraine - pages 484-493.

ЗМІСТ

ВСТУП 14

РОЗДІЛ 1. СУЧАСНІ СИСТЕМИ ЛОГІСТИКИ ТА МОДЕЛЬНИЙ МЕТОД РОЗРОБЛЕННЯ 22

1.1. Поняття логістичної системи, основні терміни та агенти в системах логістики 22

1.2. Блокчейн, як платформа для розподілених логістичних систем. 25

1.3. Розподілені системи, контрольовані приладами геолокації 30

1.4. Огляд використання передових блокчейн-рішень в логістиці 31

1.5. Смарт-контракти на базі блокчейн та мова Solidity 37

1.6. Властивості безпеки логістичних систем. 40

1.7. Формальна верифікація в проектуванні логістичних систем 45

1.8. Проектування логістичних систем та модельний метод обробки 46

1.9. Модельний спосіб розробки надійних систем з підвищеними вимогами до безпеки 47

1.10. Етапи процесу модельної розробки систем з підвищеними вимогами до безпеки. 48

1.11. Методи верифікації на етапі визначення вимог та дизайну проекту. 49

1.12. Метод модельного тестування 50

1.13. Інтеграція процедури верифікації властивостей безпеки у стандартний процес модельної розробки програмного забезпечення. 53

Висновки до Розділу 1 55

Список використаних джерел до Розділу 1. 58

РОЗДІЛ 2. АЛГЕБРАЇЧНИЙ ПІДХІД У ФОРМАЛІЗАЦІЇ, ВЕРИФІКАЦІЇ ТА МОДЕЛЬНОМУ ТЕСТУВАННІ В СИСТЕМАХ МІЖНАРОДНОЇ ТА МІСЦЕВОЇ ОБСЛУГОВУЮЧОЇ ЛОГІСТИКИ 65

2.1. Алгебра поведінок, розвиток та сфери застосування 66

2.2. Приклад формалізації системи взаємодії агентів закритої логістичної системи 67

2.3. Формальна верифікація моделей 78

2.4. Модельне тестування. 81

2.5. Верифікація і формалізації системи міжнародної логістики. 82

Висновки до Розділу 2. 85

Список використаних джерел до Розділу 2. 86

РОЗДІЛ 3. ПЕРЕВІРКА ВЛАСТИВОСТЕЙ КІБЕРБЕЗПЕКИ В ПРОЕКТУВАННІ ЛОГІСТИЧНИХ СИСТЕМ НА БЛОКЧЕЙН ПЛАТФОРМАХ. 87

3.1. Проблеми безпеки в проектуванні логістичної блокчейн платформи 87

3.2. Інсерційна семантика блокчейн-системи 89

3.3. Моделювання зловмисника в блокчейн. 94

3.4. Інсерційна семантика розумних контрактів (Smart Contracts) 96

3.5. Формалізація атаки DAO для розумних контрактів. 98

3.6. Запобігання атакам при проектуванні блокчейн системи 99

3.7. Запобігання атакам при функціонуванні блокчейн платформи 102

Висновки до Розділу 3. 103

Список використаних джерел до Розділу 3. 105

РОЗДІЛ 4. ВИКОРИСТАННЯ МОДЕЛЬНОГО СПОСОБУ В РОЗРОБЦІ ІНДУСТРІАЛЬНИХ ЛОГІСТИЧНИХ СИСТЕМ 106

4.1. Логістична система на базі фермерського господарства. 106

4.1.1. Концепція системи управління фермерським господарством. Призначення системи..... 107

4.1.2. Методи реалізації функціональних компонентів системи 109

4.1.3. Методи реалізації інформаційно-комунікаційного середовища та ядра системи..... 113

4.1.4. Методи реалізації програмного забезпечення геоінформаційної підсистеми… 115

4.1.5. Методи реалізації інтерфейсу навігаційної системи..... 119

4.2. Система моделювання та створення системи управління міжнародної відкритої логістичної системи 123

4.3 Модельний спосіб розробки контракту для мономодального перевезення на блокчейн платформі. 129

Висновки до Розділу 4. 133

Список використаних джерел до Розділу 4. 135

ВИСНОВКИ..... 136

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 139

ДОДАТКИ..... 148

ВСТУП

Актуальність теми. За сучасних умов глобалізації міжнародна торгівля розширює своє значення та масштаби. Міжнародна логістика та глобальні всеохоплюючі ланцюжки постачання світового масштабу стають звичайним явищем. При таких шалено зростаючих масштабах міжнародної торгівлі створення високонадійних міжнародних логістичних систем є однією з важливих задач сучасної науки. В сучасних умовах цілісний ланцюжок постачання вимагає постійного та безперервного контролю на всіх його етапах – від виробника і до кінцевого споживача, а також контроль на етапах постачання сировини для багатокомпонентного виробництва. Вимога надійності обміну інформацією та міжнародними стандартизованими документами та швидкість їх обробки має вкрай важливе значення, оскільки об’єм потоку процесу обробки можливих сценаріїв росте експоненційно при вирішуванні задач аналізу дотримування умов або властивостей ланцюжка перевезення. Тому перехід на використання розподілених систем стає необхідністю. Широко застосовується технологія блокчейн – розподілена база даних з записами або публічний реєстр всіх транзакцій або цифрових подій, що були виконані та поширені серед всіх учасників. Дана технологія дозволяє вирішити як стандартні існуючі проблеми міжнародних перевезень, так і новітні проблеми, пов’язані з електронним документообігом, зберіганням та опрацюванням величезних масивів інформації та створення баз даних, так само, як і обслуговування серверів, що оперують такими базами даних та масивами інформації. Важливою проблемою логістичних систем також є проблема відслідковування розміщення агентів в реальному часі за допомогою технології GPS, взаємодія усіх цих агентів та безперервний запис маршрутів. На сучасному етапі переваги використання технології блокчейн є очевидними, тому росте число наукових розробок в даному напрямку, а сама технологія все ширше застосовується на практиці.

Однак з поширенням технології зростає і проблема перевірки властивостей безпеки таких систем. З цією метою в сучасній методології існують такі напрями,

як перевірка моделей (model checking), що базується на використанні методів моделювання та статичної перевірки властивостей безпеки. При розгляданні децентралізованої моделі логістики складність перевірки зростає, тому моделювання є недостатнім і для верифікації моделі, створеної на основі блокчейн, використовують алгебраїчний підхід.

Основи для вирішення задач моделювання закладені ще Віктором Михайловичем Глушковим та розвинуті в роботах його учнів та послідовників. Також в Інституті кібернетики ім. В.М.Глушкова Національної Академії Наук України розроблено теорію інсерційного моделювання, об'єктом якої є взаємодія агентів та середовищ. Застосовуючи дані розробки, в роботі застосовано систему агентів, які діють в деяких середовищах, кожен агент при цьому має свій тип, атрибути та властивості. Поведінка агентів детермінується за допомогою алгебри поведінок з жорсткою регламентацією передумов та допустимих дій.

Система інсерційного моделювання раніше була використана в ряді проектів по верифікації та тестуванню, зокрема в проектах компанії Motorola. Досвід її використання показав, що даний підхід є ефективним для вичерпного аналізу та доведення властивостей програмних та апаратних систем, тому її використання доцільне в розробці розподілених систем, зокрема логістичних.

В системах такого рівня та складності, як міжнародні логістичні системи, та з такою великою кількістю агентів, які діють з різними цілями та інтересами, присутні деякі агенти, які цілеспрямовано завдають шкоди системам з метою отримання певних переваг ти вигод. Зокрема часто присутні випадки шахрайства – надання неправдивої чи спотвореної інформації заради отримання певної особистої вигоди, часто за рахунок чесних агентів, зокрема злочинні дії з ціллю присвоєння товару або грошових коштів. Часто зловмисники вчиняють кібер-атаки на системи з метою контролю за певними її функціями. Тому процес розробки високонадійних систем вимагає циклу розробки, який би створював системи відслідковування, виявлення та протидії випадкам неправомірних дій зловмисників – шахраїв та кібер-злочинців. Такі системи повинні зберігати функціональність та стійку протидію цим атакам. Для цього розроблено та використовується спеціальний

набір інструментів – модельний спосіб розробки логістичних систем, де на кожній стадії розробки створюються моделі згідно наперед визначених вимог, властивостей безпеки, сценаріїв, побажань замовника та інше. Згідно наведених вимог розробки перспективно виглядає комбінований метод – поєднання модельного способу розробки та використання алгебраїчного підходу та інсерційного моделювання. Дана технологія дасть змогу раціонально та ефективно використовувати ресурси при створенні міжнародних логістичних систем високого рівня надійності.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконана у відповідності з планами науково-дослідних робіт Інституту кібернетики імені В.М. Глушкова:

– НДР «Моделі, методи та засоби побудови програмно-технічних систем забезпечення захисту інформаційних ресурсів від несанкціонованого доступу», шифр М/232-2010, (№ держреєстрації 0107U007588) за етапом 3;

– НДР 1526 «Прикладна теорія і технологія побудови геоінформаційної системи реального часу» № держреєстрації 0116U001913, за етапом 3;

– ВФ.205.31 «Розробити теоретичні основи, методи та засоби інформаційних технологій підтримки трансдисциплінарних досліджень», № держреєстрації 0114U001056, за етапом 1 «Розробка теоретичних засад онтологічного опису предметної області стосовно Semantic Web»;

– НДР ВФК.205.28 «Розробити теоретичні засади, методи та інформаційні технології побудови комп'ютерних засобів та систем на основі інтегрованого використання методів обробки знань, редукційного паралелізму та реконфігурування»;

- ВФК.100.14 «Розробити теоретичні засади та методи модельного способу побудови та реінжинірингу систем, що критичні до безпеки, на основі символьного інсерційного моделювання, дедуктивних методів та методів штучного інтелекту»;
- ВФ.100.15 «Розробити теоретичні засади аналізу кіберфізичних систем на основі інсерційного моделювання»;

- В.П.100.16 «Розробити формальні методи виявлення вразливостей програмних систем».

Роль автора в цих роботах полягає в розробці моделей, методів та технологічних засобів побудови діючої системи внутрішньої обслуговуючої логістики та єдиної інтелектуальної бази даних та серверу на базі фермерського господарства «Надія», міжнародної логістики на базі ПрАТ «Ксібекс» та ТОВ «Смарт Трейдинг», створення інструментарію та методології використання розподілених баз даних (технології блокчейн) для виконання міжнародних контрактів з логістики. Крім того створено моделі логістичних систем для яких застосовано методи верифікації, кібербезпеки та тестування, що створені у відділі теорії цифрових автоматів Інституту кібернетики ім. В.М. Глушкова НАН України

Мета і завдання дослідження. Метою роботи є вирішення науково-прикладної проблеми створення технології модельного способу розробки для високонадійних розподілених систем, зокрема для систем логістики.

Для досягнення вказаної мети необхідно розв'язати наступні взаємопов'язані задачі:

– провести аналіз існуючих моделей, методів і технологічних засобів, які дозволяють створити інструментарій та методологію для використання розподілених систем, зокрема технології блокчейн в логістичних системах;

– провести аналіз властивостей безпеки та можливих атак в логістичних системах

– визначити теоретичні основи базових понять інсерційного моделювання для логістичних агентів в певному діючому середовищі;

– запропонувати архітектурно-структурні рішення, алгоритми та програмно-апаратні засоби формування і відображення взаємодії логістичних агентів та безперервного контролю за їх розміщенням та поведінками з метою дотримання безпеки функціонування;

– запропонувати рішення, методи кібербезпеки та алгоритми верифікації для перевірки властивостей цілісних логістичних систем, таких як безпека та життєдіяльність;

- визначити використання алгебраїчного та інсерційного підходу на стадіях модельного способу розробки для процедур верифікації, тестування;
- розглянути методи безпеки в логістичних системах, захисту від атак зловмисника та шахрайства та методи аналізу вразливостей;
- розглянути методи аналізу безпеки в системах на основі блокчейн платформ на основі інсерційного підходу;
- побудувати приклад модельної розробки логістичної системи.

Об'єкт дослідження – модельний спосіб розробки цілісних логістичних систем для контролю за діями агентів, інсерційне моделювання та забезпечення кібербезпеки.

Предмет дослідження – моделі, методи і технологічні засоби побудови логістичних систем в модельному способі розробки розподілених систем сучасної логістики на основі блокчейн платформ.

Методи досліджень. Для розв'язання поставлених задач був використаний комплекс методів: системний підхід Віктора Михайловича Глушкова, теорія програмування, методи дискретної математики і математичного моделювання, теорія інсерційного моделювання, теорія формальних мов та граматик, теорія алгебраїчного програмування, методи символьного моделювання та алгебраїчного співставлення.

Наукова новизна отриманих результатів. У дисертаційній роботі розв'язана актуальна науково-прикладна задача – ***створення технології модельної розробки стосовно до розподілених логістичних систем, побудованих на блокчейн платформах із застосуванням алгебраїчного підходу та інсерційного моделювання з метою дотримання властивостей безпеки та стійкості до кібератак.***

На відміну від попередніх підходів, що базуються на методах імітаційного моделювання та на ймовірнісному підході, алгебраїчний підхід працює з множинами сценаріїв та дає **доказове підтвердження** виконання властивостей безпеки та стійкості до атак.

При її вирішенні отримано такі основні наукові і практичні результати:

Вперше:

1. Запропоновано життєвий цикл модельної розробки високонадійних програмних систем логістики. Життєвий цикл включає залучення до кожного етапу розробки процедур верифікації та тестування на основі створення формальних моделей.
2. Визначено та налаштовано формальні методи для верифікації та модельного тестування логістичної системи, а саме – методи символьного моделювання та статичного доведення властивостей.
3. Розроблено формальні моделі логістичної обслуговуючої системи фермерського господарства в мові алгебри поведінок, проведена верифікація та генерація тестового набору.
4. Розглянуто інсерційну семантику блокчейн системи та методи виявлення в вразливостей, протидії атакам зловмисників у логістичній системі.
5. Приведено повну технологію створення логістичної системи, побудованої на блокчейн платформі, на прикладі системи логістичних перевезень.

Автор дисертації є співрозробником спеціальної системи логістики для фермерського господарства, розробленої та впровадженої в фермерському господарстві «Надія», що знаходиться в с. Марійка, Жашківського району, Черкаської області, на основі досліджень в дисертації.

Практичне значення отриманих результатів полягає у доведені наукових результатів до практичного використання, які підтверджені актами впровадження ФГ «Надія», ТОВ «Смарт Трейдинг», ПрАТ «Ксібекс», Херсонського Державного Університету, приватного підприємства ЛітСофт.

Особистий внесок здобувача. Основні наукові результати дисертаційної роботи отримані автором самостійно на основі досліджень відділу теорії цифрових автоматів, а у публікаціях, написаних разом із співавторами, внесок дисертанта є наступним:

[1] – автором створено огляд по рішенням з використання розподілених систем в глобальній логістичній системі та в складській логістиці та документообігу, окреслено основні проблеми безпеки в ланцюжку постачання товарів, поставлено задачі по використанню формальних алгебраїчних методів з метою аналізу та дослідження властивостей перевезень та впровадженням методу інсерційного моделювання для дослідження їх властивостей; [2][10] – автором створено модель логістичної системи на основі алгебри поведінок для фермерського господарства та для логістичної системи міжнародних перевезень; [3] [4] – автором створено модель перевірки безпечності використання систем життєдіяльності розумних будиків та виявлення зловмисних атак з метою протидії; [6] – вивчено проблеми використання картографічної інформації в автоматизованих аграрних системах управління, розроблено методи адаптації використання картографічних даних вітчизняного виробництва в прикладних аграрних програмах з використанням геолокації, зокрема на прикладі точного землеробства; [7] – розроблено теоретичні основи використання геоінформаціних систем в малих фермерських господарства; [8] – розроблено теоретичні основи, методи та засоби побудови інтерактивних геоінформаційних комплексів реального часу для використання в аграрному господарстві та обслуговуючій логістиці; [9] – вивчено та систематизовано напрями та тенденції автоматизації та інформатизації сільськогосподарських підприємств, описано методологію побудови систем управління для малих фермерських господарств з фокусом на посилення використання геоінформаційних систем в агарній сфері, розроблено методи реалізації функціональних компонентів системи для обслуговуючої логістичної системи; [11] [12] – автором створено модель протоколу консенсусу в блокчейн платформі для використання в логістичній системі відкритого та закритого типів, та досліджено властивості безпеки таких логістичних систем.

Апробація результатів дисертації. Результати дослідження повідомлялись та обговорювались на наступних наукових конференціях та міжнародних семінарах:

1. Global and regional problems of informatization in society and nature management ‘2017. V International Scientific and Practical Conference (22-23 June 2017, Kyiv).

2. Life safety in transport and production - education, science, practice. The IV International scientific-practical conference (14-16 September 2017, Herson).

3. Ideas of academic V.M. Glushkov and modern problems of artificial intelligence. The VII Ukrainian Scientific Conference “Glushkov Readings” (29 November 2019, Kyiv).

4. The 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (17-18 September 2020, Dortmund, Germany)

5. Conference on Mathematical Foundations of Informatics: Proceedings MFOI-2020 (12–16 January 2021).

6. IntelITSIS’2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 23–25, 2022, Khmelnytskyi, Ukraine.

Публікації. Основні положення та результати дисертаційної роботи опубліковано в 6 наукових працях. Серед них 4 – у виданнях, що входять до міжнародних наукометричних баз, 3 – у фахових наукових виданнях. Крім того 6 праць опубліковано у матеріалах і тезах міжнародних та міжвідомчих конференцій (у тому числі 1 одноосібно).

Структура і обсяг дисертації. Дисертаційна робота складається із анотації, вступу, чотирьох розділів, висновків, списку використаних джерел із 94 найменувань. Основний текст роботи викладено на 147 сторінках та 4 додатків на 61 сторінках. Повний обсяг дисертації становить 208 сторінки, в тому числі 2 таблиці, 24 рисунки.

РОЗДІЛ 1.
СУЧАСНІ СИСТЕМИ ЛОГІСТИКИ ТА МОДЕЛЬНИЙ МЕТОД РОЗРОБЛЕННЯ

Системи логістики – це багатоагентні системи із складною поведінкою, тому розробка таких систем є багаторівневим процесом, що покриває всі етапи розробки: створення вимог, дизайн, кодування, тестування. Такі системи є критичні до безпеки та забезпечення надійності їх роботи є головною задачею при їх проектуванні. Безпека в таких системах полягає в стійкості до атак, тому верифікація властивостей кібербезпеки має входити в розробку, як необхідна складова процесу. Крім того сам процес розробки має бути регламентованими чіткими критеріями надійності на кожному етапі, тому мають застосовуватись формальні методи та використання моделей для виконання процедур верифікації та тестування.

Модельний спосіб розробки є одним із шляхів розробки, що забезпечує надійність функцій системи, що розробляється. Використання моделей дає змогу застосовувати методи безпеки, що базуються на формальних методах, на всіх етапах розробки.

Сучасні системи логістики є розподіленими системами, а саме представляє собою набір незалежних компонент, що не мають спільного керування, окрім правил передачі даних між собою. Кожна компонента виконує певну функцію, що визначає одну із операцій логістичної системи та передає дані та іншій компоненті.

Розглянемо поняття логістичної системи із подальшим оглядом сучасної індустрії логістики, що використовують різні технології розподілених систем, які в певній мірі забезпечують дотримання властивостей безпеки.

1.1. Поняття логістичної системи, основні терміни та агенти в системах логістики.

В зв’язку зі зростаючими об’ємами промислового виробництва, ростом населення, глобалізацією та міжнародною торгівлею, в світі зростає роль

міжнародної логістики та все більше уваги привертається до зростання її ефективності та вирішення проблем. Усі учасники (агенти), що беруть участь в логістичних операціях, внаслідок своєї взаємодії формують стрункy та чітку систему постачання товарів, а також збору та обміну інформацією про такі взаємодії. Такі системи називаються логістичними. Розглянемо найбільш поширені визначення логістичної системи.

Логістична система – це цільова інтеграція логістичних елементів у межах певної економічної системи з метою оптимізації процесів трансформації матеріального потоку. Об’єктивна підстава створення логістичної системи – реалізація синергічного ефекту, який виявляється у: 1) загальному прискоренні матеріального потоку, що адекватно підвищенню реакції на бажання клієнта, 2) зменшенні сукупних витрат за рахунок усунення конфліктів часткових витрат, підвищенні рівня логістичного сервісу (рівня та якості обслуговування), що адекватно збільшенню додаткової вартості (корисності) для клієнта і формуванню додаткових конкурентних переваг [1].

Крикавський Є.В. та Чорнописька Н.В. приводять ще такі можливі визначення терміну «логістина система» [2]:

Логістична система — це складна організаційно завершена (структурована) економічна система, що складається з взаємозалежних у єдиному процесі управління матеріальними і супутніми їм потоками елементів — ланок, сукупність яких, межі і задачі функціонування об'єднані внутрішніми цілями організації бізнесу і (або) зовнішніми цілями.

Логістична система — адаптивна система із зворотними зв'язками, яка виконує ті чи інші логістичні функції (операції), складається із підсистем і має розвинуті внутрішньосистемні зв'язки та зв'язки із зовнішнім середовищем

Логістична система — організаційно-управлінський механізм координації, який дає змогу досягти ефекту завдяки чіткій злагодженості у діях спеціалістів різноманітних служб, що беруть участь в управлінні матеріальним потоком.

Різновиди класифікації логістичних підсистем [3]:

- інституціональна класифікація: логістичні системі мікрологістичні (на рівні підприємства), металогістичні (логістичні ланцюги), мезологістичні, макрологістичні, зовнішні логістичні системи (міжсистеми),

- функціональна класифікація логістичних підсистем: підсистема реалізації замовлення, транспортування, формування запасів, складського господарства, пакування, підсистема обслуговування клієнта;

- фазова класифікація логістичних підсистем: підсистема постачання, виробництва, дистрибуції (збуту), переробки та утилізації відходів, інтегрована підсистема матеріальної логістики, інтегрована підсистема маркетингової логістики, інтегрована логістична підсистема постачальників, інтегрована логістична підсистема замовників, інтегрована логістична підсистема торгівлі;

- класифікація за функціями управління: підсистема логістичного планування, логістичного управління, організації логістики, логістичного контролю, стратегічного логістичного управління, оперативного логістичного управління, система інтегрованого логістичного управління;

- предметно-структурна класифікація логістичних підсистем: підсистеми інтегрованого переміщення товарів (фізичні структури), інтегрованого переміщення інформації (інформаційні структури), регулювання й організаційно-інституціонального забезпечення логістичних процесів (логістичні структури);

- класифікація за ознакою формування ефективності: підсистеми логістичних витрат (витрати), послуг і логістичного обслуговування (результати).

Агент логістичної системи — це функціональний об'єкт, що прагне досягти своєї мети діяльності, пов'язаної з певною логістичною функцією, за виконання відповідних логістичних операцій. Макрологістична система являє собою велику логістичну систему, агентами (діючими елементами) якої є окремі суб'єкти господарювання, а мікрологістична система охоплює внутрішньовиробничу логістичну сферу одного підприємства. Агентами мікрологістичної системи є усі її учасники, які, в основному, належать одному власнику (підприємсту), тому пов’язані між собою однією ціллю діяльності та не володіють автономністю та свободою дій та діють в межах єдиної ієрархічної структури. Агенти

макрологістичної системи, навпаки, є автономними та часто змушені діяти в умовах жорсткої конкуренції, володіють повною свободою дій та вибору. Агентами макрологістичної системи, зокрема її найбільшої форми– міжнародної глобальної логістичної системи, є: виробники продукції, продавці, покупці, імпортери та експортери, торгові посередники, склади, митні термінали та інспектори, внутрішні наземні перевізники, міжнародні морські перевізники, порти, припортова інфраструктура, банки, страхові компанії, державні установи та контролюючі органи, міжнародні організації та союзи, власники транспортних засобів, вагонів та кораблів, податкові органи, незалежні організації контролю якості та багато інших.

Сукупність цих агентів утворює логістичний ланцюг. Суб'єкти господарювання та підрозділи підприємств, через які послідовно проходить логістичний потік, складають логістичний ланцюг, а сукупність ланцюгів — логістичну мережу. Логістичний ланцюг (англ. logistical chain) або ланцюг постачання — це лінійна інтегрована сукупність агентів логістичної системи (виробників, посередників, складів та інше), які виконують логістичні операції, спрямовані на доведення зовнішнього матеріального потоку від однієї логістичної системи до іншої чи від безпосереднього виробника до кінцевого споживача. Термін "ланцюг поставок" є дуже широким та охоплює як найпростіші лінійні логістичні ланцюги, так і широко розгалужені логістичні системи.

1.2. Блокчейн, як платформа для розподілених логістичних систем.

В зв’язку зі зростаючими об’ємами промислового виробництва, ростом населення, глобалізацією та міжнародною торгівлею, в світі зростає роль міжнародної логістики та все більше уваги привертається до зростання її ефективності та вирішення проблем.

IT сектор також не міг лишитись в стороні від глобальних трендів та тенденцій та широко залучається до міжнародної логістики, однією з таких прогресивних технологій є технологія розподілених систем [4], [5], що характеризується принципом децентралізації, та технологія блокчейн, як найпопулярніший різновид розподіленої системи, яка показала надзвичайну

ефективність, в тому числі і в логістичних системах різного масштабу та складності.

Розподілена база даних (англ. distributed database, DDB) — сукупність логічно взаємопов'язаних баз даних, розподілених у комп'ютерній мережі. Логічний зв'язок баз даних в розподіленій базі даних забезпечує система управління розподіленою базою даних, яка дозволяє управляти розподіленою базою даних таким чином, щоб створювати у користувачів ілюзію цілісної бази даних.

Блокчейн технологія є технологією загального, надзвичайно широкого призначення [6] та може застосовуватись в будь-якій галузі промисловості. Згідно визначенню автора цієї технології Сатоші Накамото, це «розподілена база даних з записами або публічний реєстр всіх транзакцій або цифрових подій, що були виконані та поширені серед всіх учасників» [4]. Система є надійною та захищеною – записи перевіряються публічними реєстрами та ніколи не можуть бути видаленими в майбутньому, тобто зберігаються завжди. Головними перевагами технології блокчейн є прозорість та висока ефективність [7].

В мультимодальних перевезеннях в міжнародній логістиці використання технології блокчейн в розрізі швидкості та надійності контролю та обміну інформацією виглядає особливо перспективно, так як там задіяно безліч учасників з різними функціями в різних країнах світу (виробник товару, склад, експортер, митниця та контролюючі органи держави, власники засобів перевезень, портові оператори, оператори перевантаження, імпортери, продавці та покупці, проміжні власники товарів). Тому оперативний контроль та реагування має особливе значення.

Технологія блокчейн викликала інтерес та швидко розвивається в багатьох галузях: енергетика, охорона здоров’я, фінанси, державний контроль та інше. Не виключенням є і міжнародна логістика, проте, на жаль вивчення можливостей застосування технології блокчейн в міжнародній логістиці не є достатнім [8]. Проте, без сумнівів, міжнародна логістика є однією із галузей, в якій технологія

блокчейн має надзвичайні перспективи застосування, тому кількість наукових досліджень буде зростати.

Проте ряд скандинавських країн вже виявило зацікавлення та використовують блокчейн в своїй діяльності. Це пов'язано з тим, що вони в силу обставин вимушені використовувати в основному міжнародні морські перевезення. Ряд значних міжнародних морських операторів вже почали використовувати технологію блокчейн та вивчають можливості більш широкого використання [9].

Також міжнародне судоходство є галуззю, яка жорстко регулюється Міжнародною Судоходною Організацією (International Maritime Organization -IMO) та законодавством Евросоюзу [10]. Законодавство стає щороку все суворішим, вимоги все вищими, тому ведучі морські перевізники виявили зацікавленість в подальшому розвитку технології блокчейн для оптимізації своїх бізнес-процесів.

Згідно з дослідженням міжнародної сертифікаційної та класифікаційної спільноти DNV GL один незапланований простій під завантаження-вивантаження чи запізнення може коштувати логістичному оператору $2-5 млн в день. Приблизно 50% трапляються через різноманітні механічні поломки. Але недостатній менеджмент інформації призводить до прихованих втрат, який може вимірюватись в розмірі до 20% операційного бюджету [11]. Тому вони заявляють про своє бажання вивчати та розробляти технологію великих масивів інформації в розподілених системах з подальшим її застосуванням в автоматизованій системі менеджменту ризиків використання активів. Вони називають свою технологію «cloud-based digital twin» (цифрові двійники в хмаровому сховищі).

На сьогоднішній день потік інформації типової системи постачання складає близько 100 гігабайт в день [12], по оцінкам експертів цей потік інформації зросте до 35 зетабайт до 2030 [13].

Ще одним прикладом є портові оператори. Морські перевізники та інші логістичні оператори жорстко конкурують за місце в порту та портовому складі, створюючи при цьому величезні черги, але зібрана інформаціє не поширюється між пірсами навіть одного порту. Тому системі надзвичайно важко проаналізувати та розподілити інформацію від порту до зацікавлених сторін з метою оптимізації

використання доступного ресурсу обслуговуючої логістики. Таким чином порти стають «вузьким місцем» логістичних операцій, а об'єм таких операцій щодня зростає. Порт Гамбург оцінює, що число обороту контейнерів, що проходять через порт, зросте до 25 мільйонів в рік до 2025 року, тому обробка такого величезного об'єму потребує залучення нових технологій обробки великих масивів інформації та швидкого та надійного розподілу інформації між зацікавленими сторонами, що, без сумніву, під силу блокчейн-системам [14].

Ще один інноваційний проект в міжнародній логістиці розпочав Морський та портовий департамент порту Сінгапур під назвою Smart Port Challenge. Вони співпрацюють з портом Ротердам, який розпочав подібну ініціативу (проект Onboard). Вони залучають інтернет речей до індустрії морських перевезень, пропонуючи відкриту платформу з повністю інтегрованим програмним забезпеченням для спостереженням в реальному часі за рухом суден та портовими операціями [15].

Велика фрахтова форвардна компанія Marine Transport International Limited (MTI) заявила, що з середини 2016 року використовує систему блокчейн відкритого типу під назвою TrustMeTM для контролю за відповідністю ваги брутто завантажених контейнерів, що надходять на завантаження, новим нормам постанов Міжнародної Морської Організації «SOLAS», які вступили в дію з липня 2016 року. Нове законодавство перекладає на відправника відповідальність за порушення норм гранично допустимої ваги брутто до етапу, коли контейнер поступає на завантаження на судно. Компанія почала використовувати технологію блокчейн TrustMeTM через необхідність надання бесперебійних та реальних актів перевірок портовим державним органам, перевізникам та власникам вантажу, таким чином уникаючи посередників при передачі інформації, приватних баз даних, логів та паперової роботи [8].

DNV GL являється міжнародною сертифікаційною та класифікаційною спільнотою, ключовою компетентністю якої є оцінка, консалтинг та менеджмент ризику в морській логістиці. DNV GL є найбільшою класифікаційною спільнотою, в реєстрі якої знаходиться 13175 суден та мобільних морських судоходних

установок сумарною водотоннажністю 265.4 млн. т., що являє собою 21% світового ринку [16].

DNV GL помістила всі свої 90 000 сертифікатів по системам менеджменту та ланцюгів постачання в закриту систему блокчейн, став першою, хто це зробив це в галузі сертифікації морської логістики. Кожен сертифікат має цифрову ідентифікацію, відслідковуваність та зберігається в приватній системі блокчейн. Технологія запобігає фальсифікації сертифікатів та дає можливість компанії повідомлювати про свої сертифікати прозорим та безпечним способом. Серед головних переваг користування технологією блокчейн компанія зазначає «аутенчичність (підтвердження істинності), децентралізацію та кодування». Також з вересня 2017 року компанія оснащує всі випущені та перевипущені сертифікати QR-кодами, а бажаючий за допомогою мобільного телефону може перевірити його достовірність, відсканувавши код, а система перевірить сертифікат в мережі блокчейн [17].

Також компанія DNV GL розробляє проект «Blockchains in the shipping world» (Блокчейн в світі поставок), який включає наступні модулі: складування, відвантаження, митниця, страхування, оплата. [18]

Більшість сучасних новітніх розробок фокусується на аналізі великих баз даних та на штучному інтелекті з метою зниження споживання палива та інших ресурсів, зниження викидів CO2 в атмосферу, оптимізації маршрутів, скорочення затрат часу та праці, інтеграцію з іншими популярними технологіями, такими як дрони та проектування вищезгаданих кораблів без людей на борту.

Проте бази даних, аналітичне програмне забезпечення, додатки, що базуються на інтернеті речей потребують діяльності посередників в ланцюгу доступу, що додає лишніх кроків, затрат ресурсів та коштів. Помітно, що в галузі зростає увага до подальших інновацій та технологій. Разом з тим технологія блокчейн може бути тим рішенням, яке задовольнить зростаючі потреби міжнародної логістики, такі як децентралізація, кодування, безпечність, доступність, швидкодія.

1.3. Розподілені системи, контрольовані приладами геолокації.

Доктор Мартін Стопфорд, один з передових світових експертів та аналітиків в економіці та морських перевезеннях, визначає, що «діджіталізація» (перехід в цифровий світ) є єдиним та невідворотним виходом для галузі морських перевезень. Аналізуючи економічні цикли та розвиток судобудівництва та перевезень з початку 20 століття, він заявляє, що три методи здатні змінити бізнес-модель циклів перевезень за допомогою діджіталізації, яка так необхідна галузі: «розумні» кораблі, «розумний» флот (з інтегрованою системою відслідковування та контролю торгового флоту як єдиної цілісної системи) та «розумна» глобальна логістика (з інтегрованою системою постачання «від-дверей-до-дверей»). Він додає, що великі культурні зміни повинні передувати таким змінам та для подальшого застосування додатків за технологією блокчейн в морській логістиці. Середнє комерційне судно має близько 2000 різних датчиків та сенсорів, але вони використовуються в малій мірі, в основному людьми, а повинні контролюватись автоматизованою системою. Така система передбачалась ще в 70-80 роки, але лише зараз системи GPS-навігації та блокчейн технології дозволяють втілити їх в життя. Він заявляє, що «мати технологію – це перший крок, проте масиви інформації повинні використовуватись заради того, щоб показати зростання показників продуктивності та ефективності галузі» [19].

На сьогодні практично кожне вантажне судно, що здійснює міжнародні перевезення, оснащене системою ідентифікації по радіочастотам (Radio Frequency Identification – RFID). Ця система інтегрована з «інтернетом речей», початково для відслідковування місцезнаходження судна, а потім послужила для більш широкого застосування, та зараз є основою для передових технологій на основі технології блокчейн [20]. Радіочастотне розпізнавання здійснюється за допомогою закріплених за об'єктом спеціальних міток, що несуть ідентифікаційну та іншу інформацію. Цей метод вже став основою побудови сучасних безконтактних інформаційних систем, і має стійку назву RFID-технології. Використання RFID-міток зросло до 209 мільярдів одиниць до 2021 року [21], а технології, що використовують такі мітки, дадуть змогу знизити операційні витрати на 10–25%

[22]. Проте, найбільш вживаною технологією для інтернету речей є GPS-навігація, що дозволяє з найбільшою точністю слідкувати за місцеперебуванням об’єктів [14].

В березні 2017 року Норвезький Університет Науки та Технологій разом з комерційними партнерами Kongsberg Seatex, Marintek, Maritime Robotics та Rolls-Royce Marine у водах фьйорду Трондхейм розпочали пілотний проект по запуску першого в світі автоматизованого самокерованого судна без жодної людини на борту, супроводжувати який будуть повітряні та підводні дрони.

1.4. Огляд використання передових блокчейн-рішень в логістиці

9 серпня 2018 року найбільша судоходна компанія на планеті Maersk та IT гігант IBM презентували [23] свою спільну розробку TradeLens [24] – комплексну платформу для контролю відвантажень та відслідковування суден та вантажів на базі технології блокчейн, побудовану на відкритих стандартах. На момент анонсування 94 організації брали активну участь або тестували платформу TradeLens. Екосистема TradeLens включала: понад 20 портів та портових операторів, що включають в себе 234 морських коридори, 3 морських перевізники, митні органи Нідерландів, Саудівської Аравії, Сінгапуру, Австралії та Перу, великі логістичні та транспорті компанії. В загальному, морські перевізники, об’єднані в систему TradeLens, покривають 20% долі глобального ринку ланцюгів постачання [25], [26].

В травні 2019 року перевізники MSC та CMA CGM заявили про своє бажання залучитись до мережі TradeLens. Таким чином незабаром TradeLens об’єднає трьох найбільших перевізників в світі: Maersk на першому місці, MSC на другому та CMA CGM на третьому. [27], [28]

Система TradeLens базується на трьох компонентах: Екосистема, Платформа та Торговий майданчик [29].

Екосистема – це, власне, бізнес-мережа учасників: відправники, порти, термінали, митниця, морські та наземні перевізники.

Платформа доступна у вигляді програмного забезпечення та поєднує екосистеми в єдине ціле за допомогою відкритих стандартів. Працює за допомогою

так званого «Hyperledger Fabric blockchain» та IBM Cloud (хмарне рішення від IBM).

Торговий майданчик – відкриті додатки та сервіси дозволяють TradeLens та третім сторонам публікувати пропозиції щодо сервісів поверх Платформи.

За основу TradeLens використовує технологію блокчейн від IBM для діджіталізації ланцюга постачання, дозволяючи торговим партнерам співпрацювати, установлюючи загальне широке бачення транзакції без приватності чи конфединціальності. Відправники, морські перевізники, портові оператори, порти та термінали, наземні транспортні компанії, митні органи можуть взаємодіяти більше ефективно через доступ в реальному часі до бази даних та відправних документів, а також до Інтернету Речей та даних різноманітних сенсорів широкої дії – від температурного контролю до ваги контейнера.

За допомогою блокчейн смарт-контрактів від IBM TradeLens дає можоливість цифрової взаємодії багатьох сторін, що залучені до операцій міжнародної торгівлі. Модуль торгових документів, що був представлений у бета-режимі під назвою ClearWay дозволяє імпортерам/експортерам, митним брокерам та третім зацікавленим особам, таким як державні контролюючі органи, взаємодіяти в міжорганізаційних бізнес-процесах.

Протягом 12-місячного терміну випробування Maersk та IBM працювали з десятками екосистем партнерів для ідентифікації можливостей запобігти запізненням, спричинених помилками в документації, затримками в поданні інформації та інше. Один приклад продемонстрував як TradeLens зміг скоротити транзитний час поставки товару в США на 40%, зекономивши тисячі доларів втрат. Інші приклади показали, що TradeLens змогла скоротити операційні проміжні дії, щоб відповісти на типове питання клієнта «де мій контейнер?» з 10 кроків та 5 осіб до одного кроку та однієї особи.

Як альтернативу TradeLens та як його основний конкурент в кінці 2018 року створено подібний проект під назвою Global Shipping Business Network (GSBN) [30], [31], ще на початку процесу тестування включав дев’ять великих морських перевізників, випробування проводились на базі порту Роттердам, розробкою

програмного забезпечення на основі технології блокчейн закритого типу займалась IT-компанія з Гонконку CargoSmart, яка є структурним підрозділом OOCL, створеним спеціально для цієї цілі.

Функціонал та технологія GSBN схожі до TradeLens, проте в засобах масової інформації не розголошуються достатньо широко [32]. Зазначається, що CargoSmart залучить свій багаж знань по доменам вантажоперевезень, аналізатори великих баз даних та багаторічний досвід розробок програмного забезпечення з штучним інтелектом, інтернетом речей та блокчейн технологіями для того, щоб допомогти мережі учасників покращити свої логістичні операції.

Таблиця 1.1.

Порівняння кількості учасників платформ TradeLens та Global Shipping Business Network станом на 22 квітня 2019 року [33]

TradeLens		Global Shipping Business Network	
Перевізник	Кількість контейнерів	Перевізник	Кількість контейнерів
Maersk	4,115,597	COSCO	2,822,551
PIL	405,870	CMA CGM	2,661,799
ZIM	305,247	Evergreen	1,248,375
Seaboard Marine	35,708	Yang Ming	673,354
Всього:	**4,862,422**		**7,406,079**

Було заявлено, що на етапі тестування нового програмного забезпечення воно буде працювати з документооборотом та перевезеннями небезпечних вантажів з метою спрощення процесу обміну інформацією з регулятивними органами та прискорення процесу відвантажень.

Проте, як зазначалось раніше, в травні 2019 року CMA CGM заявила про своє бажання залишити проект Global Shipping Business Network та долучитись до проекту TradeLens. Вірогідною причиною є повільні темпи розробки та перенесення старту початку тестової роботи платформи.

Ще одним інноваційним проектом є проект Xeneta від американського розробника Aberdeen Group [34]. Проект фокусується на відслідковування в реальному часі вартості морського фрахту заданого маршруту від різних перевізників, їх порівняння та обрання рекомендованого перевізника (комбінації перевізників з перевантаженням в портах транзиту) з критерієм вартості та/або часу. Хоч розробники і заявили про намір виносу бази даних в мережу блокчейн, проте на сьогодні дана технологія ще не використовується.

Технологія блокчейн також була застосована для створення системи єдиної торгівлі електроенергією між торговими біржами. Компанія Волт Маркетс, що є продавцем відновлюваної енергії, застосував відкритий блокчейн для підтвердження надійності продаж та надав можливість відслідковувати сертифікати відновлюваної енергії. В лютому 2017 інший торговий дім «Меркурія» застосували таку ж технологію в сферу торгівлі нафтою з банками ING та Société Générale banks [35].

Існує декілька інших можливих варіантів застосування технології блокчейн в міжнародній логістиці, в основному щодо питання вирішення регулювання документообігу та власності на товар. Блокчейн може вирішити питання діджіталізації бортових коносаментів, основного документу при міжнародних морських перевезеннях. Бортовий коносамент являється підтвердженням прийняття на борт вантажу перевізником та підтвердженням права власності. Отримання цього документу часто затягується через банківські процедури, що призводить до того, що корабель перевізника прибуває в порт призначення до моменту отримання покупцем бортового коносаменту.

Ізраїльський проект Wave сфокусований на створенні «безпаперової» торгівлі, зі створенням усіх транспортних документів (бортовий коносамент, інвойс, сертифікат походження, сертифікат відповідності, банківський акредитив) в електронному виді та подальшому зберіганні за технологією блокчейн. Їх головним конкурентом є американська фірма Skuchain, що знаходиться в Каліфорнії.

Шведська компанія SkyCell створила спеціальний блокчейн-асоційований контейнер з температурним режимом для повітряних перевезень біофармацевничних препаратів. Як відомо, даний тип продукції особливо чутливий до перепадів температур, тому контейнер здатен підтримувати температурний режим з відхиленням до 0,1% від заданого, однак розміри такого контейнера дуже малі та підходять лише для перевезень специфічної продукції авіаційним транспортом [36].

Американська асоціація вантажних перевізників BiTA створила на базі технології блокчейн мобільний додаток для пошуку клієнтів та вантажів, який покриває 85% наземних вантажних перевезень в США та Канаді [37].

Microsoft також активно підключились до тенденцій використання блокчейн. В партнерстві з розробником програмного та апаратного забезпечення для відслідковування вантажів та транспорту Adents розробили платформу Adents NovaTrack, яка дозволяє побачити увесь шлях руху товарів та місцезнаходження протягом всього ланцюжка постачання. Початково платформа розроблялась з метою боротьби з підробками медичних препаратів та вакцин, але пізніше розробники розширили можливу сферу застосування технології [38].

Існують також ряд проектів від менш відомих розробників, які покликані спростити взаємодію між учасниками міжнародної торгівлі та логістики, коротко розглянемо найбільш перспективні з них. Розробники Citizens Reserve анонсували запуск закритого блокчейн протоколу «Zerv», який покликаний стати «операційною системою для ланцюга постачання». Мережа, доступ до якої забезпечується токенами з реальним фінансовим забезпеченням, створена підтримувати стабільні безризикові транзакції між ключовими учасниками в ланцюгу постачання [39]. Автоматизацією комерційних процесів та розрахунків зайнялись розробники компанії Libelli. Вони розробили систему блокчейн, яка виступає в ролі банківського агента-посередника між продавцем та покупцем. Система створює систему смарт-контрактів згідно побажань учасників угоди та є альтернативою паперового банкового акредитиву. Libelli заявляють, до їх система виконує ту ж роботу, але вдесятеро раз дешевше, ніж банківська установа. Також

вони мають намір добавити функції відслідковування транспортного шляху, походження товару, аукціонні торги та інше, чим не займається банк-посередник [40]. Інший проект під назвою OriginTrail займається наданням надійної та перевіреної інформації учасникам ланцюжку постачання. Ланцюг постачання часто представляється, як один з найефективніших способів застосування технології блокчейн, але розробники заявляють, що поточні децентралізовані рішення не здатні забезпечити відповідний рівень взаємодії, стабільності та ефективності. Заявлено, що їх протокол OriginTrail відповідає високим вимогам децентралізованої мережі [41]. Варто також згадати проект ShipChain, ще одну систему відслідковування вантажів, що опирається на блокчейн. Система відслідковує товар з моменту, коли він покидає конвеєр фабрики, та до його доставки до дверей покупця. Відповідна інформація продукується на всіх етапах ланцюгу постачання. Система може виконувати смарт-контракти, як тільки виконуються задані вимоги. Для прикладу, тільки водій з логістичної компанії підтверджує доставку товару покупцю, система автоматично маркує продукцію та завдання, як виконане, та надсилає новий маршрут водієві [42].

Українські науковці також не відстають від світових тенденцій та беруть активну участь в запровадженні передових технологій в логістиці. Відповідні теоретичні напрацювання проводяться в багатьох наукових інститутах України, зокрема в Інститут кібернетики ім. В.М. Глушкова, Одеському національному морському університеті [69], Київському національному університеті імені Тараса Шевченка, зокрема роботи професора А.В. Анісімова [70] та багатьох інших.

Однак потрібно зазначити, що через новизну технології в світі в Україні розроблення практичного застосування розподілених систем в міжнародній логістиці знаходиться на початковому етапі та поки, в основному, містить теоретичні та консультаційні рекомендації, практичні впровадження знаходяться в стані розробки та вивчення.

Починаючи з 2016 року українські урядові установи проводять міжнародні конференції з передовими розробниками щодо запровадження використання технології блокчейн в міжнародній логістиці в основних портах України [71].

В кінці 2016 року Правління НБУ одобрило та презентувало дорожну карту Cashless Economy, в якій вперше прописано плани по використанню технології blockchain в Україні.

Міністерство аграрної політики та продовольства в кінці 2017 року заявило про плани запровадити технологію Blockchain в Державному земельному кадастрі.

В квітні 2017 року Кабинет міністрів домовився з американською технологічною компанією Bitfury Group про створення в Україні повномасштабної системи електронного управління з технологією Blockchain.

Міністр інфраструктури України Володимир Омелян заявив, що порти Миколаїв, Одеса і Південний задіють систему e-Port 4.0, окремі компоненти якої будуть працювати на базі технології блокчейн [72]. За словами міністра, по всій морської галузі України буде задіяний успішний досвід стивідорної компанії «Ольвія» щодо впровадження системи електронного управління e-Port. Це зробить діяльність портів більш прозорою та ефективною. Омелян підкреслив, що в порту «Ольвія» на 34,4% прискорилося оформлення вантажів, а також терміни їх обробки. За його словами, очолюване ним відомство вже два роки активно впроваджує цифрові технології у морську галузь та займається автоматизацією процесів у портах. Для розробки системи електронного управління портом e-Port були залучені Адміністрація морських портів України, морські порти, державні стивідорні компанії і ВАТ «Укрзалізниця».

Однак даний проект ускладнений потребами в значних коштах інвестицій, зокрема для впровадження цієї системи логістичним компаніям потрібно вкласти у розвиток блокчейну $12 млрд на протязі 3-5 років – а потім щороку інвестиції в цю галузь будуть збільшуватися майже вдвічі.

1.5. Смарт-контракти на базі блокчейн та мова Solidity

Смарт-контракт (англ. Smart contract — “розумний контракт”) — різновид угоди в формі закодованих математичних алгоритмів, укладення, зміна, виконання і розірвання яких можливо лише з використанням комп'ютерних програм (Блокчейн платформ) в рамках мережі Інтернет. Можна сказати, що йдеться про

врегулювання відносин сторін шляхом закріплення їх вираженої волі у формі певного коду, який придатний для зчитування комп'ютером. Смарт-контракт базується на основі чіткої логіки і перевірки, або виконання через криптографічні протоколи та інші механізми цифрової безпеки, являє собою різке поліпшення в порівнянні з традиційним контрактом, навіть для деяких традиційних видів договірних положень, які можуть бути передані під владу комп'ютерних протоколів [43], [44].

Кілька формальних мов були розроблені та запропоновані для визначення договірних положень. [45], [46].

Для розуміння поняття «смарт-контракт», що діє в мережі блокчейн «Ethereum» необхідно спочатку згадати біткоін – першу та все ще найпопулярнішу відкриту систему блокчейн. Блокчейн біткоін був розроблений з єдиною метою: передавання валюти біткоін від одного власника до іншого. Однак з поширенням технології люди почали додавати «метадані» в процесі транзакцій для досягнення інших цілей, таких як завірення та коригування документів.

Система блокчейн Ethereum виникла дещо пізніше та на відміну від інших систем блокчейн Ethereum дає змогу будь-кому створювати «контракти» всередині блокчейн. Цей контракт є комп’ютерною програмою з асоційованою мініатюрною базою даних, яка може бути змінена лише програмою, яка нею володіє. Якщо користувачі блокчейн бажають внести зміни до бази даних, вони повинні вислати повідомлення з цифровим підписом до цього контракту. Код контракту перевіряє ці повідомлення та вирішує як реагувати [47].

Контракти в Ethereum можуть бути написані на одній з декількох мов програмування, найпопулярнішими з яких є Solidity та Serpent. Як і більшість мов програмування, вони відповідають критерію «повноти за Тюрингом», що також означає, що вони мають кільцеву структуру, тобто виконують операції повторно до того часу, коли виконуються задані умови.

Коли в блокчейн створюється Ethereum-контракт, він визначає початкові дані та стан бази даних. Потім контракт призупиняє роботу, очікуючи поки користувач блокчейн (або інший контракт) відправляє повідомлення про транзакцію, після

чого контракт знову починає працювати. В залежності від прописаного коду він може ідентифікувати джерело повідомлення, бути тригером для іншого контракту, змінювати базу даних чи надсилати відповідь на запит користувача. Всі ці кроки виконуються незалежно на будь-якій ноді в мережі блокчейн з ідентичними результатами.

Виконання Ethereum-контрактів вимагає певних комп'ютерних потужностей для проведення розрахунків та обчислень та зберігання інформації в мережі. Дана проблема вирішується комісією від клієнта. Користувач, який створює транзакцію, платить за комп'ютерні обчислення, які є тригером контракту, цю комісію отримує «майнер», що підтверджує інформацію та заносить її в блок. Комісія утримується покроково згідно етапам виконання контракту.

Solidity — одна з чотирьох мов (три інші: Serpent, LLL і Mutan), спроектованих для трансляції в байт-код віртуальної машини Ethereum. Отримала широке поширення з появою технологій блокчейну, зокрема стека технологій на основі Ethereum, для створення програмного забезпечення розумних контрактів. Мова була запропонована в серпні 2014 року Гейвіном Вудом (Gavin Wood). Надалі розробка мови була виконана під керівництвом Крістіана Райтвизнера (Christian Reitwiessner) командою Solidity в рамках проекту Ethereum [48].

Solidity — об'єктно-орієнтована та предметно-орієнтована мова програмування високого рівня для виконання розумних контрактів, які керують поведінкою грошових рахунків всередині платформи Ethereum [49]. Мова Solidity була створена під впливом C++, Python та JavaScript та розроблена з ціллю взаємодії з віртуальною машиною Ethereum (Ethereum Virtual Machine або EVM). Solidity — статично типізована мова, що підтримує спадкування, бібліотеки та комплексну типізацію, визначену користувачем. На мові Solidity можна створювати контракти з функцією голосування, колективного фінансування (краудфандинг), сліпі аукціони та гаманці з мульти-підписами.

Хоч мова Solidity виглядає дещо схоже на JavaScript та C++, проте вона має ряд синтаксичних додатків, щоб бути придатною та зручною для написання контрактів в Ethereum.

1.6. Властивості безпеки логістичних систем.

Логістичні системи в своїй роботі вирішують ряд задач, щоб підтримувати безпеку в перевезеннях. Такі системи є критичними до безпеки, бо порушення властивостей може призвести до катастрофічних наслідків

Розглянемо типові проблеми, що зустрічаються при перевезенні товарів в міжнародній логістиці та які впливають на якість товарів та придатність до їх подальшого використання:

1) Дотримання відповідного температурного режиму. Деяким товарам (заморожена риба та м'ясо) необхідна достатньо низька температура (-18-20^{o}) для підтримання їх в стані глибокої заморозки, іншим товарам (свіжі овочі та фрукти, живі рослини, квіти) необхідна стабільна помірна температура (+15-20^{o}) для підтримання їх у свіжому стані. Навіть короткочасне коливання температури в заданому режимі може негативно позначитись на їх якості або навіть привести до повної втрати товару.

2) Дотримання відповідної вологості повітря. Відомо, що ряд товарів бояться високої вологості або різких перепадів вологості. Такими товарами, для прикладу, є електротехніка, медикаменти, ряд продовольчих товарів, таких як борошно, крупи та інше. Зазвичай прості вантажні контейнери є достатньо герметичними, щоб захищати товар від надмірного зволоження, також в контейнер в різних місцях закладається спеціальний силікатний гель, який вбирає надлишки вологи. Проте при тривалих морських та океанічних перевезеннях судно перетинає декілька кліматичних зон, потрапляє під зливи та й просто тривалий час перебуває під впливом вологого морського повітря.

3) Термін придатності продукту та його стан до початку транспортування. При дотриманні оптимальних умов транспортування товарів вони можуть витримати тривалий час міжнародних перевезень, який може сягнути двох місяців і навіть більше. Однак слід враховувати стан товарів до початку завантаження та час їх зберігання до моменту продажу або підтвердження угоди. Цей контроль

досягається шляхом ветеринарного, фіто-санітарного, епідеміологічного та іншого контролю, який виконується інспекторами відповідних державних органів під час процедур імпортного та експортного митного оформлення. Тому необхідно забезпечувати швидкий та безперешкодних доступ до цих документів усім зацікавленим сторонам. Для прикладу розглянемо, як Walmart, мережа супермаркетів в США, застосували технологію блокчейн з метою контролю повного ланцюжку постачання продуктів харчування. В кооперації з IBM вони почали роботу над пілотним проектом, який створений з метою посилення прозорості ланцюга постачання та більш ефективного відслідковування шляху товару, після випадку масового отруєння забрудненим листовим салатом. З системою блокчейн, над якою вони зараз працюють, було б можливо миттєво відслідкувати всі партії забрудненого салату, швидко їх прибрати з полиць та блокувати постачання від цього виробника [50]. Нещодавно Walmart провели відкрите публічне тестування відслідковування повного шляху постачання манго з Мексики та замороженої свинини з Китаю [51].

4) Можливість відслідковування повного ланцюжку постачання товарів з моменту їх виробництва та до моменту отримання кінцевим споживачем. Імпортер, покупець, державні органи країни-отримувача повинні бачити увесь ланцюжок руху товарів та країни їх транзиту з метою відповідного контролю в разі транзиту вантажу через зони можливої небезпеки. Яскравим прикладом служить нещодавні спалахи епідемії африканської свинячої чуми в Україні та ряду сусідніх країн, що призвело до створення карантинних зон та заборони імпорту чи експорту свинячого м’яса з цих регіонів з метою локалізації регіонів поширення епідемії [52], [53]. Однак недобросовісні продавці шляхом транзиту через треті країни та заміну документів через посередників змогли обходити ці заборони та обмеження. Частими також є випадки завезення незвичних для даного регіону шкідників з країн транзиту, в результаті ці шкідники через відсутність природніх ворогів швидко призводили до великих втрат урожаю [54], [55]. Тому зробити увесь ланцюжок постачання товарів максимально прозорим, відкритим та доступним для

контролюючих державних органів є вкрай важливо. Можливо навіть корегувати маршрут для уникнення зон карантину та епідемій в реальному часі.

5) Можливість відслідковувати місцезнаходження вантажу в реальному часі. Для планування оплат, обміну дозвільною документацією, відповідного державного контролю необхідно знати максимально точний час прибуття вантажу в країну призначення. Сучасні технології GPS та навігації дають змогу це зробити, проте доступ до цієї інформації повинні мати не лише логістичні оператори та перевізники, а й усі зацікавлені особи та органи. Однією з перших почала звертати увагу на цю проблему відома китайська торгова онлайн платформа Alibaba – через свою дочірню компанію Lynx International, Alibaba застосували технологію блокчейн для відслідковування інформації в міжнародній логістиці. Зараз система робить записи специфікації товару, деталі транспортування, митниці, державних інспекцій. Як вони самі зазначили, технологія блокчейн стала для них ідеальним рішенням [50].

6) Легкість документообігу та підтвердження оригінальності документів та країни походження вантажів. Лише два десятиліття назад документообіг був повністю паперовим, що призводило до величезних втрат часу на створення документів, перевірку їх оригінальності та пересилання поштою оригіналів зацікавленим особам та державним органам контролю. Сучасні технології електронного документообігу не лише значно скорочують час та полегшують створення документів, але й значно знижують можливість підробки або заміну документів, що було частим явищем в епоху паперових документів. Технологія блокчейн та смарт-контрактів взагалі зводить можливість підробки чи корегування документів шахраями практично до нуля.

7) Аварії та інші надзвичайні ситуації в морі, що призводять до псування чи втрати вантажу або значної затримки часу постачання вантажу внаслідок арешту судна до остаточного рішення суду, міжнародного морського арбітражу та реакції страхових компаній. Відомо багато випадків, коли через велику кількість учасників та постраждалих від морських аварій суд затягується на багато років. Наприклад, всесвітньо відомою є аварія в Чорному морі комерційного контейнерного танкера

CMA CGM Verlaine, що перевозив близько 8700 контейнерів різного типу на борту, та круїзного судна Odessa Star, в результаті чого значна частина вантажу була втрачена, інша частина значно пошкоджена, а контейнери з температурним режимом були відключені від мережі енергопостачання на значний час, крім того для утримання судна на плаву та запобігання втрати всього вантажу команда була вимушена викинути певну частину контейнерів за борт [56], [57]. Судові слухання почалися в жовтні 2010 року, однак станом на вересень 2019 року ситуація все ще остаточно не врегульована. Час судових слухань значно б скоротився за умови достатнього обсягу достовірної та перевіреної інформації про маршрути суден, їх дії безпосередньо до та під час аварій, а також за наявності в контейнерах датчиків, що збирають та передають інформацію про стан товару.

8) Своєчасна оплата за товар та гарантії безпеки продавця та покупця. Існує багато варіантів та умов оплати за товар в залежності від стану його готовності та етапу постачання, однак продавець та покупець завжди в тій чи іншій мірі несуть ризик втрати вантажу або коштів. Дана проблема вирішується шляхом механізму банківського акредитиву, коли банк-посередник виступає гарантом здійснення угоди. Класична схема акредитиву виглядає так: банк-гарант утримує кошти покупця як передплату та передає ці кошти або частину коштів поетапно продавцю за умови виконання ним певних умов, зазначених в контракті, наприклад замитнення товару, приймання на борт судна, висилання оригінальних документів, прибуття в порт призначення та інше. Однак дана класична схема має ряд значних недоліків та незручностей: банк, як фінансовий посередник, утримує значну комісію за свої послуги, дана операція сильно розтягнута в часі через пересилання паперових документів поштою, а також покупець у виді передплати за товар заморожує кошти на значний період часу. Згідно досліджень [58] щодня в світі в секторі міжнародної логістики заморожується 140 мільярдів доларів США через питання оплати, а також в середньому компанія змушена чекати 42 дні до моменту отримання оплати.

9) Затримки в строках постачання. Згідно спільних досліджень компаній IBM та Maersk [59] в середньому простий контейнер з температурним режимом

проходить через 30+ різних інстанцій та організацій, що вимагає 200+ комунікаційних зусиль. Будь-яка затримка на будь-якому етапі призводить до зриву строків поставок, а в деяких випадках навіть до втрати вантажу. Також раніше вже зазначалось про черги в доках портів для вивантаження суден.

10) Пошук клієнтів / вантажів або перевізників / транспорту. Дана проблема на даний час не є критичною та практично вирішена, тому що, як зазначалося вище, на базі технології блокчейн вже створено безліч платформ та служб, що допомагають клієнтам та перевізникам знайти один одного та контактувати напряму [60].

Для уникнення або мінімізації наслідків зазначених проблем вимагається повний та безперервний контроль за станом заданих параметрів на будь-якому відрізку ланцюга постачання, особливо на стадіях перевантаження між різними логістичними операторами, наприклад між сухопутним та морським транспортом або перевантаження контейнеру з судна в порт. На короткий час контейнер відключається від енергопостачання, тому надзвичайно важливо стежити за заданими параметрами в цей період та відстежувати можливі критичні відхилення від оптимального стану та тривалість такого періоду. Для прикладу, мінімальна температура зберігання м’яса або риби глибокої заморозки становить -18^{o}C [61], при підготовці до тривалого по часу перевантаженню доречно знизити температуру до -20-22^{o}C, що дає змогу утримувати температуру в оптимальному діапазоні протягом 15-20 годин за умови, що герметичний контейнер-рефрижиратор не буде відкриватися. Зазвичай цього часу достатньо, щоб перевантажити контейнер та підключити його до іншої мережі енергопостачання. Однак при мультимодальних міжнародних перевезеннях, коли використовується багато логістичних операторів та перевантажень (сухопутна доставка в країні імпорту та експорту, один або більше морських перевізників, портовий оператор в країні імпорту та експорту) вкрай важливо мати безперервний запис станів заданих параметрів якості товару з метою контролю та виявлення випадків порушень цих параметрів, а також чіткого визначення часу та місця таких порушень, та логістичних операторів, винних в цьому.

Технологія блокчейн та смарт-контрактів якщо не повністю вирішують зазначені тут проблеми, то значно мінімізують їх негативний вплив, особливо ефективно в поєднанні з технологіями GPS та програмованими датчиками, що розміщуються всередині контейнерів та контролюють визначені користувачем параметри.

1.7. Формальна верифікація в проектуванні логістичних систем

Системи міжнародних перевезень із часом стають складнішими і користувачу логістичних застосунків стає все важче відслідковувати та передбачати проблеми, пов’язані із перевезенням. Великі об’єми даних ускладнюють і автоматичну обробку можливих сценаріїв, оскільки їх кількість росте експоненційно при вирішуванні задач аналізу дотримування умов або властивостей ланцюжка перевезення.

Дану проблему можна вирішувати впровадженням формальних алгебраїчних методів з метою аналізу та дослідження властивостей перевезень. З цією метою в сучасній методології існують такі напрями, як перевірка моделей (model checking), що базується на використанні методів моделювання та статичного доведення властивостей. Метод широко використовується для верифікації формальних моделей, в тому числі логістичних, що реалізують деякий ланцюг постачання. Такі інструменти, як MDIST [62], SPIN[63], MaceMC[64], TLC [65] показали свою практичність в застосуванні до певних індустріальних областей.

Певна складність існує в формалізації моделей, тому що найбільш розвинуті та розширені формальні методи верифікації використовують складну математичну нотацію, наприклад системи Coq [66] та TLA.

При аналізі безпеки перевезень в моделях також проводиться формальне доведення властивостей безпеки, при чому використовуються логіки вищих порядків та автоматична система доведення теорем [67].

Логістичні моделі відносяться до розподілених систем, в яких бере участь деяка кількість агентів, що взаємодіють між собою. У верифікації та тестуванні

розподілених систем виникає явище експоненційного вибуху, спричиненого паралельною активністю агентів, що завдає труднощі в аналізі сценаріїв логістики.

Логістичні моделі відносяться також до класу систем, що критичні до безпеки, тому аналіз та доведення їх властивостей стає одною із основних задач, що має вирішуватись ще на ранніх стадіях розробки моделей.

Для оцінки безпеки в логістиці використовуються так звані блок діаграми надійності (Reliability Block Diagrams) [68], які представляють ланцюг постачання. Здебільшого безпека перевезень аналізується за допомогою моделювання.

У відкритих системах такого рівня, масштабу та складності, як міжнародні логістичні системи, де задіяна така велика кількість агентів, очевидно, що існує велика кількість зловмисників та шахраїв, які зацікавлені втрутитись в процес створення, передачі, обміну інформації для досягнення своїх цілей, зокрема привласнення товарів та коштів, а також отримання певних переваг при наданні неправдивої інформації до державних контролюючих органів. Якщо зі зловмисниками нижчих рівнів, які використовують подання неправдивої інформації, доволі легко боротись шляхом посилення надійності обміну інформацією в розподілених системах, то зі зловмисниками вищого рівня, які застосовують кібератаки на розподілені системи, потрібно приділяти особливу увагу. Тому при формальній верифікації міжнародних логістичних систем обов’язково проводиться перевірка властивостей стійкості та протидії цих систем кібератакам зловмисників.

1.8. Проектування логістичних систем та модельний метод обробки

При проектування логістичних систем виникає необхідність в дотриманні властивостей безпеки та надійності протидії шахрайству та атакам зловмисників. Більшість систем розробляється в межах усталених процесів розробки програмних систем. Одним із таких процесів для високонадійних та критичних до безпеки систем є життєві цикли моделі CMM (Capability Maturity Model). Існують декілька різновидів життєвих циклів в моделі CMM: водоспадна модель, V-модель, спіральна та інші моделі. Основними є водоспадна та V-модель, які включають в

усі етапи життєвого циклу розробки – збір вимог, етап дизайну, етап кодування та етап тестування. Світові стандарти в автомобільній, медичній, авіа-космічній індустріях потребують дотримання певних умов на циклах розробки, в тому числі використання формальних методів. Необхідними умовами є верифікація та тестування як коду, так і властивостей безпеки. Деякі стандарти включають пошук вразливостей у коді.

При розгляданні децентралізованої моделі логістики складність перевірки властивостей безпеки зростає, тому моделювання є недостатнім, і для верифікації моделі використовують алгебраїчний підхід. Особлива складність виникає при проектуванні розподілених систем, зокрема створеної на основі блокчейну, де паралельно функціонує велика кількість агентів.

Модельний спосіб розробки вимагає використання моделей на кожному етапі розробки. Формальні методи використовуються на всіх етапах розробки в процедурах верифікації та тестування.

1.9. Модельний спосіб розробки надійних систем з підвищеними вимогами до безпеки

Під «надійністю систем» мається на увазі здатність зберігати значення їх параметрів у визначеному діапазоні для функціонування системи в заданих умовах. Поняття «безпеки системи» визначається сукупністю станів системи, в яких зовнішні та внутрішні фактори не призводять до втрати повного або часткового функціонування. Завдяки специфіці свого використання деякі системи вимагають підвищення вимог до надійності та безпеки, так як збій в роботі таких систем (ядерна енергетика, авіація, медицина) призводять до людських смертей, катастроф, екологічного забруднення чи втрати унікального обладнання. Процес розробки таких систем базується на зменшенні ризику або повному виключенні загрози неспрацювання умов безпеки в роботі системи. Розробка таких систем включає процедури верифікації та валідації. Під верифікацією розуміємо процес перевірки властивостей та вимог, що були визначені у початковій фазі розробки

системи. Під валідацією розуміємо відповідність продукту потребам користувачів та замовника.

Високі вимоги до процесу розробки програм визначають необхідність використання формальних методів на кожному етапі процесу розробки системи. Процесу верифікації підлягає артефакт кожної етапу процесу розробки від визначення вимог до тестування на завершальному етапі. Процедуру верифікації також доцільно застосовувати на фазі інтеграції готового продукту. Очевидно, що більш ефективною для виявлення порушення умов безпеки є верифікація на ранніх стадіях процесу розробки.

При розробці високонадійних систем, критичних до безпеки, є доцільним використання модельного способу розробки, тобто результатом кожного етапу процесу є модель, відображена у відповідних формальних специфікаціях. Виявлення можливості порушення може бути здійснене на всіх етапах при використанні модельного способу розробки.

1.10. Етапи процесу модельної розробки систем з підвищеними вимогами до безпеки.

У стандартному процесі розробки програмного забезпечення визначено кілька етапів, від попередніх переговорів із замовником та збір вимог до кінцевого продукту від нього, до тестування системи та інтеграції продукту до іншої системи.

Процес розробки розпочинається від етапу визначення вимог до майбутнього програмного забезпечення у вигляді вільного письмового опису або формальних та напівформальних специфікацій. При цьому вимоги мають бути несуперечливі та повністю визначати функціонування системи. Другий етап – дизайн продукту, де визначається структура продукту та інтерфейси між частинами системи для злагодженої роботи різних груп розробників. На етапах кодування та тестування створюється код та визначається тестовий набір у результаті взаємодії команди розробників з командою тестування.

У розробці систем з підвищеними вимогами до безпеки процес перевірки безпеки інтегрується у стандартний процес у вигляді верифікації та тестування на

кожному етапі розробки. При користуванні модельним методом на кожному етапі розробки створюються артефакти, що є об’єктом верифікації. Формальна модель вимог представлена у відповідній формальній мові, фаза дизайну проекту визначається формальною моделлю. Етап кодування завершується створенням початкового коду, який є вхідним матеріалом для команди тестування.

Верифікація формальних вимог та моделі дизайну є невід’ємною частиною інтегрованого процесу розробки систем з підвищеними вимогами до безпеки. Існує два основних види верифікації: перший – це досяжність властивостей, другий – тестування розробником формальних специфікацій відповідними тестами. Останнє розглядається як спеціальний етап після кодування та верифікації початкового коду.

До властивостей, що перевіряються під час верифікації, відносимо:

– Стандартні властивості несуперечливості вимог та повноти вимог (відсутність тупикових станів та взаємного блокування);

– Властивості безпеки, що визначені замовником та специфічні для проектованого класу систем;

– Властивості надійності та життєздатності;

– Локальні властивості (інваріанти, контракти) та властивості, що визначають відповідність вимогам;

– Властивості, що специфічні для певних класів систем, що проектуються – конкуруючі процеси, взаємне виключення, когерентність станів, динамічне взаємне блокування.

1.11. Методи верифікації на етапі визначення вимог та дизайну проекту.

Визначення вимог – початковий етап розробки програми, пристрою або системи. Це можуть бути вимоги до програмного забезпечення, обладнання, мікропроцесору, до мережевого або телекомунікаційному протоколу та інше. Визначення вимог – це більше, ніж опис побажань замовника, наприклад, узгодження тих аспектів, на основі яких приймається або тестується продукт.

Ми розглядаємо інформаційні системи, тобто системи обробки інформації. Це можуть бути пристрої або програми, системи реального часу або багатоагентні розподілені системи. Увага приділяється дослідженню властивостей поведінки систем і взаємодії агентів з середовищем та іншими агентами.

Замовник формулює вимоги у своєму баченні та термінології, часто це довільний словесний опис натуральною мовою. Робить він це на рівні абстракції, відповідної його розумінню. При цьому такі вимоги, що описують поведінку майбутнього продукту, представлені природною мовою, можуть складати великі обсяги текстових документів та ілюстровані за допомогою великої кількості таблиць, графіків, блок-схем. Виконавець може бути присутнім при формування вимог замовником та відразу перетворювати їх в частково формалізований вигляд. Чим ближче представник замовника та виконавця до технічних фахівців, тим детальніше вони можуть сформулювати вимоги, іншими словами - на більш низькому рівні абстракції.

Під дизайном проекту у процесі розробки програмного забезпечення мається на увазі створення специфікацій високого рівня абстракцій. Це являє собою документ, у якому визначені частини системи, їх взаємодія та інтерфейси між ними. Може бути також представлений детальний дизайн кожної або деяких частин системи. В якості дизайну може розглядатися модель, яка була створена вручну згідно специфікацій вимог або створена генерацією з формальної моделі вимог при подальшому ручному регулюванні та модифікуванні. Тестування на етапі дизайну вирішує питання відповідності моделі дизайну вимогам, тобто, модель може перевірятися за допомогою тестів, що походять з вимог.

1.12. Метод модельного тестування

Метод модельного тестування (Model based testing) полягає в автоматизації створення наборів для тестування відповідно до деяких критеріїв, визначених користувачем, такими як покриття, метод тестування, набір властивостей, що тестуються, включно з властивостями безпеки. В MBT-методі тестові набори створюються автоматично при використанні моделей, які описують поводження

системи, що тестується, на відміну від традиційного способу, де кожен тест повинен створюватися вручну.

Ефективність та результативність модельного тестування полягає в повному чи частковому вирішенні таких проблем:

1. У процесі розробки програмного забезпечення тестування посідає значну частину, що досягає 40% працезатрат;
2. Тестування є одним з найбільш працезатратних процесів, у якому присутня суттєва доля процесу ручного створення тестів;
3. Ручне тестування стає неефективними через розвиток складних систем та їх все більшого ускладнення, що призводить до збільшення кількості коду.

Тестування процесу розробки програмного забезпечення є стандартизованою діяльністю, що складається з етапів планування й документування, створення та виконання тестів. Залежно від задач тестування розглядаються різні види тестування – загальносистемне, інтеграційне, регресивне, а також багато інших з вибором відповідних технік, відомих, як техніка «чорної скриньки», коли код, що тестується, є недосяжним та використовується лише інтерфейс коду, і техніка «білої скриньки», коли код є відкритим і може бути використаний при тестуванні.

Початковим етапом MBT-методу є створення тестової моделі. Вхідним матеріалом для цього є початковий набір вимог. У множині вимог виокремлюються поведінкові вимоги, що містять інформацію про можливі сценарії роботи системи. В них зафіксовано запити до системи та очікувані відповіді системи, засновані на алгоритмах, які необхідно реалізувати.

MBT-тестування в основному застосовується для розробки інтерактивних програм, які взаємодіють із зовнішнім середовищем у реальному часі, та основний режим роботи яких – це запит-відповідь або стимул-реакція. Інтерактивні програми мають нетривіальні поведінкові моделі, на відміну від обчислювальних програм, які за початковими даними обчислюють результат і виводять його наприкінці роботи. З іншого боку, ми можемо спостерігати внутрішню поведінку обчислювальної програми (розгалуження і цикли). Для цього ми перетворюємо її

на інтерактивну шляхом включення додаткових операторів взаємодії з навколишнім середовищем.

До інтерактивних програм належать також системи взаємодіючих програм. У моделях цих програм мають бути засоби організації паралельних обчислень. Кількість сценаріїв для паралельних процесів може бути як завгодно великою, тому MBT-метод тут є незамінним.

Створення тестової моделі являє собою формалізацію інформації, представленої в поведінкових вимогах у термінах мови, що є вхідною для обраної системи генерування тестів. Маючи тестову модель, сумісну з вхідною мовою, для генерації тестів потрібно отримати набір сценаріїв поводження системи, що відповідають вихідним вимогам. Отримані сценарії в подальшому використовуються для завершального генерування тестів, вид яких обумовлений вже наступним інструментом, який виконує тести, тестуючи готову систему або її більш детальну модель. Генеровані сценарії містять інформацію про дані, що сприймаються на вході системи, що тестується, та її відповіді. Ці дані входять у тест і використовуються системою виконання тестів для аналізу відповідей системи у порівнянні з очікуваними відповідями.

Схема MBT може деталізуватися в залежності від методів і інструментів генерування та виконання тестів. Так, можливе генерування абстрактних сценаріїв, що покривають не конкретні значення, а множини значень. Виконання ж тестового набору є, як правило, виконанням тестів з конкретними значеннями. Тоді виникає необхідна процедура конкретизації сценаріїв із абстрактних тестів. При цьому розглядаються різні методики розгляду конкретних значень, наприклад, розглядаються граничні значення або довільні значення з деякої множини. При цьому конкретизації піддаються як вхідні сигнали на систему, що тестується, так і на очікувані відповіді.

1.13. Інтеграція процедури верифікації властивостей безпеки у стандартний процес модельної розробки програмного забезпечення.

Процес верифікації повинен бути повністю інтегрованим в процес розробки програмних систем з підвищеними вимогами до безпеки. Процес верифікації на всіх етапах розробки відбувається наступним чином (Рис.1.1.).

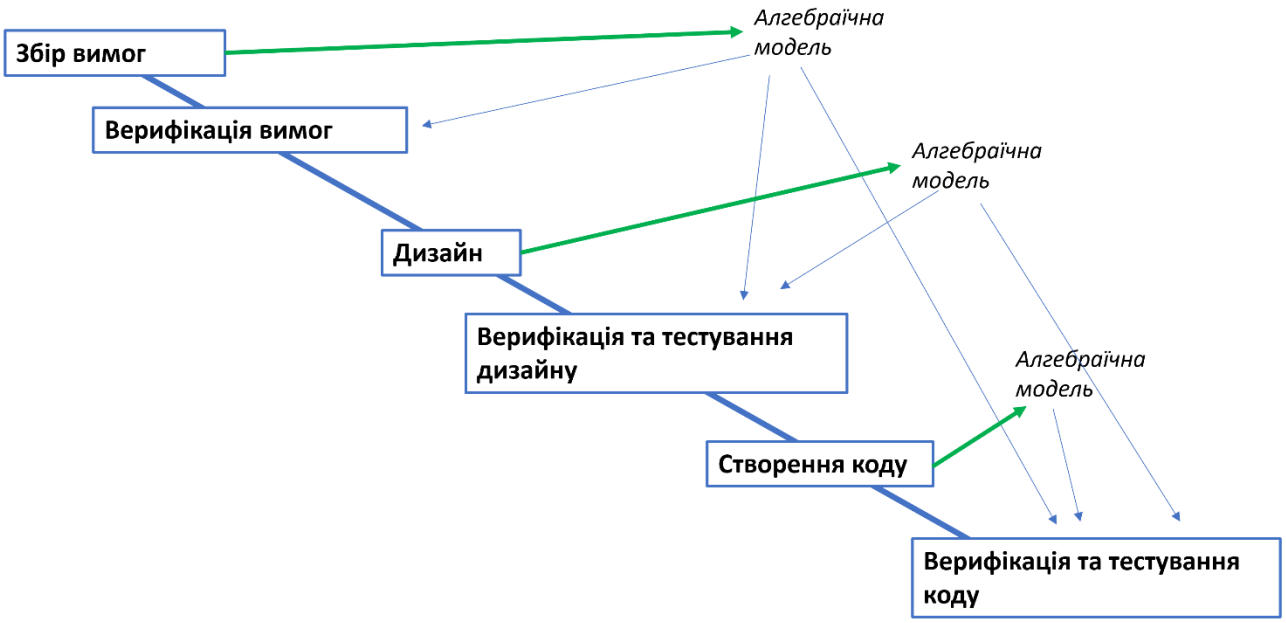


Рис.1.1. Процес верифікації в модельному способі розробки.

Етап визначення вимог. Створюється формальна модель вимог, яка є входом для верифікаційної процедури даного етапу, результатом якої є верифікаційний вердикт з представленням можливих сценаріїв, що ведуть до знайдених порушень. При наявності таких сценаріїв формальна модель та початкові вимоги корегуються та процес верифікації повторюється. Формальна модель вимог також є джерелом множини сценаріїв, що можуть бути перетворені у тести для моделі дизайну або коду.

Також з цієї моделі може бути згенерована модель дизайну або її каркас для подальшої деталізації або навіть каркас коду, якщо моделювання дизайну пропускається.

Етап створення моделі дизайну проекту. Модель дизайну може бути створена вручну, або деталізована з каркасу, або згенерована повністю з вимог. В останньому випадку верифікація моделі може бути пропущена, оскільки виключена можливість привнесення помилки.

На цьому етапі модель може бути верифікована на необхідні властивості. Також модель може бути верифікована на відповідність вимогам на їх рівні абстракції. Це може бути здійснено двома шляхами.

Перший спосіб – це використати множину сценаріїв, які можуть бути згенеровані з формальної моделі вимог як множина тестів, та виконати їх на моделі дизайну.

Другий спосіб може мати місце при автоматичному формуванні каркасу моделі на рівні вимог. У цьому випадку формули, які визначають умови на множині вимог (зокрема перед- та післяумови), можуть бути перенесені на модель дизайну та верифіковані статичним методами.

Етап кодування. Початковий код може бути створений як повністю вручну, так і частково, як деталізований із згенерованого каркасу. Джерелом генерації є модель дизайну. Початковий код також може бути згенерований повністю.

Якщо і модель, і код створювалися вручну, то можливе тестування на двох рівнях абстракції – як на рівні вимог так і на рівні дизайну моделі, тобто перевірка коду на відповідність як до вимог, так і до дизайну моделі. Окремо може бути проведена процедура верифікації.

Тестування є заключною фазою верифікації програми, тому на цьому етапі міряється покриття початкового коду тестами, що визначає якість продукту.

Кожна фаза має циклічну структуру. На кожному етапі може бути проведено одну і більше ітерацій, під час яких виправляються порушення, знайдені при верифікації або тестуванні. При цьому згідно процесу кількість знайдених порушень має зменшуватись на кожній ітерації. У противному випадку можливе перепланування та переробка дизайну та повторення попередніх етапів. Тому важливим фактором є знаходження помилок та порушень на більш ранніх етапах, ніж на тестуванні та інтеграції.

Етап інтеграції. На цьому етапі проводиться системне тестування, яке доцільно проводити відповідно до тих атрибутів моделей, які впливають на перетин властивостей інтегрованих моделей. На цьому етапі можливо проведення як тестування, так і верифікації, відповідно до аспекту перетину властивостей.

На етапі тестування використовуються тести, згенеровані із моделей вимог та дизайну.

Висновки до Розділу 1

Отже, підсумовуючи все вищесказане, ми виділили глобальні проблеми, які накопичились в сучасній системі міжнародної логістики, та які можуть бути вирішені за допомогою застосування технології блокчейн та методів модельної розробки систем:

- Децентралізація та відмова від використання потужних стаціонарних серверів, які накопичують та зберігають величезні об’єми інформації.
- Безпека зберігання та обміну інформацією. Частково вирішується децентралізацією, мінімізуючи загрози пошкодження чи збою у роботі стаціонарних серверів шляхом винесення інформації в розподілені системи, такі як блокчейн, де інформація дублюється на багатьох носіях (нодах). Також безпека досягається шляхом кодування та шифрування інформації та можливістю доступу до неї тільки тим особам, які мають до неї безпосереднє відношення. Ще однією великою перевагою є неможливість фізичного втручання людини до попередньо внесеної інформації та зміни нею цієї інформації зі зловмисних чи інших мотивів, так як технологія блокчейн аналізує та валідує будь-які зміни інформації, що вносяться до розподіленої бази даних.
- Доступність – клієнт з мінімальним рівнем технічних знань має можливість здійснювати необхідні йому операції за допомогою комп’ютера чи мобільного телефону за допомогою спеціальних додатків та програм.
- Швидкодія – розподілені системи прискорюють реакцію робочої системи на запит оператора, ніж стаціонарний сервер, який в часи максимального

навантаження може досить довго обробляти інформацію та відповідати на запит оператора.

- Економія часу та людських ресурсів – розподілені системи обробляють інформацію та відповідно реагують на неї в тисячі швидше, ніж людина, що дозволяє економити час та людські ресурси.
- Усунення людського фактору. Більшість затримок в роботі системи та, як наслідок, втрат часу та ресурсів відбувається через людський фактор (неправильне трактування інформації, пізня реакція на збій, ігнорування сигналів системи, низька кваліфікація, природня фізична обмеженість швидкості реакції людини та аналіз інформації, тощо), тому доцільно намагатися максимально усунути залежність системи від людського фактору.

Розглянувши основні проблеми в ланцюжку постачання та контролю якості товарів у міжнародній логістиці, ми бачимо величезний потенціал розвитку та застосування технології блокчейн та смарт-контрактів на її основі, які вирішують або значно спрощують ці проблеми:

- Можливість постійного та безперервного контролю дотримання заданих параметрів, наприклад температурного режиму та вологості, відслідковування порушень та чітке визначення часу, місця та етапу такого порушення.
- Можливість відслідковування повного шляху товару в ланцюжку постачання, тобто від дверей виробника до дверей покупця.
- Прискорення документообігу, спрощення порядку створення та верифікації документів, підвищення рівня надійності та підтвердження оригінальності документів та інформації.
- Відслідковування великої кількості об’єктів логістики в реальному часу, корегування оптимального маршруту, пошук найближчого та найбільш відповідного характеристикам вантажу транспортного засобу, можливість отримання незалежної правдивої інформації про дії логістичних операторів під час колізій, аварій та інших небажаних випадків.

- Відслідковування, виявлення та протидія випадкам шахрайства, агентам зі злочинними намірами, які навмисно подають неповну, спотворену або змінену інформацію з метою отримання персональної вигоди або державних пільг.

- Протидія кібератакам, або зловмисникам вищого рівня, які використовують складні методи атак та впливу на саму систему захисту інформації, що базується на технології розподілених систем.

Зростання надійності та безпеки міжнародних торгових операцій, строків постачань та оплати можливо досягти через використання смарт-контрактів, прискорення розрахунків та виконання умов контрактів.

Системи міжнародних перевезень та логістики з часом стають складнішими і та містять в собі все білішу кількість активних агентів, тому цілісний процес контролювати стає дедалі важче. Дану проблему можна вирішувати впровадженням формальних алгебраїчних методів з метою аналізу та дослідження властивостей перевезень. З цією метою в сучасній методології існують такі напрями, як перевірка моделей, що базується на використанні методів моделювання та статичного доведення властивостей. При аналізі безпеки перевезень в моделях також проводиться формальне доведення властивостей безпеки, при чому використовуються логіки вищих порядків та автоматична система доведення теорем. Логістичні моделі відносяться також до класу систем, що критичні до безпеки, тому аналіз та доведення їх властивостей стає одною із основних задач, що має вирішуватись ще на ранніх стадіях розробки моделей, а також при їх проектуванні виникає необхідність в дотриманні властивостей безпеки та надійності протидії шахрайству та атакам зловмисників. При розробці необхідними умовами є верифікація та тестування як коду, так і властивостей безпеки. При розгляданні децентралізованої моделі логістики складність перевірки властивостей безпеки зростає, тому моделювання є недостатнім, і для верифікації моделі використовують алгебраїчний підхід. Модельний спосіб розробки вимагає використання моделей на кожному етапі розробки. Формальні методи використовуються на всіх етапах розробки в процедурах верифікації та тестування. Метод перевірки моделей (MBT) передбачає створення тестової моделі на мові, що

є вхідною для обраної системи генерування тестів, а далі отримується набір сценаріїв поводження системи, що відповідають вихідним вимогам. Ці сценаріх використовуються при тестуванні готової системи або більш делатізовану модель цієї системи. Також в процес розробки програмних систем з підвищеними вимогами до безпеки повинен бути повністю інтегрованим процес верифікації властивостей безпеки.

Список використаних джерел до Розділу 1.

1. Економічна енциклопедія: У трьох томах. Т. 2. / Редкол.: …С. В. Мочерний (відп. ред.) та ін. – К.: Видавничий центр “Академія”, 2000. – 864с.

2. Крикавський Є.В., Чорнописька Н.В. Логістичні системи: Навч. посібник. – Львів: Видавництво Національного університету ”Львівська політехніка”, 2009. – 264 с.

3. Крикавський Є.В. Логістичне управління: Підручник. – Львів: Видавництво Національного університету ”Львівська політехніка”,2005. – 684 с.

4. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)

5. Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V.: Blockchain technology: beyond bitcoin. Appl. Innov. 2(6), 6–10 (2016)

6. McPhee, C., Ljutic, A.: Blockchain. Technol. Innov. Manag. Rev. 7(10), 3–5 (2017)

7. Electricity journal homepage: blockchain technology: will it make a difference? Electr. J. 30 (3), 86–87 (2017)

8. Finextra: Marine Transport International Applies Blockchain to Shipping Supply Chain (2016). URL: https://www.finextra.com/pressarticle/66223/marine-transportinternational-applies-blockchain-to-shipping-supply-chain

9. Solesvik, M.Z.: Interfirm collaboration in the shipbuilding industry: the shipbuilding cycle perspective. Int. J. Bus. Syst. Res. 5(4), 388–405 (2011)

10. Official website of International Maritime Organization –IMO. URL: http://www.imo.org

11. Den Norske Veritas—DNV GL: Making your Asset Smarter with the Digital Twin. URL: https://www.dnvgl.com/article/making-your-asset-smarter-with-the-digital-twin-63328

12. The Economist. The data deluge, 25 Feb 2010. URL: http://www.economist.com/node/15579717

13. Tien, J.M.: Internet of connected servgoods: considerations, consequences and concerns. J. Syst. Sci. Syst. Eng. 24(2), 130–167 (2015)

14. Lacey, M., Lisachuk, H., Giannopoulos, A., Ogura, A.: Shipping smarter: IoT opportunities in transport and logistics. The Internet of Things in Shipping. Dupress-Deloitte (2015)

15. Offshore Energy Today: Meet Onboard, the Maritime Internet of Things. URL: http://www.offshoreenergytoday.com/oeec-meet-onboard-the-maritime-internet-of-things/?utm_source=emark&utm_medium=email&utm_campaign=daily-update-offshore-energy-today-2017-10-02&uid=207087

16. Maritime: Six Maritime Start-Ups That Are Changing the GAME. URL: https://knect365.com/talentandtraining/article/1149354e-68d9-4e74-9f91-a900ac869526/6-maritime-startupsthat-are-changing-the-game

17. Den Norske Veritas—DNV GL: Certificates in the blockchain. URL: https://www.dnvgl.com/assurance/certificates-in-the-blockchain.html

18. Den Norske Veritas—DNV GL: Blockchains in the shipping world. URL: https://www.dnvgl.com/expert-story/maritime-impact/Blockchains-in-the-shipping-world.html

19. Splash 24: Dr. Martin Stopford on the Future of Shipping. URL: http://splash247.com/dr-martinstopford-future-shipping

20. Kondratenko, Y.P., Kozlov, O.V., Korobko, O.V., Topalov, A.M.: Internet of things approach for automation of the complex industrial systems. In: ICTERI-2017, CEUR Workshop Proceedings Open Access, vol. 1844, pp. 3–18 (2017). URL: https://pdfs.semanticscholar.org/3ff6/4e4a07be1e8c2f0b16f4736397be1405218a.pdf

21. Tachizawa, E.M., Alvarez-Gil, M.J., Montes-Sancho, M.J.: How “smart cities” will change supply chain management. Supply Chain Manag. Int. J. 20(3), 237–248 (2015)

22. Hahn, G.J., Packowski, J.: A perspective on applications of in-memory analytics in supply chain management. Decis. Support Syst. 76(1), 45–52 (2015)

23. Maersk and IBM Introduce TradeLens Blockchain Shipping Solution. Aug. 9, 2018. URL: https://newsroom.ibm.com/2018-08-09-Maersk-and-IBM-Introduce-TradeLens-Blockchain-Shipping-Solution

24. TradeLens – official website. URL: https://www.tradelens.com

25. Forbes. IBM-Maersk Blockchain Platform Adds 92 Clients As Part Of Global Launch. URL: https://www.forbes.com/sites/michaeldelcastillo/2018/08/09/ibm-maersk-blockchain-platform-adds-92-clients-as-part-of-global-launch-1/#c469d5068a4a

26. Coindesk. IBM and Maersk Struggle to Sign Partners to Shipping Blockchain. Oct 26, 2018. URL: https://www.coindesk.com/ibm-blockchain-maersk-shipping-struggling

27. Computerworld. Maersk adds two big shipping firms to its blockchain ledger. 29 травня 2019р. URL: https://www.computerworld.com/article/3398923/maersk-adds-two-big-shipping-firms-to-its-blockchain-ledger.html

28. TC. IBM-Maersk blockchain shipping consortium expands to include other major shipping companies. URL: https://techcrunch.com/2019/05/28/ibm-maersk-blockchain-shipping-consortium-expands-to-include-other-major-shipping-companies/

29. TradeLens official brochure. Solution brief. Edition two.

30. SupplyChain Dive. 9 ocean carriers, terminal operators join new blockchain initiative to rival TradeLens. Nov. 7, 2018. URL: https://www.supplychaindive.com/news/ocean-carriers-new-blockchain-cosco-cma-cgm/541630/

31. CargoSmart. Global Shipping Business Network. URL: https://www.cargosmart.ai/en/

32. Globe news wire. Top Ocean Carriers and Terminal Operators Initiate Blockchain Consortium. November 06, 2018. URL:

https://www.globenewswire.com/news-release/2018/11/06/1646014/0/en/Top-Ocean-Carriers-and-Terminal-Operators-Initiate-Blockchain-Consortium.html

33. Supply Chain Dive. Maersk blockchain solution TradeLens adds ZIM. April 22, 2019. URL: https://www.supplychaindive.com/news/maersk-blockchain-solution-tradelens-adds-zim/553146/

34. Ball, B.: Reducing Global Logistics Cost with Benchmarking and Shipping Container Pricing Transparency. Aberdeen Group (2016)

35. Dhanji, T.: Blockchain—Where Oil and Gas Traders Dare to Trade. Ernest Young Publications (2017). URL: https://www.linkedin.com/pulse/blockchain-where-oilgas-traders-dare-tread-talib-dhanji

36. Freight Waves.: Swiss firm brings blockchain to the biopharmaceutical cold chain. 02/23/2018. URL: https://www.freightwaves.com/news/blockchain/skycellblockchaincoldchain

37. John G. Smith.: Block by Block: How blockchain will transform trucking. 18.01.2018. URL: https://www.todaystrucking.com/block-block-blockchain-will-transform-trucking/

38. Ana Alexandre.: New Blockchain-Based Supply Chain System Is Presented by Microsoft and Adents. URL: https://cointelegraph.com/news/new-blockchain-based-supply-chain-system-is-presented-by-microsoft-and-ardents

39. Suku. The future of supply chain is here. URL: https://www.suku.world/

40. DHL Trend Research. URL: https://www.logistics.dhl/content/dam/dhl/global/core/documents/pdf/glo-core-blockchain-trend-report.pdf

41. OriginTrail. URL: https://origintrail.io

42. Shipchain. THE END-TO-END LOGISTICS PLATFORM OF THE FUTURE: TRUSTLESS, TRANSPARENT TRACKING. URL: https://shipchain.io/

43. Nick Szabo - Smart Contracts: Building Blocks for Digital Markets. URL: www.fon.hum.uva.nl

44. Nick Szabo. Formalizing and Securing Relationships on Public Networks.

45. Welcome to ERights.Org. URL: erights.org

46. A Formal Language for Analyzing Contracts. URL: vwh.net

47. Gideon Greenspan.: Smart contracts make slow blockchains. URL: https://www.multichain.com/blog/2015/11/smart-contracts-slow-blockchains

48. StackEdit Viewer / Benoit Schweblin. URL: https://stackedit.io/viewer#!url=https://gist.githubusercontent.com/gavofyork/31b35cd2252a00d0d057/raw/16de06189d2175d2e31b300f1f8531e20c927635/solidity-original

49. Solidity — Solidity 0.2.0 documentation. URL: https://solidity.readthedocs.io/en/latest/

50. Zigurat. Blockchain Success cases: Supply Chain and Logistics. URL: https://www.e-zigurat.com/innovation-school/blog/blockchain-success-cases/

51. Robert Hackett.: Walmart and 9 Food Giants Team Up on IBM Blockchain Plans. URL: https://fortune.com/2017/08/22/walmart-blockchain-ibm-food-nestle-unilever-tyson-dole/

52. Карта випадків африканської чуми свиней в Україні та зони карантину. URL: http://www.asf.vet.ua/index.php/asfinukraine

53. Державна служба України з питань безпечності харчових продуктів та захисту споживачів: Африканська чума свиней? URL: http://www.asf.vet.ua/index.php/purpose-project/about-asf/124-african-swine-fever

54. Укрінформ. Нашестя метеликів – розплата за людську необачність. 03.06.2019 URL: https://www.ukrinform.ua/rubric-economy/2713807-nasesta-metelikiv-rozplata-za-ludsku-nenazerlivist.html

55. Petropavlivka.City. Миколаївку атакує гусінь. 05.06.2019 URL: https://petropavlivka.city/read/experiance/33845/mikolaivku-atakue-gusin

56. Carnet Maritime. Collision CMA CGM Verlaine et Odessa Star. 04.04.2010 URL: http://carnet-maritime.com/accidents-naufrages/collision-cma-cgm-verlaine-et-odessa-star.html

57. Информационный портал «Транспортный бизнес Украины». К ситуации столкновения контейнеровоза Verlaine и теплохода Odessa Star 4 апреля в Мраморном море - Международная юридическая служба. URL: http://tbu.com.ua/news/k_situatsii_stolknoveniia_konteinerovoza_verlaine_i_teplohoda

_odessa_star_4_aprelia_v_mramornom_more___mejdunarodnaia_uridicheskaia_slujba_foto.html

58. Cristina Commendatore.: Blockchain in trucking: What about the middlemen? 20.10.2017 URL: https://www.fleetowner.com/electronic-security/blockchain-trucking-what-about-middlemen

59. Robert Hackett.: IBM and Maersk Are Creating a New Blockchain Company. 16.01.2017 URL: https://fortune.com/2018/01/16/ibm-blockchain-maersk-company/

60. Winnesota.: HOW BLOCKCHAIN IS REVOLUTIONIZING THE WORLD OF TRANSPORTATION AND LOGISTICS. URL: https://www.winnesota.com/blockchain

61. Temperature Measurement in the Fish Industry. URL: http://www.fao.org/3/x5992e/X5992e01.htm

62. Yang, J., Chen, T., Wu, M., Xu, Z., Liu, X., Lin, H., Yang, M., Long, F., Zhang, L., Zhou, L. 2009. MODIST: transparent model checking of unmodified distributed systems. URL: https://www.usenix.org/legacy/events/nsdi09/tech/full_papers/yang/yang_html/index.html

63. Spin. URL: http://spinroot.com/spin/whatispin.html

64. Killian, C., Anderson, J. W., Jhala, R., Vahdat, A. 2006. Life, death, and the critical transition: finding liveness bugs in system code. URL: http://www.macesystems.org/papers/MaceMC_TR.pdf

65. Lamport, L., Yu, Y. 2011. TLC—the TLA+ model checker. URL: http://research.microsoft.com/en-us/um/people/lamport/tla/tlc.html

66. Wilcox, J. R., Woos, D., Panchekha, P., Tatlock, Z., Wang, X., Ernst, M. D., Anderson, T. 2015. Verdi: a framework for implementing and formally verifying distributed systems. Proceedings of the ACM SIGPLAN 2015 Conference on Programming Language Design and Implementation: 357-368. URL: https://homes.cs.washington.edu/~mernst/pubs/verify-distsystem-pldi2015-abstract.html

67. Towards Formal Reliability Analysis of Logistics Service Supply Chains using Theorem Proving Waqar Ahmed , Osman Hasan , and Sofi`ene Tahar EPiC Series in

Computing Volume 40, 2016, Pages 1–14 IWIL-2015. 11th International Workshop on the Implementation of Logics

68. Y. Li and H. Yi. Research on the Inherent Reliability and the Operational Reliability of the Supply Chain. u- and e-Service, Science and Technology, 7(1):104–112, 2014.

69. Щербина В.В. Проблеми та завдання розвитку портової логістики України // Розвиток методів управління та господарювання на транспорті: Зб. наук. праць, 2019. № 2 (67). С. 89-101. DOI: 10.31375/2226-1915- 2019-2-89-101.

70. Анісімов А.В., Заславський В.А., Фаль О.М. Основи інформаційної безпеки та захисту інформації у контексті євроатлантичної інтеграції України. // ДП «НВЦ» Євроатлантикінформ Київ, 2006. – 103 с.

71. ТЕХНОЛОГИЯ BLOCKCHAIN В ЛОГИСТИКЕ. 07.07.2017. URL: https://logist.fm/publications/tehnologiya-blockchain-v-logistike

72. Українські морські порти впровадять блокчейн-рішення на базі системи e-Port. 18 СЕРПНЯ 2018. URL: https://mind.ua/news/20187830-ukrayinski-morski-porti-vprovadyat-blokchejn-rishennya-na-bazi-sistemi-e-port

РОЗДІЛ 2.
АЛГЕБРАЇЧНИЙ ПІДХІД У ФОРМАЛІЗАЦІЇ, ВЕРИФІКАЦІЇ ТА МОДЕЛЬНОМУ ТЕСТУВАННІ В СИСТЕМАХ МІЖНАРОДНОЇ ТА МІСЦЕВОЇ ОБСЛУГОВУЮЧОЇ ЛОГІСТИКИ

Стабільність функціонування, стійкість до зовнішніх загроз та виявлення та усунення вразливостей є необхідними властивостями комплексних систем міжнародної та внутрішньої обслуговуючої логістики. Такі системи програмного забезпечення є критично важливими до безпеки та вимагають розробок з модельно-орієнтованим підходом для надійності. Модельний підхід до розробки передбачає створення моделей як інструментів на кожному етапі розробки програмного забезпечення для застосування методів верифікації, тестування та валідації.

Методи математичного моделювання з такими інструментами, як алгебра поведінок, дають можливість керувати логістичними системами та забезпечувати їх безпеку та захищеність за допомогою формальної верифікації та формалізації за допомогою тестування на основі моделей. Алгебраїчні моделі логістичних систем можуть бути використані для аналізу поведінки всіх залучених агентів і доведення їх здатності виконувати свої цілі та здатності всієї системи постійно існувати та залишатися стабільною.

Формальна верифікація використовується в аналізі та верифікації бізнес-логістики вже декілька десятиріч. Формальні мови такі як BPMN [1], UML [2], SysML[3] використовуються для опису логістичних бізнес-процесів, а досить велика кількість методів такі як VDM[4], SPIN [5] та інші застосовується для перевірки властивостей безпеки. Ускладнення специфіки та збільшення інформаційного навантаження обумовило використання більш новітні методики та відповідні математичних теорії.

У цьому дослідженні продемонстровано, як алгебраїчний підхід, оснований на алгебрі поведінок, може бути ефективно використаний для перевірки властивостей безпеки складних систем. В якості прикладів ми застосовуємо її до реальної закритої логістичної системи місцевого обслуговування (фермерське

господарство) та відкритої міжнародної логістичної системи (експортні поставки виробництва вищезгаданого фермерського домогосподарства).

2.1. Алгебра поведінок, розвиток та сфери застосування

На початку 1970-х, Віктор Михайлович Глушков, директор Інституту кібернетики Національної академії наук України, заснував комп'ютерну наукову школу, яка сфокусувалась на алгебраїчних методах. Тематика дослідження варіювалась від автоматизованих доведень теорем до алгебраїчного моделювання та від наукових концепцій до промислових прикладних програм. Пізніше в 2000-х Система Алгебраїчного Програмування (APS) [1] як складова частина цієї програми досліджень розширилась за допомогою включення в себе концепції взаємодії перехідних систем у деяких алгебраїчних середовищах. Ключовою ідеєю є інсерційне функціонування, яке визначає поведінку взаємодії перехідних систем в алгебраїчному середовищі. Ця концепція розроблена в контексті системи інсерційного моделювання (IMS) [2]. Алгебра поведінок та основні специфікації протоколу використовуються для формального опису моделі. Такі алгебраїчні інструменти, як теореми та абстрактні алгоритми, служать основою для формальних методів верифікації. Алгебраїчний підхід та інсерційне моделювання дозволяють довести або спростувати властивості, подавши контр-приклади для системи з довільною кількістю агентів. Використовуючи формальну модель програми на якомусь рівні абстракції, ми можемо генерувати різні сценарії поведінки агентів або груп агентів. Ці сценарії є символьними і можуть бути проілюстровані конкретними прикладами за допомогою формальних методів. Генерація символічних сценаріїв забезпечує гарне висвітлення поведінки, а отже, отримані конкретні сценарії можуть розглядатися як тестовий набір для створеного програмного забезпечення.

Інсерційне моделювання фокусується на моделях побудови та вивченні взаємодії агентів та середовищ у складних розподілених мультиагентних системах. Загальні концепції інерційного моделювання – це ієрархія середовищ та агентів, занурених у ці середовища, взаємодія цих агентів, середовищ та середовищ вищого

рівня, двосторонній вплив агентів та середовищ один на одного, зміна поведінки набору агентів під час занурення в нові середовища. Середовище - це агент, який має функцію занурення. Агенти розглядаються як системи переходу атрибутів. У таких системах стани визначаються значеннями атрибутів. Агенти мають набір атрибутів, які визначають тип агента. Атрибути середовища пов'язані з глобальними атрибутами, які відомі всім агентам.

В 1997 Гілберт та Летичевський ввели поняття алгебри поведінок [3] як інструмент та невід'ємна частина інсерційного моделювання. Алгебра поведінок - це універсальна алгебра двох видів. Основний вид - це набір поведінок, а другий - це сукупність дій. Ця алгебра має дві операції, три термінальні константи та відношення наближення. Операції позначаються префіксами a.u (де «a» є дією, а «u» є поведінкою) та недетермінований вибір поведінки u + v. Кінцевими константами є Δ (успішно визначене), 0 (тупіковий стан) та ⊥ (невизначена поведінка). Відношення⊑ є частковим порядком в наборі поведінок з мінімальним елементом ⊥. Алгебра поведінок також доповнена двома операціями: паралельними (||) та послідовними (;) композиціями поведінок.

Одним з прикладів вираження поведінок є:

B0 = a1.a2.B1 + a3.B2,

B1 = a4, B2 =…

Мається на увазі, що поведінка B0 може бути представлена, як послідовність дій a1 та a2 та поведінки B1, або як послідовність дії a3 та поведінки B2. Поведінка B1 закінчується після дії a4 [4], [6].

2.2. Приклад формалізації системи взаємодії агентів закритої логістичної системи

Розглянемо застосування на практиці алгебри поведінок на прикладі діючої закритої логістичної системи фермерського господарства. Закрите середовище містить в собі набір агентів, що мають певні атрибути, властивості та список дій, які вони виконують за певних передумов. Контроль та взаємодія агентів даної системи здійснюється за допомогою Централізованої бази даних (сервера), яка

збирає інформацію про стан кожного агенту та координує їх взаємодію за допомогою команд. Централізована база даних є одним із агентів системи, функція якого є безперервне отримання, накопичення та обробка інформації від агентів. За умови виконання певної передумови Централізована база даних надсилає агенту сповіщення-команду (коротке сервісне повідомлення) про початок виконання певної дії. Активність кожного агенту описана математичною моделлю поведінок, яка передбачає хід усіх можливих дій такого агенту. Моделювання поведінок усіх агентів дає змогу проаналізувати цілісну комплексну логістичну систему в роботі та виявити її вразливості.

Список основних агентів та їх дій наведено в Таблиці 2.1.

Таблиця 2.1.

Список агентів та їх дій

Агент	Список дій
Агроном	- дає команду про готовність сої для збору урожаю
Комбайн (один або більше)	- простій (або зупинка) - збір урожаю та вивантаження на ходу у вантажівку - технічне обслуговування - заправка - заміна водія при цілодобовій роботі позмінно
Бензовоз	- простій (або зупинка) - забір пального на базі - заправка транспортних засобів та механізмів у разі отримання сигналу від централізованої бази даних про низькі об’єми палива, отримані від датчиків палива цих засобів та механізмів - заміна водія при цілодобовій роботі позмінно - технічне обслуговування - заправка
Вантажівка (одна або більше)	- - простій (або зупинка) - супровід комбайна під час збору урожаю та приймання сої на ходу - транспортування на склад та вивантаження - технічне обслуговування - заправка - заміна водія при цілодобовій роботі позмінно
Склад та лінія переробки	простій (або зупинка) - приймання сої та зважування - контроль показників якості

	- контроль вологості та досушка, провіювання - переробка сої на соєву олію - зважування готової продукції - відвантаження готової продукції - заміна працівників при цілодобовій роботі позмінно - передача отриманої інформації на всіх етапах до Централізованої бази даних
Фура	- простій (або зупинка) - очікування завантажування на складі - завантажування та зважування - доставка кінцевому споживачеві (по Україні або за кордон)
Сервісна служба (механік)	- - простій (або зупинка) - виконання технічного обслуговування - заправка - заміна водія при цілодобовій роботі позмінно
Централізована база даних (ЦБД)	- безперервний збір та обробка інформації від GPS та інших датчиків від всіх агентів на всіх етапах - передача команд та інформації відповідним агентам - формування бази даних - опрацювання отриманої інформації на всіх етапах від агенту Склад та лінія переробки, внесення до операційної бази даних - надання інформації зацікавленим сторонам та агентам

Далі розглянемо приклади запису моделей поведінок агентів за допомогою алгебри поведінок, зокрема розглянемо поведінки декількох базових агентів:

1) Комбайн

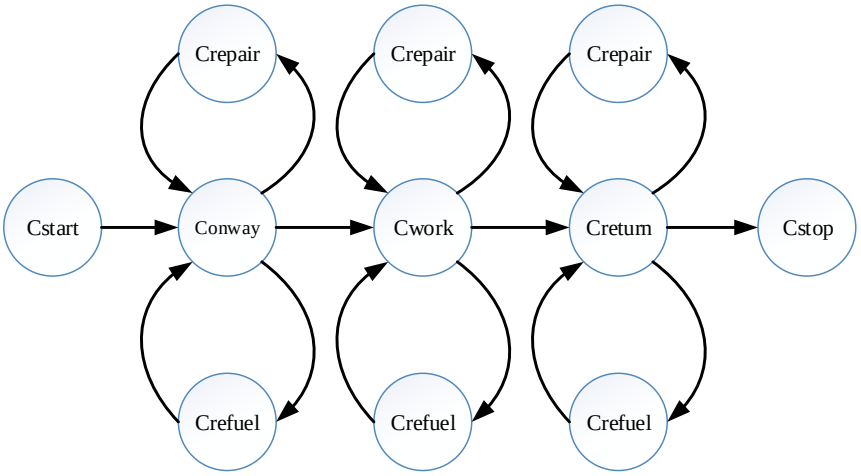


Рис. 2.1. Схема математичної моделі комбайну.

Математична поведінкова модель:

```
C = Cstart.Conway.Cwork.Creturn.Cstop
Conway = Cmove.Conway + Cmove.Crepair.Conway +Cmove.Crefuel.Conway
Creturn =
Cmove.Creturn + Cmove.Crepair.Creturn+ Cmove.Crefuel.Creturn
Cmove = (PdX.Cmove + PdY.Cmove) + Conplace
```

Поведінка переміщення комбайна до робочого місця Conway та назад до бази Creturn складається з поведінки циклу Cmove, що закінчується Conplace та поведінок ремонту Crepair та заправки Crefuel.

```
Cwork = Cgath.Cwork + Cgath.Crepair.Cwork + Cgath.Crefuel.Cwork
Cgath = (SdX.Cgath + Sdy.Cchdir.Cgath) + Send
```

Поведінка роботи комбайна Cwork складається з поведінки циклу Cgath, що закінчується Send та поведінок ремонту Crepair та заправки Crefuel.

```
Crepair = Cstop.Cdamage,
Поведінка ремонту комбайна Crepair є дії Cstop та Cdamage.
Crefuel = Cstop.Cfuel
```

Поведінка заправки комбайна Crefuel є дії Cstop та Cfuel.

2) Бензовоз

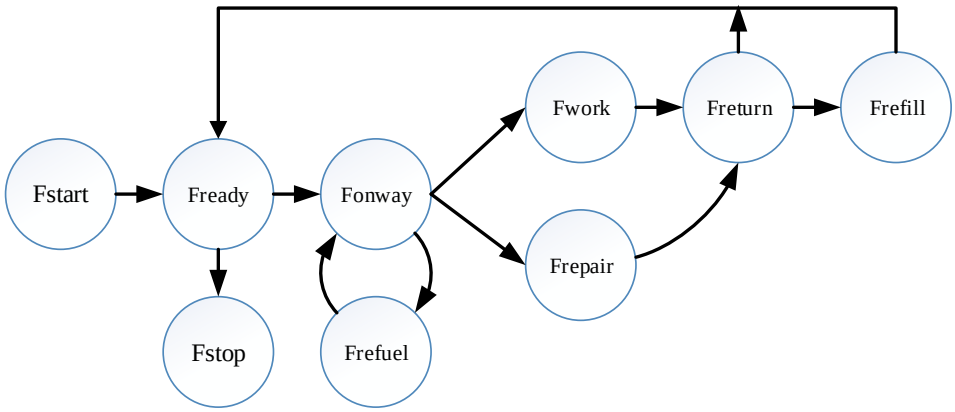


Рис. 2.2. Схема математичної моделі бензовозу.

Математична модель поведінки:

```
F = Fstart.Fcycle.Fstop,
Fcycle = Fready.Fcycle + Fready.Fonway.Fwork.Freturn.Frefill.Fcycle +
 Fready.Fonway.Freturn.Fcycle
```

Поведінка робочого циклу бензовоза Fcycle є циклом зміни дій:

```
Fonway = Fmove.Fonway + Fmove.Frepair + Fmove.Frefuel.Fonway,
Freturn =
Fmove.Freturn + Fmove.Frepair.Freturn + Fmove.Frefuel.Freturn
```

Поведінка переміщення бензовоза до робочого місця Fonway та назад до бази Freturn складається з поведінок ремонту Frepair, заправки Frefuel та поведінки циклу Fmove.

```
Fmove = (SdX.Fmove + SdY.Fmove) + Fonplace
```

Поведінка переміщення бензовоза до робочого місця Fmove є циклом зміни положення, що закінчується Fonplace або поведінками ремонту Frepair та заправки Frefuel.

```
Frepair = Fstop.Fdamage,

Поведінка ремонту бензовоза Frepair є дії Fstop та Fdamage.
Frefuel = Fstop.Ffuel.Fwork.
```

Поведінка заправки бензовоза Frefuel є дії Fstop, Ffuel та Fwork.

3) Механік

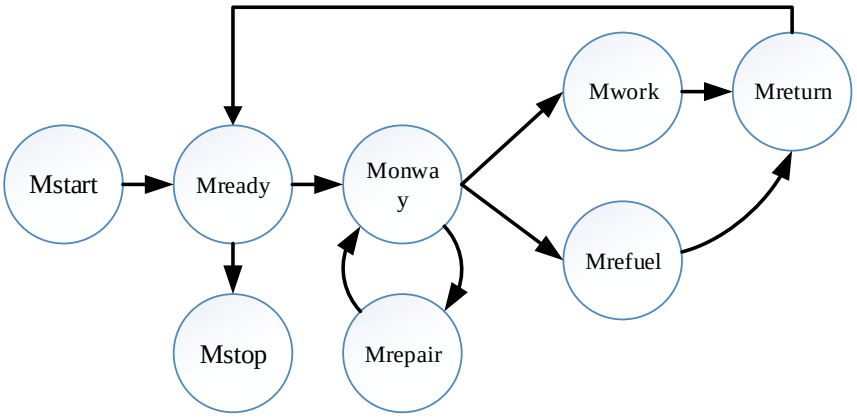


Рис. 2.3. Схема математичної моделі механіка.

Математична модель:

```
M = Mstart.Mcycle.Mstop
Mcycle = Mready.Mcycle + Mready.Monway.Mwork.Mreturn.Mcycle + Mready.
Monway.Mreturn.Mcycle
```

Поведінка робочого циклу механіка Mcycle є циклом зміни дій

```
Monway = Mmove.Monway + Mmove.Mrefuel + Mmove.Mrepair.Monway
Mreturn =
Mmove.Mreturn + Mmove.Mrepair.Mreturn + Mmove.Mrefuel.Mreturn
```

Поведінка переміщення механіка до робочого місця Monway та назад до бази Mreturn складається з поведінок ремонту Mrepair, заправки Mrefuel та поведінки циклу Mmove

```
Mmove = (SdX.Mmove + SdY.Mmove) + Monplace
```

Поведінка переміщення механіка до робочого місця Mmove є циклом зміни положення, що закінчується Monplace або поведінками ремонту Mrepair та заправки Mrefuel

```
Mrepair = Mstop.Mdamage.Mwork
```

Поведінка ремонту механіка Mrepair є дії Mstop та Mdamage

```
Mrefuel = Mstop.Mfuel
```

Поведінка заправки механіка Mrefuel є дії Mstop, Mfuel та Mwork

4) Вантажівка

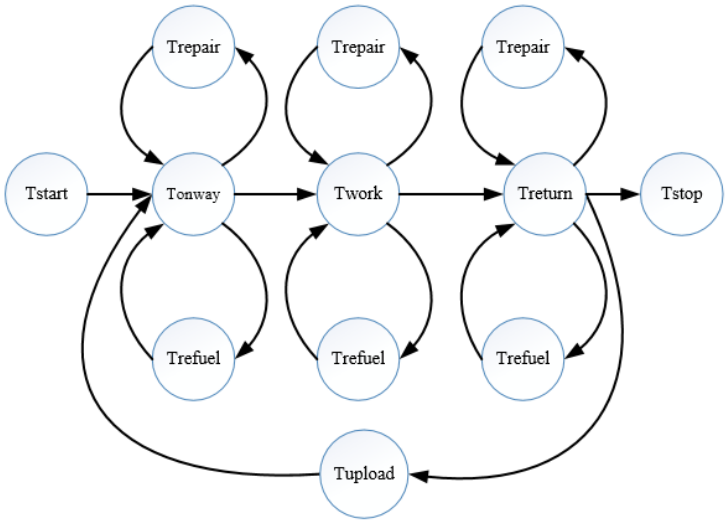


Рис. 2.4. Схема математичної моделі вантажівки.

Математична модель:

```
T = Tstart.Tonway.Twork.Treturn.Tstop
 Tonway = Tmove.Tonway + Tmove.Trepair.Tonway + Tmove.Trefuel.Tonway
Treturn =
Tmove.Treturn + Tmove.Trepair.Treturn + Tmove.Trefuel.Treturn
 Tmove = (PdX.Tmove + PdY.Tmove) + Tonplace
```

Поведінка переміщення вантажівки до робочого місця Tonway та назад до бази Treturn складається з поведінки циклу Tmove, що закінчується Tonplace та поведінок ремонту Trepair та заправки Trefuel

```
Twork = Tgath.Twork + Tgath.Trepair.Twork + Tgath.Trefuel.Twork + Tga
th.Tunload.Twork
Tgath = (SdX.Tgath + Sdy.Tchdir.Tgath) + Send
```

Поведінка роботи вантажівки Twork складається з поведінки циклу Tgath, що закінчується Send та поведінок ремонту Trepair та заправки Trefuel

```
Trepair = Tstop.Tdamage
```

Поведінка вантажівки під час ремонту

Trepair є дії Tstop та Tdamage

```
Trefuel = Tstop.Tfuel
```

Поведінка заправки вантажівки Trefuel є дії Tstop та Tfuel

```
Tunload = Treturn.Treload.Tonway
```

Поведінка вигрузки вантажівки Tunload є дії Treturn, Treload та Tonway

5) Схема математичної моделі Склад

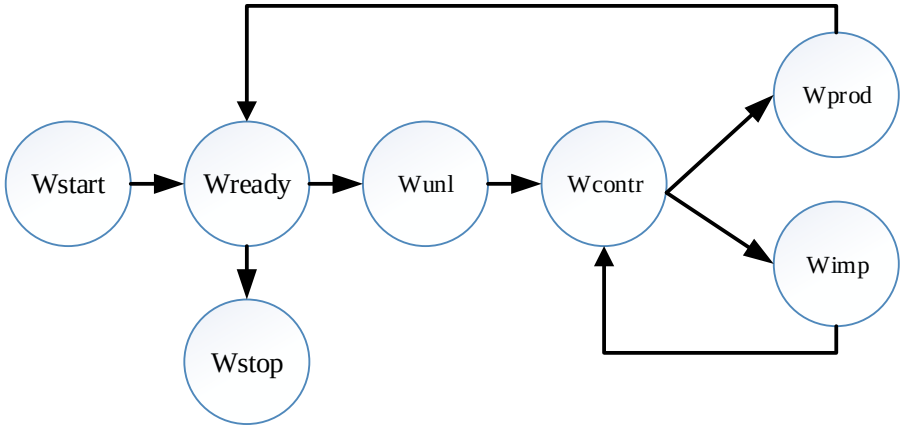


Рис. 2.5. Схема математичної моделі Склад.

Математична модель:

```
W = Wstart.Wcycle.Wstop
```

Поведінка циклу роботи:

```
Wcycle = Wready.Wcycle + Wready.Wunload.Wctrcycle.Wcycle
```

Поведінка циклу контролю:

```
Wctrcycle = Wcontrol.Wimprove.Wctrcycle + Wcontrol.Wproduction
```

6) Схема математичної моделі Лінія переробки

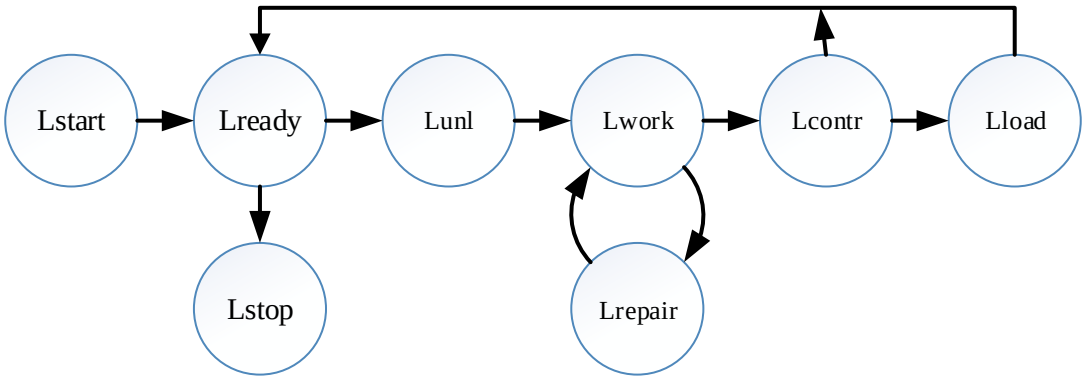


Рис. 2.6. Схема математичної моделі Лінія Переробки.

Математична модель:

```
L = Lstart.Lcycle.Lstop
```

Поведінка циклу роботи:

```
Lcycle = Lready.Lcycle + Lready.Lunload.Lwork.Lcontrol.Lcycle + Lready. Lunload.Lwork.Lcontrol.Lload.Lcycle
```

Поведінка циклу виробництва олії:

```
Lwcycle = Lwork.Lwcycle + Lwork.Lrepair. Lwcycle
```

7) Схема математичної моделі Фура

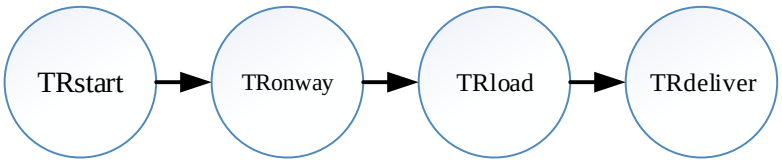


Рис. 2.7. Схема математичної моделі Фура.

Математична модель:

```
TR = TRstart.TRonway.TRload.TRdeliver
TRonway =  TRmove.TRonway
 TRmove = (PdX.TRmove + PdY.TRmove) + TRonplace
```

8) Централізована база даних (ЦБД)

Централізована база даних є агентом, який виконує функцію контролю усіх інших агентів, збору, обробки та обміну інформації від агентів та формування єдиної бази даних, тому даний агент присутній на всіх етапах дій усіх без

виключення агентів безперервно. Тому ми не описуємо математичну модель даного агенту, так як він сам є інструментом контролю та керування післяумов та передумов та інформаційним середовищем для усіх агентів.

Крім поведінок ми маємо також список дій, що можуть бути представленими перед- та післяумовою. Всі дії агентів приведені у вигляді діаграм MSC (Message Sequence Charts).

1) Список дій та їх передумов та післяумов агенту Комбайн

```
Cstart = (1 -> <> (X==Xs) && (Y ==Ys))
/*Початок роботи*/
Cdamage = ((1 <= i <= combineQuantity) -> <send help(i)>
combine(i).state = DAMAGE)
/*Якщо трапилась поломка відіслати сигнал про допомогу та перейти в
стадію DAMAGE*/
Cstop = ((1 <= i <= combineQuantity) -> <> combine(i).state = STOP)
/* Зупинитися та перейти в стадію STOP*/
Cfuel = ((1 <= i <= combineQuantity) && (combine(i).fuelLevel <= 0.2
* combine(i).fuelVolume) -> <send help(i)> combine(i).state = FUEL)
/* Якщо рівень пального менше норми відіслати сигнал про допомогу та
перейти в стадію FUEL */
PdX = ((X < Xp) -> <> X = X + deltaX)
/*Якщо не на місці призначення, то рухатись на якусь deltaX */

PdY = ((Y < Yp) -> <> Y = Y + deltaY)
/*Якщо не на місці призначення, то рухатись на якусь deltaY */

SdX = (((dir > 0) && (X < XLe)) || ((dir < 0) && (X > XLs)) -> <> X =
X + deltaX * dir)
/*Якщо не кінець довжини ділянки, то рухатись на якусь deltaX */

SdY = ((((dir > 0) && (X > XLe)) || ((dir < 0) && (X < XLs))) && (Y <
YWe) -> <> Y = Y + deltaY)
/*Якщо кінець довжини ділянки, але не кінець ширини поля, то
починаємо
нову смугу на полі*/
Cchdir = ((((dir > 0) && (X > XLe)) || ((dir < 0) && (X < XLs))) &&
(Y < YWe) -> <> dir = -dir)
/*Якщо кінець довжини ділянки, але не кінець ширини поля, то змінюємо
напрямок*/

Conplace = ((X == Xp) &&(Y == Yp) -> <> 1)
/*Якщо комбайн на місці призначення*/
Send = ((((dir > 0) && (X > XLe)) || ((dir < 0) && (X < XLs))) && (Y
== YWe) -> <> 1)
/*Якщо кінець поля*/
```

2) Список дій та їх передумов та післяумов агенту Бензовоз

```
Fstart = (1 -> <> (X==Xs) && (Y ==Ys) )
/*Початок роботи*/
Fready = (1 -> <> fuelTruck(i).state = READY)
/*Перехід в стан готовності*/
Fwork = (1 -> <> fuelTruck(i).state = WORK)
/*Перехід в стан виконання заправки*/
Fdamage = ((1 <= i <= fuelTruckQuantity) -> <send help(i)>
fuelTruck(i).state = DAMAGE)
/*Якщо трапилась поломка відіслати сигнал про допомогу та перейти в
стадію DAMAGE*/
Fstop = ((1 <= i <= fuelTruckQuantity) -> <> fuelTruck(i).state =
STOP)
/* Зупинитися та перейти в стадію STOP*/
Ffuel = ((1 <= i <= fuelTruckQuantity) && (fuelTruck(i).fuelLevel <=
0.2 * fuelTruck(i).fuelVolume) -> <> fuelTruck(i).state = FUEL)
/* Якщо рівень пального менше норми перейти в стадію FUEL */
Frefill = ((1 <= i <= fuelTruckQuantity) &&
(fuelTruck(i).cycternLevel <= 0.2 * fuelTruck(i).cycternVolume) -> <>
fuelTruck(i).state = REFILL)
/* Якщо рівень пального в цистерні менше норми перейти в стадію
REFILL */

PdX = ((X < Xp) -> <> X = X + deltaX)
/*Якщо не на місці призначення, то рухатись на якусь deltaX */

PdY = ((Y < Yp) -> <> Y = Y + deltaY)
/*Якщо не на місці призначення, то рухатись на якусь deltaY */
Fonplace = ((X == Xp) &&(Y == Yp) -> <> 1)
/*Якщо бензовоз на місці призначення*/
```

3) Список дій та їх передумов та післяумов агенту Механік

```
Mstart = (1 -> <> (X==Xs) && (Y ==Ys) )
/*Початок роботи*/
Mready = (1 -> <> mechanic(i).state = READY)
/*Перехід в стан готовності*/
Mwork = (1 -> <> mechanic(i).state = WORK)
/*Перехід в стан виконання ремонту*/
Mdamage = ((1 <= i <= mechanicQuantity) -> < > mechanic(i).state =
DAMAGE)
/*Якщо трапилась поломка перейти в стадію DAMAGE*/
Mstop = ((1 <= i <= mechanicQuantity) -> <> mechanic(i).state = STOP)
/* Зупинитися та перейти в стадію STOP*/
Mfuel = ((1 <= i <= mechanicQuantity) && (mechanic(i).fuelLevel <=
0.2 * mechanic(i).fuelVolume) -> < send help(i) > mechanic(i).state =
FUEL)
/* Якщо рівень пального менше норми відіслати сигнал про допомогу та
перейти в стадію FUEL */

PdX = ((X < Xp) -> <> X = X + deltaX)
```

```
/*Якщо не на місці призначення, то рухатись на якусь deltaX */

PdY = ((Y < Yp) -> <> Y = Y + deltaY)
/*Якщо не на місці призначення, то рухатись на якусь deltaY */
Monplace = ((X == Xp) &&(Y == Yp) -> <> 1)
/*Якщо механік на місці призначення*/
```

4) Список дій та їх передумов та післяумов агенту Вантажівка

```
Tstart = (1 -> <> (X==Xs) && (Y ==Ys))
/*Початок роботи*/
Tdamage = ((1 <= i <= truckQuantity) -> <send help(i)> truck(i).state
= DAMAGE)
/*Якщо трапилась поломка відіслати сигнал про допомогу та перейти в
стадію DAMAGE*/
Tstop = ((1 <= i <= truckQuantity) -> <> truck(i).state = STOP)
/* Зупинитися та перейти в стадію STOP*/
Treload = ((1 <= i <= truckQuantity) -> <> truck(i).state = UNLOAD)
/* Зупинитися та перейти в стадію UNLOAD*/
Tfuel = ((1 <= i <= truckQuantity) && (truck(i).fuelLevel <= 0.2 *
truck(i).fuelVolume) -> <send help(i)> truck(i).state = FUEL)
/* Якщо рівень пального менше норми відіслати сигнал про допомогу та
перейти в стадію FUEL */
PdX = ((X < Xp) -> <> X = X + deltaX)
/*Якщо не на місці призначення, то рухатись на якусь deltaX */

PdY = ((Y < Yp) -> <> Y = Y + deltaY)
/*Якщо не на місці призначення, то рухатись на якусь deltaY */

SdX = (((dir > 0) && (X < XLe)) || ((dir < 0) && (X > XLs)) -> <> X =
X + deltaX * dir)
/*Якщо не кінець довжини ділянки, то рухатись на якусь deltaX */

SdY = ((((dir > 0) && (X > XLe)) || ((dir < 0) && (X < XLs))) && (Y <
YWe) -> <> Y = Y + deltaY)
/*Якщо кінець довжини ділянки, але не кінець ширини поля, то
починаємо
нову смугу на полі*/
Tchdir = ((((dir > 0) && (X > XLe)) || ((dir < 0) && (X < XLs))) &&
(Y < YWe) -> <> dir = -dir)
/*Якщо кінець довжини ділянки, але не кінець ширини поля, то змінюємо
напрямок*/

Tonplace = ((X == Xp) &&(Y == Yp) -> <> 1)
/*Якщо вантажівка на місці призначення*/
Send = ((((dir > 0) && (X > XLe)) || ((dir < 0) && (X < XLs))) && (Y
== YWe) -> <> 1)
/*Якщо кінець поля*/
```

5) Список дій та їх передумов та післяумов агенту Склад

```
Wstart = (1 -> <>1)
/*Початок роботи*/
Wready = (1 -> <> warehouse.state = READY)
/*Перехід в стан готовності*/
Wunload =
Wcontrol
Wimprove
Wproduction
Mstop = ((1 <= i <= mechanicQuantity) -> <> mechanic(i).state = STOP)
/* Зупинитися та перейти в стадію STOP*/
```

Детальніше приклади формалізації викладені в Додатку Б, Додатку В та Додатку Г.

2.3. Формальна верифікація моделей

Дана методика формалізації дозволяє застосувати далі формальні методи верифікації, таких як перевірка властивостей безпеки та випадки загрози для безперервної життєдіяльності системи. Надзвичайно важливо перевіряти деякі ключові фактори, які впливають на можливість виконання злагоджених дій усіх агентів системи, що в результаті і відображається в безперервній та продуктивній діяльності системи. Розглянемо основні властивості безпеки, що перевіряються в логістичній системі фермерського господарства.

1. Терміни збору урожаю. Дана властивість може бути виражена формулою в базовій логічній мові, що вибрана для формалізації дій (T<= 90 днів). При розгляді довільної кількості комбайнів та обслуговуючого транспорту така ситуація може бути при нераціональній кількості транспортних та стану техніки. Для перевірки властивості використовуються методи моделювання такі як символьне, конкретне та змішане моделювання системи. На початку враховуються такі умови, як кількість допустимих несправностей техніки (як арифметична нерівність), конкретна кількість одиниць техніки та відстані до місця роботи. Для визначених початкових даних проводиться конкретне моделювання та визначається властивість порушення термінів збору врожаю. При використанні зворотнього символьного моделювання від стану, що відповідає дотримання термінів та

гіпотетичних можливих кількостей одиниць, несправностей та відстаней до місця роботи, можливо визначити початкові умови при яких ця властивість не буде порушена. Слід відмітити, що дана технологія не вирішує задачу оптимізації, а лише перевіряє можливість її реалізації. Методи символьного моделювання розглянуті в [7], [8], [9].

2. Достатність кількості обслуговуючих агентів (типу бензовоз чи механік) для обслуговування життєдіяльності головних агентів системи, здатність вчасно прибути до заданої точки призначення для виконання заданих Центральною базою даних дій. Дані властивості перевіряються аналогічно при висуванні деяких гіпотез при проектуванні задачі або реальної наявності техніки. Такі властивості також представляються за допомогою нерівностей (або рівностей).

3. Перевірка злагодженості та узгодженості алгоритмів дій агентів (наприклад, дії бензовоза при одночасному виклику обслуговування декількох агентів). Дана властивість може бути виражена за допомогою алгебри поведінок. Наприклад, чи встигне бензовоз заправити всі комбайни, якщо вони одночасно зупиняться, за встановлений термін. Маємо властивість безпеки (T<=2 год) та саму поведінку:

Cdamage.Cdamage.Cdamage.Cdamage.X;(Trepair.Trepair.Trepair.Trepair).

Розглянутий приклад ілюструє поведінку чотирьох комбайнів, які мали несправність та були полагоджені за деякий час. Чи відповідає цей час встановленому визначає конкретне або змішане моделювання. Зауважимо, що при символьному моделюванні можливо оцінити властивість для довільної кількості комбайнів.

4. Вплив непередбачуваних факторів типу погоди, рельєфу та інше. Дана властивість вивчається за допомогою встановлення можливих дій, що моделюють дані фактори та відповідними видами моделювання.

5. Дотримання умов перевезення вантажу (в нашому випадку зібраного урожаю) при передачі від одного агенту іншому. При перевезенні розглядаються додаткові фактори середовища, які моделюються в процесі верифікації. Це час

перевезення, вологість, температура, які беруться до уваги в моделі, як додаткові атрибути системи.

Формальна верифікація із застосуванням алгебраїчного підходу полягає у вирішенні задачі досяжності властивостей або їх порушення. Якщо властивість досяжна в даній моделі, то можна побудувати приклад сценарію, що веде до порушення властивості або до виконання її.

Символьне моделювання полягає в побудові символьного сценарію наступним чином.

1-й крок. Визначення початкової формули середовища над атрибутами. Для моделі логістики у фермерському господарстві – це визначення початкових координат агентів, їх початкові властивості (кількість палива, кількість робітників).

2-й крок – це визначення, яка наступна дія можлива згідно поведінки. Для цього відповідно до поведінки розглядаємо передумову можливої дії. Якщо кон'юнкція

E∧Рвиконувана, то дія можлива. E – поточна формула середовища, P – передумова.

3-й крок – це зміна формули середовища. Для цього використовуємо теорію предикатних трансформерів (PREDICATETRANSFORMERS) [10] та відповідно змінюємо формулу середовища згідно із постумовою.

4-й крок – перевіряємо чи досяжна властивість. Для цього перевіряємо на виконуваність кон'юнкції $E \wedge X$, де X – формула властивості, E – поточна формула середовища.

5-й крок – перевіряємо можливу властивість закінчення моделювання, якщо вона не виконується, то повторюємо крок 2.

В символьному моделюванні існує специфіка роботи із розподіленими системами. По-перше в розподілених системах є висока ступінь паралелізму і тому маємо комбінаторний вибух із перестановками можливих дій. Але експоненціальний вибух значно знижується, якщо використовувати властивість перестановки або взаємозамінності дій в поведінці. Інформаційно незалежні дії є

взаємозамінні в формулі поведінки. Тому перед моделюванням може стояти задача визначення властивості взаємозамінності для дій.

Доведення взаємозамінності та саме моделювання можливо проводити відносно незалежного інформаційного середовища. Тому використовуємо поняття шару (slice), в якому абстрагуємось від інформаційно незалежних атрибутів. В прикладі логістичної системи маємо шар, що визначає координати, та шар, що визначає кількість палива. Тим самим ми зменшуємо складність та підвищуємо гарантію можливості доведення досяжності властивості.

2.4. Модельне тестування.

Маючи в якості артефактів етапу вимог та дизайну моделі відповідного рівня абстракції, можливо отримати набір тестів, який буде використовуватися для тестування на відповідних етапах модельної розробки. Тести, що були згенеровані на етапі вимог та дизайну використовуються для тестування, як на етапі дизайну та на етапах модульного, інтеграційного та системного тестування.

Використовуючи моделі для системи логістики фермерського господарства, створені в попередніх підрозділах для верифікації умов безпеки, можемо отримати тестовий набір методом символьного моделювання.

При генерації тестового набору у вигляді сценаріїв поведінки можемо визначити критерій покриття моделі. Основні критерії, які використовуються при генерації тестів це покриття всіх вершин графу, покриття всіх ребер графу та покриття всіх станів графу. Генерація покриття всіх станів є вичерпною, тобто ми отримуємо всі сценарії поведінки системи. При покритті станів використовують поняття еквівалентності станів, що дає змогу згенерувати набір сценаріїв з точністю до визначеної еквівалентності. Але у вичерпній генерації може виникнути явище експоненціального або комбінаторного вибуху, що приведе до дуже тривалого або нескінченого процесу.

Генерації із іншими видами покриття регулюються відповідними алгоритмами та можуть отримувати відповідну множину сценаріїв за менший час. Але разом із тим є проблема недосяжності вершин або ребер та в цьому випадку

набір сценаріїв з відповідним покриттям згенерувати неможливо. Тому розглядають допустиму процентну оцінку кількості згенерованих тестів.

Аналогом покриття в алгебрі поведінок є покриття всіх дій та всіх можливих пар дій, що допустимі в поведінці. Графічним представленням поведінки є UCM граф, в якому є вершини та ребра і тоді поняття покриття переноситься на покриття елементів графу UCM.

Для системи логістики фермерського господарство було згенеровано набір сценаріїв із покриттям всіх ребер графу поведінки.

2.5. Верифікація і формалізації системи міжнародної логістики.

Відкрита система міжнародної логістики є продовженням описаної закритої логістичної обслуговуючої системи фермерського господарства. Вихідний продукт фермерського господарства (наприклад, соя) є вхідним продуктом для відкритої міжнародної логістичної системи (переробка сої на соєву олію та її експорт). Істотною відмінністю цієї системи є теоретично нескінченна кількість не пов’язаних між собою незалежних агентів кожного типу, і, як наслідок, постійна зміна середовищ існування. Аналогічно з закритою логістичною системою, де функцію збору інформації та контролю виконує Централізована база даних, в даній відкритій системі існує подібний агент – Система. Однак на відміну від закритої системи, де агенти діють злагоджено та в спільних інтересах, вимоги надійності та безпеки до відкритої системи значно вищі, так як кожен з незалежних агентів діє в своїх особистих інтересах. Тому для даних цілей доцільно застосовувати технологію розподілених децентралізованих систем (блокчейн) замість стаціонарного серверу. Список типових основних агентів, їх функції та інформацію, яку вони надають розподіленій системі, наведено в Таблиці 2.2.

Таблиця 2.2.

Список агентів, їх функцій та інформації, яку вони надають централізованій системі

Агент	Функції	Інформація, яку надає для системи
Виробник (він же експортер) товару	- виробництво товару - пошук покупця - аналіз цін (пропозицій) та вибір оптимального покупця - відвантаження товару - пошук оптимальних логістичних агентів – портових операторів та морських перевізників	наявний об’єм товару до відвантаження - базові характеристики та якість товару (замовляється експортером) - узгоджена ціна - узгоджений покупець - узгоджені з Покупцем умови відвантаження та оплати всіх дій (хто оплачує послуги посередників та на яких етапах здійснення) - дата готовності до відвантаження - узгоджені логістичні агенти – портовий оператор та морський перевізник
Портовий оператор	- здійснення внутрішньої обслуговуючої логістики від виробника до завантаження на морське судно	- дата подачі контейнера виробнику згідно дати готовності до відвантаження, надану виробником (пізніше або в день його дати) - підтвердження здійснення та дата проміжних етапів: відвантаження у виробника, прибуття в порт, завантаження на судно та інші можливі етапи
Морський перевізник	- приймає на борт судна контейнер від Портового оператора та здійснення перевезення	- факт та дата прийняття контейнера на борт судна
Покупець товару	- пошук продавця - аналіз цін (пропозицій) та вибір оптимального продавця - оплата за товар	- - бажаний об’єм товару для відвантаження - мінімальні характеристики та якість товару (підтверджується продавцем) - узгоджена ціна - узгоджений продавець - узгоджені з продавцем умови відвантаження та оплати всіх дій (хто оплачує послуги посередників та на яких етапах здійснення) - підтвердження наданих продавцем логістичних агентів – портовий оператор та морський перевізник

Банк кожного з агентів	- здійснює оплату за товар чи послуги згідно інформації, отриманої від свого агенту	інформація про суму та дату здійснення оплати, кому з агентів та за яку роботу

Поведінка та дії агентів описуються аналогічно моделі фермерського господарства. До моделі включаються відстані та географічна інформація про точки в яких діють агенти. В якості властивостей, які можуть бути перевірені, можуть бути наступні:

- Своєчасна доставка;
- Дотримання температурного режиму;
- Дотримання режиму вологості;
- Здатність протистояти зовнішнім факторам.

Властивості перевіряються такими методами алгебри поведінок, як алгебраїчне зіставлення, символьне моделювання системи із довільною кількістю агентів та статичні методи доведення. Специфікою системи є використання децентралізованих систем, які є однотипними, але зберігають всі атрибути, які стосуються всього ланцюга постачання, дані про перевозку, транзакції з оплатою тощо. Кожен агент зберігає файл з інформацією про ланцюг постачання або транзакції. Ми розглядаємо два типи використання формальних методів в даній моделі.

1. Моделювання ланцюгу постачання для перевірки властивостей. В даній процедурі ми перевіряємо чи відповідає вибраний ланцюг постачання вимогам перевезення. Перевіряються також властивості супротиву до зовнішніх небажаних дій – погоди, шахрайства, кібератаки.

2. Моніторінг перевезення згідно вибраної моделі безпеки. Дана процедура призначена для виявлення порушення властивостей безпеки заздалегідь при вивченні сценарія поведінки. При моніторінгу разом із моделюю безпеки може бути використана модель класифікації, що створена за допомогою машинного навчання.

Висновки до Розділу 2.

Методи математичного моделювання з такими інструментами, як алгебра поведінок, дають можливість керувати логістичними системами та забезпечувати їх безпеку та захищеність за допомогою формальної верифікації та формалізації за допомогою тестування на основі моделей. Алгебраїчні моделі логістичних систем можуть бути використані для аналізу поведінки всіх залучених агентів і доведення їх здатності виконувати свої цілі та здатності всієї системи постійно існувати та залишатися стабільною. Алгебраїчний підхід та інсерційне моделювання дозволяють довести або спростувати властивості, подавши контр-приклади для системи з довільною кількістю агентів. Використовуючи формальну модель програми на якомусь рівні абстракції, ми можемо генерувати різні сценарії поведінки агентів або груп агентів. Ці сценарії є символьними і можуть бути проілюстровані конкретними прикладами за допомогою формальних методів. Генерація символічних сценаріїв забезпечує гарне висвітлення поведінки, а отже, отримані конкретні сценарії можуть розглядатися як тестовий набір для створеного програмного забезпечення.

Специфікою міжнародних логістичних систем є необхідність безперервного контролю місцезнаходження кожного агенту в певний момент часу, тобто орієнтація рухомих об’єктів в часі та просторі. Тому для верифікації на всіх рівнях модельної розробки необхідно створити модуль, яка має специфіку – протоколи руху. При цьому логістична система з міркувань безпеки збереження та обміну інформацією є розподіленою та синхронізованою по часу.

В символьному моделюванні існує специфіка роботи із розподіленими системами. Її складність для логістичних систем полягає в тому, що в ній діють велика кількість незалежних один від одного агентів при великій кількості сценаріїв, що вирішується символьним моделюванням та алгоритмами налаштування інтерлівінгу (шарування). Також проблема значно спрощується, якщо використовувати властивість перестановки або взаємозамінності дій в

поведінці. Тому перед моделюванням може стояти задача визначення властивості взаємозамінності для дій. Доведення взаємозамінності та саме моделювання можливо проводити відносно незалежного інформаційного середовища. Для логістичних систем використовуємо специфічне моделювання із використанням шарів, в якому абстрагуємось від інформаційно незалежних атрибутів. Основними та найважливішими шарами в відкритих чи закритих системах логістики різних рівнів є їх координати у просторі, рівень та стан критично важливих показників агентів траспортування та характеристики товару, що визначають його умови безпечного перевезення.

Список використаних джерел до Розділу 2.

1. BPMN (Business Process Model and Notation). www.bpmn.org

2. UML (Unified Modelling Language). www.uml.org

3. SysML (Systems Modelling Language). www.sysml.org

4. VDM (Vienna Development Method). www.vienna.cc/e/evdm.htm

5. SPIN. http://spinroot.com/spin/whatispin.html

6. APS (Algebraic Programming System).www.apsystem.org.ua

7. A. Letichevsky, O. Letychevskyi, and V. Peschanenko, “Insertion Modeling and Its Applications,”Computer Science Journal of Moldova, vol. 24, no. 3, 2016, pp. 357-370.

8. A. Letichevsky and D. Gilbert, “Interaction of agents and environments,”inRecent Trends in Algebraic Development Technique, LNCS 1827,Springer-Verlag, 1999.

9. Letychevskyi O,, Letichevsky A., Peschanenko V., and Weigert T., “Insertion modeling and symbolic verification of large systems,”I LNCS 9369 SDL 2015: Model-Driven Engineering for Smart Cities. Springer, 2015, pp. 3-18.

10. «Свойства предикатного трансформера системы VRS» / Летичевский А.А., Годлевский А.Б., Песчаненко В.С., Потиенко С.В. // Кибернетика и системный анализ – 2010. – № 4. – С. 3–16.

РОЗДІЛ 3.
ПЕРЕВІРКА ВЛАСТИВОСТЕЙ КІБЕРБЕЗПЕКИ В ПРОЕКТУВАННІ ЛОГІСТИЧНИХ СИСТЕМ НА БЛОКЧЕЙН ПЛАТФОРМАХ.

3.1. Проблеми безпеки в проектуванні логістичної блокчейн платформи

Блокчейн технології в системах логістики використовуються з метою підвищення безпеки функціонування системи та протидії атакам, а також із метою підтримки цілісності та актуальності даних. Порівняємо дві архітектури систем на верхньому рівні абстракції з точки зору взаємодії агентів.

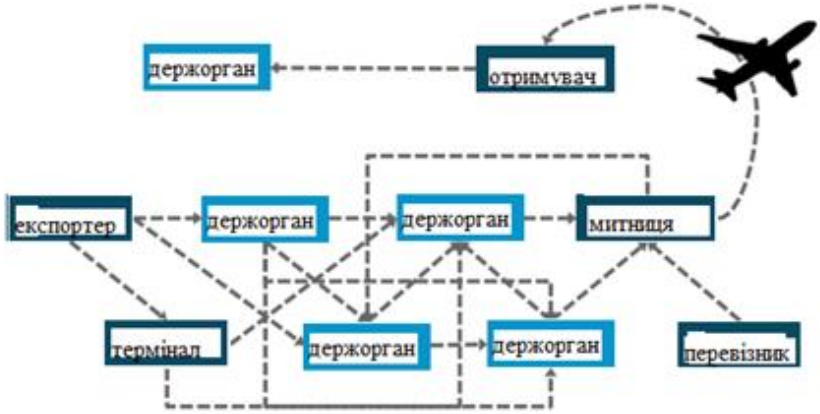


Рис. 3.1. Традиційна архітектура логістичної системи.

Впродовж ланцюжка постачання традиційного способу в процесі обміну сповіщеннями задіяна велика кількість агентів, кожен із яких має свої способи передачі сповіщень в різних форматах та відповідна передача документів між інстанціями, включаючи органи влади. При такій взаємодії інформація може губитись або спотворюватись відповідно до специфіки агентів. В процесах обміну документацією має місце шахрайство, зміна умов контракту, договірні випадки, адміністративний фактор.

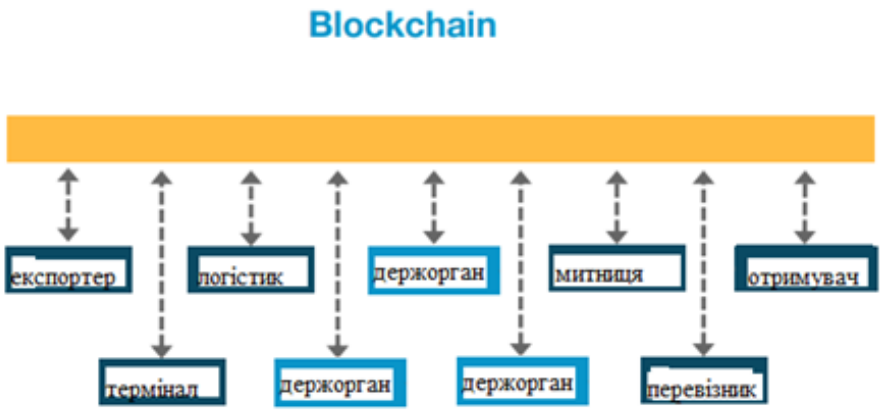


Рис.3.2. Блокчейнова архітектура логістичної системи

У випадку блокчейнової архітектури ланцюжку постачання в системі існує миттєвий та безпечний доступ до інформації про це ланцюжок постачання, відсутня централізація та адміністративний вплив. Цілісність даних підтримується в блоках, які неможливо спотворити. Мережевий криптозахист забезпечує високу ступінь безпеки на рівні неавторизованого проникнення. Перевірка глобальних властивостей безпеки, що мають підтримуватись при поставках, має більше можливостей для перевірки та більш захищена при блокчейновій архітектурі.

Попри це існують атаки, що можуть статися при певних умовах в структурі блокчейнової мережі. Якщо в мережі з'являється досить велика кількість зловмисників, то можливо заволодіти контролем над побудовою блокчейнової мережі та проводити атаки, що ведуть до незаконного присвоєння коштів та товарів. Зокрема, якщо система використовує ланцюжок постачання за допомогою розумних контрактів (смарт-контракти), то зловмисник може використовувати вразливості коду розумного контракту.

Таким чином при проектуванні системи необхідно оцінити ймовірність атаки та забезпечити стійкість системи за допомогою відповідного алгоритму консенсусу та верифікації розумних контрактів. Алгоритм консенсусу визначає правила поведінки агентів у децентралізованій системі, а розумні контракти визначають індивідуальні дії кожного з агентів при визначенні в системі ланцюжку постачання.

В процесі проектування будується модель взаємодії агентів на рівні блокчейну. Визначається, які сповіщення та інформація передаються між агентами та що складає зміст транзакцій, які формують блоки. Якщо для цього створюється алгоритм консенсусу блокчейнової мережі, то властивості безпеки перевіряються на рівні цього алгоритму. В цьому випадку будується модель алгоритму та проводиться відповідна верифікація.

Логістичні взаємодії визначаються на рівні розумних контрактів. Для розподіленої логістичної системи можуть додаватися або видалятися із мережі агенти із власними розумними контрактами, які визначають транзакції та відповідні артефакти взаємодії. Кожен новий розумний контракт, що з'являється в мережі, має бути верифікований в контексті взаємодії агентів.

Верифікація визначає два головних набори властивостей:

- глобальні властивості ланцюгу постачання, які зменшують ризики порушення норм, правил та умов логістики та забезпечують якісне виконання логістичних угод;
- властивості безпеки, що можуть виникнути у випадку наявності зловмисників та можливого шахрайства.

У методах верифікації знову розглядаємо алгебраїчний, зокрема інсерційний підхід в побудові моделей, та використання формальних методів верифікації.

3.2. Інсерційна семантика блокчейн-системи

В загальному вигляді алгоритм блокчейну можна розглядати як взаємодію вузлів деякої мережі. Кожен вузол може виступати в якості генератора блоку, який створює та надсилає блоки всім іншим вузлам відповідно до алгоритму консенсусу. З іншої сторони, вузол може також приймати блоки та іншу необхідну інформацію від інших вузлів для побудови блокчейну.

Модель блокчейн платформи можна розглядати як взаємодію деякого числа агентів у середовищі, в якому вони обмінюються сповіщеннями. Сповіщення містять інформацію, що є необхідною для побудови блокчейну.

Визначимо тип агенту NODE. Кожен агент має структуру блокчейн, яка накопичує блоки, що приходять від інших вузлів. Ці агенти, відповідно, мають атрибути, що зберігають блокову інформацію та посилання на наступні блоки.

При створенні блокчейну на кожному вузлі блоки створюють орієнтовний ациклічний граф. При нормальній роботі вузлів, тобто, якщо відсутній зловмисник та блоки не губляться при пересилці, послідовність блоків складає просто список, в якому блок стоїть на місці, що визначається відповідним часовим інтервалом. Граф визначає можливі альтернативні історії або послідовності отриманих блоків.

Середовище має атрибут TIME_SLOT, який визначається деяким часовим інтервалом, протягом якого має бути надісланий відповідний блок та прийнятий всіма іншими блоками. Алгоритм консенсусу визначає, який саме вузол має надсилати блоки згідно із умовами. При алгоритмі PoW (proof of work), право генерувати блок отримує вузол, який виконає швидше складну обчислювальну задачу, а алгоритм PoS (proof of stake) визначає вузол із найбільшим фінансовим потенціалом. В даному розгляді алгоритм не визначає визначення переможця та припускаємо, що вузли рівноправні та генерують блоки послідовно.

Механізм виникнення альтернативних історій пояснює наступна діаграма.

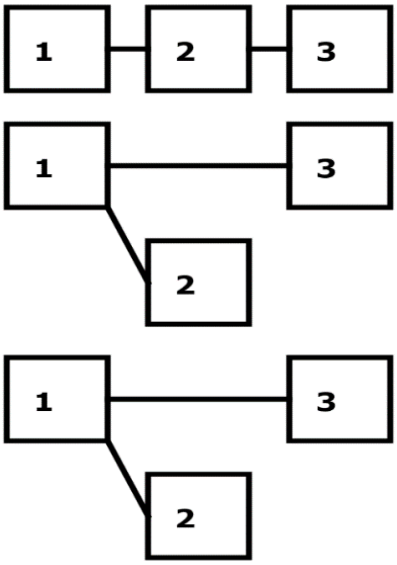


Рис.3.3. Утворення альтернативних історій при затримці відправлення блоків.

На рисунку показано три блокчейни трьох агентів. У першого агенту всі блоки прийшли вчасно, у свій визначений проміжок часу. У двох інших третій блок прийшов раніше другого в зв'язку із затримками у мережі, але другий вже був визначений, як послідовник першого блоку при генерації, тому виникає розгалуження ланцюжку, як альтернативна історія.

Для аналізу та моделювання роботи алгоритму блокчейн системи введемо атрибути, що визначають характеристики ациклічного орієнтовного графа.

CHAINS – кількість альтернативних *ланцюжків* в блокчейні. Кожен ланцюжок визначає відповідну історію або послідовність створення блоків, яка може бути відхилена при *фіналізації*. Процес фіналізації в алгоритмі визначає актуальний блокчейн без альтернативних історій.

HANG(j) – визначає множину висячих вершин у графі.

При моделюванні важлива також довжина ланцюжка. Тому вводяться атрибути номера ланцюжка максимальної та мінімальної довжини – MAX, MIN. Атрибут LEN(j) визначає довжину ланцюжка.

Кожен вузол має атрибут посилання на попередній блок REF_BLOCK(j). Атрибути BC та BB визначають останній створений та отриманий блок.

Булевська змінна Forking істинна, якщо немає в блокчейні альтернативних історій.

Початкова формула середовища визначає значення атрибутів для всіх агентів-вузлів.

```
Forall (int:i) (1<=i<=N) (Node(i).CHAINS == 1) && (Node(i).HANG(1) ==
0) && (Node(i).MAX == 1) && (Node(i).MIN == 1) && (Node(i).LEN(1) =
0) && (BN == 0)
```

Формула визначає початковий стан всіх агентів-вузлів, що мають початковий блок (нульову вершину), що називається *генезіс-блок*. Атрибут середовища BN визначає кількість згенерованих блоків.

Далі визначаємо алгоритм блокчейну, який буде описано за допомогою алгебри поведінок. Спершу формалізуємо дії кожного агент-вузла в моделі. Семантика кожної дії визначається передумовою та післяумовою. Крім того дія

ілюструється процесною компонентою, яка важлива при моделюванні та отриманні трас-сценаріїв поведінки.

1. Дія створення блоку агентом-вузлом. Передумова визначає номер таймслоту, в який вузол може генерувати блок із відповідно зміною атрибутів BN та BM.

createBlock(i) = (TIME_SLOT == i) -><NewBlock> BN = BN + 1, Node(i).BB = BN+1,

2. Створення посилання на попередню висячу вершину в графі, яка знаходиться у ланцюжку максимальної довжини. Відповідно нова вершина після з'єднання стає висячою вершиною та помічається відповідним номером блоку. Довжина ланцюжка відповідно збільшується.

createReferenceFair(i) = 1 -><>
Node(i).REF_BLOCK(BN) = Node(i).HANG(Node(i).MAX);
Node(i).HANG(Node(i).MAX) = BN;
Node(i).LEN(Node(i).MAX) = Node(i).LEN(Node(i).MAX) + 1,

Згідно алгоритму консенсусу чесний вузол намагається продовжити найдовший ланцюжок для подальшої найшвидшої фіналізації.

3. Після отримання нової вершини в ланцюжку визначаємо нові ланцюжки із мінімальною та максимальною довжиною (якщо такі існують) в блокчейні агенту-вузла.

checkMin(i) = Forall (1<=j<= Node(i).CHAINS) && (Node(i).LEN(j) <Node(i).LEN(Node(i).MIN)) -><>Node(i).MIN = j,

checkMax(i) = Forall (1<=j<= Node(i).CHAINS) && (Node(i).LEN(j) >Node(i).LEN(Node(i).MAX)) -><>Node(i).MAX = j,

4. Надсилання блоку. Дана дія пересилає створений блок усім іншим агентам-вузлам.

sendBlock(i,b) = Forall (1<=j<=N && !(j==i)) ->< NODE(j): send(b, Node(i).REF_BLOCK(b)> 1,

5. Отримання блоку іншими вузлами.

receiveBlock(i) = 1 -><receive(x,y)>Node(i).REF_BLOCK(x) = y; Node(i).BC = x; Node(i).Forking,

6. Перевірка необхідності робити розгалуження.

```
checkForking(i) = Forall j, 1<=j<= Node(i).CHAINS && (Node(i).HANG(j) == Node(i).REF_BLOCK(Node(i).BC) -><> !(Node(i).Forking),
```

7. Створення розгалуження.

```
doForking(i) = Node(i).Forking -><>
Node(i).CHAINS = Node(i).CHAINS + 1;
Node(i).LEN(Node(i).CHAINS) = 1;
Node(i).HANG(Node(i).CHAINS + 1) = Node(i).BC(i),
```

8. Створення нової вершини у випадку відсутності розгалуження.

```
doHang(i) = ! Node(i).Forking -><>
Node(i).LEN(Node(i).CHAINS) = Node(i).LEN(Node(i).CHAINS) + 1;
Node(i).HANG(Node(i).CHAINS + 1) = Node(i).BC
```

9. Наступний тайм-слот.

```
nextTimeSlot = (TIME_SLOT <= N) -><> TIME_SLOT =  TIME_SLOT + 1
```

Далі опишемо поведінкові рівняння.

Маємо паралельну композицію агентів, що синхронізовані по значенню TIME_SLOT

```
B0 =((B(1) || B(2) || B(3) || B(4) || B(5));(nextTimeSlot.B0 + !nextTimeSlot.Delta)),
```

Поведінка кожного агенту протягом тайм-слоту складається із двох частин – створення блоку, якщо маємо відповідний тайм-слот, а також отримання блоків.

```
B(i) = (CreateMode(i);ReceiveMode(i)),
```

Створення блоку та відправка його іншим агентам-вузлам.

```
CreateMode(i) = createBlock(i).sendBlock(i),
```

Отримання блоків в циклі та відправлення затриманих блоків.

```
ReceiveMode(i) = ((receiveBlock(i).checkForking(i).(doForking(i) + doHang(i)));(checkMin(i).checkMax(i));(ReceiveMode(i) + sendBlock(BB(i),i));(ReceiveMode (i) + end(i))),
```

Створення блоку та створення посилання із перевіркою максимальних та мінімальних ланцюжків.

```
    CreateBlock(i)                                                    =
createBlock(i).((createReferenceFair(i).sendBlock(BN,i)               +
createReferenceFraud(i));(checkMin(i).checkMax(i)))
```

3.3. Моделювання зловмисника в блокчейн.

Зловмисник, на відміну від чесного користувача, буде мати іншу стратегію. Ті агенти-вузли, що контролюються зловмисникам, будуть намагатися створити альтернативну історію і продовжувати саме ті ланцюжки, які не є найдовшими. Введемо атрибут FRAUD, що буде визначати зловмисника. Перепишемо стратегію додавання блоку до ланцюжка зловмисником наступним чином.

```
createReferenceFraud(i) = Node(i).FRAUD -><>
Node(i).REF_BLOCK(BN) = Node(i).HANG(Node(i).MIN);
Node(i).HANG(Node(i).MIN) = BN;
Node(i).LEN( Node(i).MIN) = Node(i).LEN(Node(i).MIN) + 1,
```

Вставимо відповідно передумову !Node(i).FRAUD у дію чесного користувача createReference(i).

При даній стратегії фіналізація ланцюжка може наступити для історії, що контролюється зловмисником. Фіналізація, тобто вибір лише однієї історії із всіх альтернативних, залежно від алгоритму, може визначатись довжиною найдовшого ланцюжка або різниці в довжині від альтернативних. Намагання створити альтернативну історію зловмисником може викликати легалізацію *подвійної оплати*.

Розглянемо сценарій, при якому можлива подвійна оплата.

Нехай маємо множину агентів, серед яких існує певна кількість зловмисників. За правилами алгоритму консенсусу кожен із агентів при існуванні розгалуження у своєму блокчейні має продовжувати найдовший ланцюжок, тоді як зловмисник буде продовжувати той ланцюжок, який він контролює.

Сценарій подвійної оплати розпочинає агент, який платить транзакцію постачальнику, а потім знову ж ту саму криптовалюту платить іншому постачальнику. Транзакція вважається легітимною, якщо в блокчейні з’явився ланцюжок певної довжини (в оплаті біткойн – це 7 блоків). Це значить, що з великою ймовірністю наступить фіналізація цього ланцюжка. З іншої сторони, подвійну оплату легко виявлять при фіналізації блоків. Тому задача зловмисника якнайдовше не робити фіналізацію ланцюжків. За цей час, поки зловмисник утримує розгалуження в ланцюжку, отримувач криптовалюти може відпустити товар і коли наступить фіналізація, то кошти будуть ліквідовані в блокчейні. Така ситуація має бути у випадку, коли більшість агентів є зловмисниками. На рисунку показано ситуацію, коли зловмисники змагаються із чесними валідаторами по утриманню найдовшого ланцюжку. Чесні агенти мають номер 1 та 3, усі інші є зловмисники. Вже на другому кроці зловмисники затримують генерацію блока два, та відправляють його в наступний тайм-слот. Із-за цього відбувається розгалуження ланцюжку та третій номер продовжує ланцюжок від номера 1. Далі зловмисники продовжують по черзі перший та другий ланцюжок, намагаючись запобігти фіналізації.

Оскільки процес є повністю децентралізованим, то ніхто не знає, чи продовжується найдовший ланцюжок. Це можуть виявити агенти-валідатори, але вони також можуть бути зловмисниками. Таким чином, контролюючи всю мережу, зловмисник створює довгий ланцюжок, що є підтвердженням для отримувача транзакції, що транзакція легітимна, як це і буде при взаємодії чесних агентів.

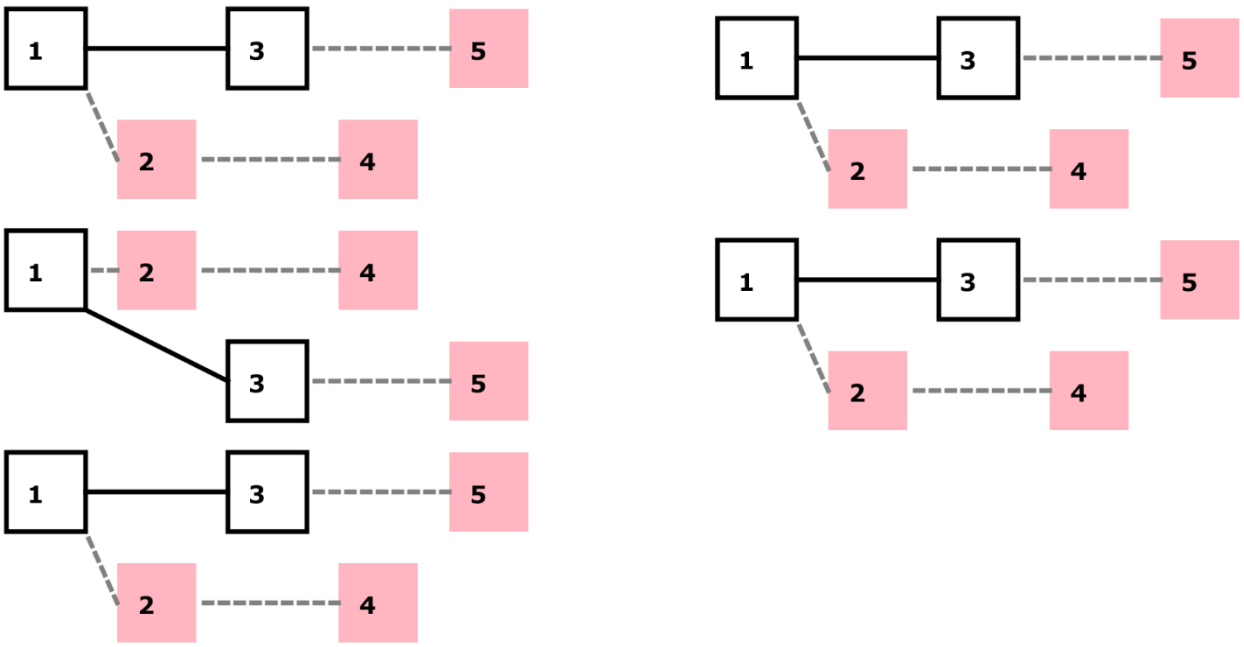


Рис. 3.4. Моделювання атаки «Подвійна оплата»

Більш детально формалізація блокчейн наведена в Додатку Б.

3.4. Інсерційна семантика розумних контрактів (Smart Contracts)

Розумні контракти в Solidity, що використовуються в логістичних системах на блокчейн платформах, є об'єктом атак зі сторони доступних дій в ньому. Кожен контракт на мові Solidity є середовищем, в якому взаємодіють агенти, які беруть участь в ланцюгу постачання. Взаємодія агентів відбувається за допомогою викликів публічних (public) функцій в розумних контрактах, якими агенти надсилають криптовалюту на баланс контрактів та можуть взаємодіяти із іншими контрактами. Відповідно інші агенти можуть реагувати на виклики функцій і таким чином поведінка розумного контракту є послідовність викликів функцій мови Solidity або дій, з точки зору інсерційної семантики. Фактично взаємодія агентів представлена паралельною композицією поведінок агентів:

a1 ||a2 ||…||an

де ai – поведінка i-го агенту. Ця поведінка визначається логікою коду розумного контракту, який теж виступає як агент. Розглянемо простий контракт в мові Solidity, який визначає дії інвесторів деякої компанії. Це може бути представлено декількома функціями.

```
contractSimpleDAO{
mapping(address => uint) public credit;
functiondonate(address to) {
credit[to] += msg.value;
}
functionwithdraw(uint amount) {
if(credit[msg.sender]>= amount) {
msg.sender.call.value(amount)();
      credit[msg.sender]-=amount;
   }
  }
```

Даний приклад детально представлено в роботах [1] та [2]. Розглянемо формалізацію даного розумного контракту в алгебрі поведінок.

Конструкція mapping може бути представлена, як атрибут агенту SImpleDAO. Це буде функціональний символ, який позначимо, як credit(x), де x є цілий числений атрибут довільної ємності.

Функція donate визначається реакцією на надсилання ефіру (криптовалюти) до розумного контракту та збільшення рахунку користувача на відповідну кількість токенів. Функція параметризована індексом користувача та кількістю надісланого ефіру. Запишемо її як алгебраїчну дію.

```
donate(x,value) = SimpleDAO:1 -><receive ether(x,value)>
User(x).money = User(x).money - value;
SimpleDAO.credit(x) = SimpleDAO.credit(x) + value,
```

Дія безумовна та визначається тригером, що є надходженням до контракту. Зміна кількості токенів вказана у післяумовах. Дія також містить процесну компоненту, що ілюструє процес отримання токенів, який не інтерпретується.

Функція withdraw, повернення інвестицій, може бути представлена двома діями:

```
withdraw1(x,value)   =   SimpleDAO:(SimpleDAO.credit(x)>=value)-
><call(withdraw,x,value)),send(x,value)>User(x).money = User(x).money
+ value
```

```
withdraw2(x, value) = SimpleDAO:1 -><>SimpleDAO.credit(x) = SimpleDAO.credit(x) - value
```

Поведінка взаємодії агенту SimpleDAO з агентами-користувачами User1(x, value)та User2(x, value)може бути представлена наступними рівняннями:

```
SP = User1(1, 10000) || User2(2, 10000),
User1(x, value) = donate(x, value).donate(x, value).withdraw1(x, value).fallback1(x, value).withdraw2(x, value),
User2(x, value) = donate(x, value).withdraw1(x, value).fallback2(x, value).withdraw2(x, value),
```

В прикладі показана схема за мінімально можливої кількості учасників, агент SimpleDAO виступає як фінансовий інститут та два користувачі, при цьому User1 є чесним користувачем, а User2 – зловмисником. На прикладі User1 вносить 2 рази якусь суму на рахунок, а знімає 1 раз. User2 ж, маючи тільки, наприклад, 10000 одиниць коштів, вносить їх один раз, а знімає 4 рази.

Поведінка fallback з’являється, згідно семантикою Solidity, після переказу коштів і, якщо вона не представлена в даному контракті, то може бути визначена одним із агентів серед користувачів.

fallback1(x, value) = empty,
fallback2(x, value) розглядається як довільна поведінка.

Більш детально формалізація смарт-контрактів наведена в Додатку В.

3.5. Формалізація атаки DAO для розумних контрактів.

Атака є поведінкою одного з агентів разом із впровадженням зловмисного контракту, який взаємодіє із поточним. В літературі існує деяка кількість прикладів, один із яких є атака DAO для контракту в мові Solidity. Ця атака експлуатувала семантику fallback функції і вразливості контракту при зніманні токенів із рахунку після виконання fallback функції.

Для проведення атаки було створено контракт, шо мав взаємодіяти із контрактом SimpleDAO. В цьому контракті були визначена поведінка виклику функцій, як самого агенту так і агенту SimpleDAO. Крім того була визначена функція fallback, яка рекурсивно викликала функцію withdraw (x,y). Таким чином,

загальна кількість кредитних коштів не зменшувалась, а загальна кількість токенів контракту перераховувалось на рахунок зловмисника.

```
fallback2(x, value) = withdraw1(x, value).withdraw1(x, value).withdraw1(x, value)
```

Сама атака визначалася наступними діями зловмисника:

1. Відсилання 1 ефіру на рахунок контракту SimpleDAO функцією donate;

2. Зняття одного ефіру із рахунку;

3. Виклик fallback функції зловмисника, породжує нескінченне знімання ефіру із гаманця користувача.

Дана атака використовувала ту властивість, що fallback функції може визначатись іншим користувачем і сам контракт мав відповідну вразливість.

3.6. Запобігання атакам при проектуванні блокчейн системи

Одним із методів виявлення потенційних вразливостей в коді або моделі системи є метод алгебраїчного співставлення моделі поведінки системи із шаблоном атаки. На відміну від відомих методів співставлення шаблонів, наприклад вірусів, даний метод аналізує в динаміці поведінку вирішенням поведінкових рівнянь.

В загальному вигляді метод визначається наступним алгоритмом.

Нехай дано систему рівнянь в алгебрі поведінок.

B0 = B1 || B2 || … || Bm

$Bi = a_i^1.Bi^1 + a_i^2.Bi^2 + \ldots + a_i^n.Bi^{ni}$

$Bi^j = a_i^{j1}.Bi^{j1} + a_i^{j2}.Bi^{j2} + \ldots a_i^{j1}.Bi_j^{nij}$

Дана система рівнянь представлена в канонічному вигляді, тобто кожне рівняння є сумою поведінок, поданих у вигляді префіксингу дії та поведінки:

$$B = \sum_{i=1}^{n} a_i . B_i$$

Задано шаблон деякої поведінки.

$X = a_{i1}.a_{i2}.Y.a_{ik}$

де Y – довільна поведінка в термінах визначених дій. Дії можуть відноситись до різних агентів.

Необхідно знайти множину сценаріїв поведінки, заданою системою поведінкових рівнянь, що відповідають даному шаблону або досяжні із початкової поведінки. Сценарій поведінки є послідовність дій, які досяжні в поведінці.

Задача розділяється на дві підзадачі.

1. Знайти послідовність дій, що відповідає даному шаблону, тобто в даному випадку. Дана задача зводиться до вирішення поведінкових рівнянь. Рішенням рівняння буде множина сценаріїв поведінок, що відповідають шаблону: $Z.a_{i1}.a_{i2}.Y.a_{ik}$ або множина сценаріїв поведінок, що починаються із початкової дії початкової поведінки та веде в поведінку a_{ik} через поведінки $a_{i1}a_{i2}$. Задача може визначати скільки таких сценаріїв поведінки необхідно знайти. В багатьох задачах їх може бути нескінченна кількість, тому знаходять один сценарій або множину сценаріїв, що покривають систему поведінкових рівнянь. Рівняння вирішується за допомогу співставлення методом переписування. Метод вирішення такого рівняння реалізовано в системі алгебраїчного програмування (АПС).

2. Знайдений сценарій може бути недосяжний. Якщо працюємо в атрибутному середовищі, то маємо брати до уваги семантику дії. Дія може бути недосяжна в зв’язку з тим, що не існує атрибутів, що роблять можливим існування такого сценарію. Таким чином необхідно довести досяжність сценарію за допомогою символьного моделювання. В кожному шаблоні в діях може бути задано перед- та післяумова. При моделюванні ми перевіряємо алгебраїчне співставлення середовища моделювання із передумовою дії в шаблоні.

Нехай задано середовище *E* на деякому кроці моделювання, який співставлений із дією в якій передумова є предикат *P*. Тоді дія досяжна в сценарії, якщо формула $E \wedge P$ – виконувана, тобто існують такі значення атрибутів, при яких формула істинна.

Шаблон можливості атаки визначає дії зловмисника, поведінка якого не відповідає поведінці консенсусу. Наприклад у випадку подвійної оплати в шаблоні може бути умова, що зловмисник подовжує не максимальний ланцюжок, а коротший. Шаблон складається із однієї дії, що порушує правило консенсусу.

Шаблон успішної атаки визначає дії зловмисника, при яких досяжна атака. Одним із прикладом є поведінка зловмисника, коли він порушує правило консенсусу для того, щоб отримати нефіналізований ланцюг довжиною X, що перевищує кількість створених блоків від транзакції, що вважається достатньою для того, щоб постачальник зарахував транзакцію. Постачальник розблоковує поставку товару, маючи підтвердження тільки кількістю блоків від транзакції прийому криптовалюти. При фіналізації та верифікації блоків встановлюється, що в пулі транзакцій є дві однакових транзакції та одну із них не зараховують та списують криптовалюту із рахунку постачальника. Але товар може вже бути відправлено в іншу країну із валідними документами, де він буде недосяжний.

При проектуванні систем на основі блокчейну необхідно провести моделювання такої атаки та зрозуміти її ймовірність. Зловмисник може з’явитись в будь-якій блокчейн системі та за можливості атакувати мережу і мережа може в процесі роботи розпізнати такі дії, якщо є відповідний моніторинг. Але оцінити при яких умовах атака буде успішна необхідно при модельній розробці. Наприклад, атака подвійної оплати може успішно закінчитись, якщо в мережі знаходиться 51% процент шкідливих вузлів.

Для атаки подвійної оплати шаблон є наступним.

X = a1.a2, де a1 – отримання блоку, a2 – подовження блокчейну згідно посилання на попередній блок ланцюжку. a2 = (REF = HANG(i)) & (MAX != LENGTH(i)) -> 1. Тобто в передумові шаблону є формула,що вказує нате, що попередній блок, з яким зв’язується в ланцюжку поточний новий блок, не належить до ланцюжка із максимальною довжиною.

Умова успішної атаки Exist i,j , LENGTH(i)>=LENGTH_CONFIRMED && LENGTH(j)>=LENGTH_CONFIRMED, тобто побудовані такі два ланцюжки,

довжина яких більше довжини ланцюжку від транзакції, що необхідна для її підтвердження. Наприклад для біткоін така довжина дорівнює 7.

При проектуванні розумних контрактів визначаються норми розробки (policies), що включають запобігання таких випадків, як DAO-атаки та інші. Але автоматичне виявлення вразливостей розумних контрактів згідно шаблонів відомих атак також необхідне при модельній розробці. Для виявлення поведінки зловмисника необхідно відповісти на питання «Чи містить розумний контракт шаблон можливої атаки?»

Маючи модель паралельної композиції розумних контрактів, також використовується метод співставлення із шаблоном. Розглянемо шаблон атаки DAO.

```
DAOAttackBehavior = MaliciousCycle,
MaliciousCycle  = withdraw1(x,amount). MaliciousCycle
```

При вирішенні поведінкового рівняння для паралельної композиції контрактів ми можемо знайти послідовність дій, що приводить до цього шаблону. Досяжність визначається умовою:

```
(OLD_CREDIT == credit(x)) && (credit(x) >= amount))
```

Ця умова перевіряється при будь-якій відправці криптовалюти і якщо ми щось пересилаємо, а загальна кількість криптовалюти не змінюється, то це може мати ознаки атаки DAO. Виявлення можливостей атаки DAO з огляду на довільні дії користувачів відносно функції fallback є однією із задач верифікації на етапах модельної розробки.

3.7. Запобігання атакам при функціонуванні блокчейн платформи

Розглянемо засоби виявлення кібератак, які не належать до стадії модельної розробки системи, але важливі при її експлуатації. При функціонуванні блокчейн платформи моніторинг може виявити ознаки атаки згідно шаблону. Для шаблонів атак, що визначаються однією чи двома діями виявляти ознаку атаки вже при виконанні дій може бути запізно. При чому ознаки, що визначені шаблоном,

можуть відноситись і до нормальної поведінки користувача. Виявлення атаки відбувається також співставленням дій або транзакцій, що проходять при роботі системи та вирішення виконуваності умови ознак атаки. Для визначення ознак атаки використовують моделі класифікації сценаріїв, що отримані за допомогою машинного навчання.

Використання машинного навчання розділяється на два етапи. Перший – це моделювання атак на блокчейн та отримання даних для тренування нейронної мережі. Другий етап – це вже використання нейронної мережі, як моделі класифікації. Нормальний потік даних буде класифікуватись як звичайна робота користувача, а відхилення, відповідно, як ознака атаки. У випадку машинного навчання ознаки атак можуть бути виявлені до факту закінчення атак та прийняті відповідні міри, які мають бути враховані на етапі модельної розробки.

Моделі машинного навчання іноді дають похибку, тому атака може бути несправжньою, а виникати внаслідок природних аномалій у мережі. Для підтвердження атаки необхідно використовувати алгебраїчні умови та алгебраїчні шаблони вразливостей та атак.

Гібридні методи моніторингу функціонування блокчейн платформи комбінують використання машинного навчання та методи алгебраїчного співставлення. Якщо при виявленні моделлю класифікації підозрілої поведінки почати співставлення із можливим алгебраїчним шаблоном, то можливе більш точне виявлення факту атаки. Треба мати на увазі, що гарантовано атака розпізнається при повному співставленні із алгебраїчним шаблоном.

Висновки до Розділу 3.

Блокчейн технології в системах логістики використовуються з метою підвищення безпеки функціонування системи та протидії атакам, а також із метою підтримки цілісності та актуальності даних. Блокчейн технології забезпечують високий рівень захисту та надійності з точки зору крадіжки або підміни даних, ідентифікації. Мережевий криптозахист забезпечує високу ступінь безпеки на рівні

неавторизованого проникнення. Однак є ряд атак, які треба виявляти при проектуванні систем, наприклад аналіз можливих атак та аналіз відомих вразливостей, а також при функціонуванні систем, де відслідковується поведінка агентів. При проектуванні системи необхідно оцінити ймовірність атаки та забезпечити стійкість системи за допомогою відповідного алгоритму консенсусу та верифікації розумних контрактів. Алгоритм консенсусу визначає правила поведінки агентів у децентралізованій системі, а розумні контракти визначають індивідуальні дії кожного з агентів при визначенні в системі ланцюжку постачання. Коли в глобальній мережі постачання з'являються нові агенти, то вони обов'язково повинні пройти процедуру верифікації на відповідність головним наборам властивостей, що визначають безпеку. При виконанні смарт-контрактів атака є поведінкою одного з агентів разом із впровадженням зловмисного контракту, який взаємодіє із поточним в мові Solidity.

Одним із методів виявлення потенційних вразливостей в коді або моделі системи є метод алгебраїчного співставлення моделі поведінки системи із шаблоном атаки. На відміну від відомих методів співставлення шаблонів, наприклад вірусів, даний метод аналізує в динаміці поведінку вирішенням поведінкових рівнянь. Задача полягає в знаходженні множини сценаріїв поведінки, заданою системою поведінкових рівнянь, що відповідають даному шаблону або досяжні із початкової поведінки. Задача полягає в знаходженні послідовності дій, що відповідає даному шаблону та перевірці алгебраїчного співставлення середовища моделювання із передумовою дії в шаблоні. Шаблон можливості атаки визначає дії зловмисника, поведінка якого не відповідає поведінці консенсусу. Шаблон успішної атаки визначає дії зловмисника, при яких досяжна атака, тому задача при розробці системи на основі блокчейну – провести моделювання такої атаки та зрозуміти її ймовірність, а оцінити при яких умовах атака буде успішна необхідно при модельній розробці. При проектуванні розумних контрактів визначаються норми розробки, що включають запобігання випадків атак, але автоматичне виявлення вразливостей розумних контрактів згідно шаблонів відомих атак також необхідне при модельній розробці. Маючи модель паралельної

композиції розумних контрактів, також використовується метод співставлення із шаблоном. При вирішенні поведінкового рівняння для паралельної композиції контрактів ми можемо знайти послідовність дій, що приводить до цього шаблону.

Виявляти ознаки кібератак потрібно продовжувати після стадії модельної розробки системи, вже в процесі експлуатації блокчейн платформи. Для визначення ознак атаки використовують моделі класифікації сценаріїв, що отримані за допомогою машинного навчання. Для цього моделюють атаки на блокчейн та в подальшому використовують отримані дані для тренування нейронної мережі. Далі нейронної мережі використовують, як моделі класифікації. Нормальний потік даних буде класифікуватись як звичайна робота користувача, а відхилення, відповідно, як ознака атаки. Отримані результати враховуються на етапі модельної розробки.

Гібридні методи моніторингу функціонування блокчейн платформи комбінують використання машинного навчання та методи алгебраїчного співставлення.

Список використаних джерел до Розділу 3.

1. A. Letichevsky, O. Letychevskyi, and V. Peschanenko, “Insertion Modeling and Its Applications,”Computer Science Journal of Moldova, vol. 24, no. 3, 2016, pp. 357-370.

2. A. Letichevsky and D. Gilbert, “Interaction of agents and environments,”inRecent Trends in Algebraic Development Technique, LNCS 1827,Springer-Verlag, 1999.

РОЗДІЛ 4.
ВИКОРИСТАННЯ МОДЕЛЬНОГО СПОСОБУ В РОЗРОБЦІ ІНДУСТРІАЛЬНИХ ЛОГІСТИЧНИХ СИСТЕМ

В даному розділі висвітлюється два приклади використання модельного методу розробки. Перший – це приклад логістичної системи фермерського господарства, яка використовує моделювання на рівні моделей вимог та другий - приклад формалізації вимог та подальша розробка для логістичної системи, що визначає та супроводжує функціонування ланцюжка постачання.

4.1. Логістична система на базі фермерського господарства.

Для перевірки практичного застосування теоретичного матеріалу, викладеного в даній роботі, було створено діючу логістичну систему на базі фермерського господарства «Надія», що знаходиться в селі Марійка, Жашківського району Черкаської області та ПрАТ «Ксібекс», яка є міжнародним логістичним оператором та суб'єктом міжнародної торгівлі. ФГ «Надія» вирощує сою на власних полях, використовуючи власну сільськогосподарську техніку, та переробляє її в соєву олію на власній лінії переробки. Дану закриту логістичну систему обслуговуючого типу використано для створення діючої системи управління фермерським господарством та для розробки методології технології модельної розробки та системи моделювання для подібних закритих логістичних обслуговуючих систем. ФГ «Надія» виробляє готову експортну продукцію, тобто є першим учасником в міжнародному ланцюжку постачання (supplychain). ПрАТ «Ксібекс», отримавши у розпорядження продукцію від ФГ «Надія», стає агентом-постачальником та оператором у цьому міжнародному ланцюжку постачання, так як дана продукція експортується до Польщі наземним транспортом (мономодальне перевезення без перевантажень) та до Китаю (морське та мультимодальне перевезення з багатьма послідовними учасниками логістичного процесу з процесом передачі продукції на етапах перевантаження). Тому даний підхід та методологія побудови управлінських систем та системи моделювання випробовується від

моменту створення продукції в закритій логістичній системі обслуговуючого типу аж до кінцевого споживача через увесь шлях в відкритій міжнародній логістичній системі.

4.1.1. Концепція системи управління фермерським господарством. Призначення системи

Сучасні технології управління системою фермерського господарством, як і будь-якою іншою закритою системою, вирішуються сьогодні на основі різноманітних підходів, але мають у своїй основі системний підхід. Це дозволяє розглянути цілісний процес управління підприємством як сукупність пов’язаних функціональних задач – поточний контроль місцезнаходження, стану та функціональної задачі кожного елементу системи, а також планування їх взаємодії та резервування ресурсів у разі переходу якихось агентів в стан непрацездатності [1].

Система функціонує як фреймворк з нарощуваною модульною структурою відповідно до вирішуваних задач. Тобто центром системи (ядром) виступає централізована база даних (ЦБД), яка служить для збору, обробки, зберігання та обміну інформації, яку вона приймає, та яку передає агентам цієї системи. При цьому цих агентів змінна кількість, вони можуть підключатись до системи. Також деяка кількість постійних агентів знаходиться у стані неактивності або простою.

В якості ядра системи управління використано технологію гео-інформаційних систем (ГІС), оскільки майже вся інформація про стан сільськогосподарського підприємства характеризується просторовою прив’язкою або безпосередньо, або опосередковано. Виходячи з цього, якщо використовувати у якості ядра ГІС-систему безпосередньо, а не розділяти інформаційні потоки на дві складові та, відповідно, обробляти їх окремими підсистемами, оперативність обробки та функціональні характеристики можна покращити.

З метою вирішення задач підтримки надання рішень та визначення завдань для агентів, які взаємодіють всередині системи, та побудови сценаріїв виконання робіт та операцій, побудови прогнозної інформації, система містить відповідні

моделі та супровідні дані, а також інтерфейсні засоби взаємодії з цими моделями. Канали зв'язку системи є резервованими як логічно, так і фізично, для забезпечення цілісності та достовірності даних.

Система виступає комунікаційною платформою підприємства, яка забезпечує надійний обмін даними по технологічним операціям та організаційним і управлінським рішенням.

Враховуючи фактори застосування обчислювальних та комунікаційних пристроїв у різноманітних задачах, апаратна та програмна платформа створена гнучкою та різноманітною. Але при цьому ядро обчислювальної системи є уніфікованим для усієї лінійки пристроїв.

Функціонування системи опирається на такі інструменти, що допомагають вирішити типові задачі для фермерського господарства:

- визначити розташування ділянок та паїв,
- організувати оптимальну логістичну інфраструктуру у межах господарства,
- здійснювати оперативний моніторинг рухомих об'єктів та облік їх роботи;
- здійснювати оперативну інформаційну підтримку для операторів на рухомих об’єктах;
- здійснювати моніторинг стану ділянок на поточний час – де та які роботи виконано, чи виконуються, які технологічні операції передбачено до виконання, які ресурси потрібні для їх виконання та таке інше;
- організація ефективної комунікаційної системи господарства що дозволяє не тільки отримувати інформацію, але й здійснювати зворотній зв'язок.

Поки що неможливо створити організацію повністю автоматичної роботи системи через специфічні фактори сільського господарства та велике значення «людського ресурсу» та «людського фактору».

У ієрархії підприємства її агенти займають різні рівні підпорядкування, від керівних, які визначають відповідний час та готовність до початку виконання дії іншими агентами, до виконавчих, які безпосередньо виконують певну роботу за

умови команди, та обслуговуючих, які відповідають за безперервність та стабільність виконання роботи головними агентами-виконавцями. Ця структура визначає хто кому підпорядкований та чиї накази повинен виконувати, а також кому повинен доповідати про виконання чи невиконання розпоряджень та наказів.

Отже роботи по розробці системи управління логістичною системою фермерського господарства проведено у кількох основних напрямках:

- розробка компонентів для отримання, реєстрації та візуалізації гео-інформаційної та навігаційної інформації;
- розробка компонентів інформаційно-комунікаційної мережі;
- розробка компонентів ядра системи.

В структурі системи управління фермерським господарством, ядро системи вирішує задачі з реалізації логіки роботи системи, управління потоками інформації, її реєстрації, та зберігання, а основні функції, які підлягають реалізації, це:

- отримання інформації від її джерела;
- реєстрація параметрів, які характеризують інформаційне повідомлення та стан джерела;
- доставка інформації до отримувача;
- реєстрація параметрів, які характеризують стан отримувача;
- реєстрація параметрів обміну інформацією.

Вимоги до системибулопредставлено у формальному вигляді за допомогою специфікацій алгебри поведінок. Зразки специфікацій вимог подані в додатку 1 до даної роботи.

4.1.2. Методи реалізації функціональних компонентів системи

Для впровадження модельного методу в розробці застосовувася дизайн специфікацій компонент на мові діаграм, що транслювались в мову специфікацій для подальшої верифікації та відповідності вимогам.

З метою зменшення складності структури комунікаційної складової вирішено використовувати системи динамічної IP адресації для рухомих складових системи, що зробило необхідним введення додаткової інформації для логічної

ідентифікації об'єктів. Така організація системи є найбільш гнучкою, та вимагає фіксованих IP адрес тільки для компонентів які входять до складу ядра системи управління. Логіка роботи комунікаційної підсистеми у системі управління розуміє потребу реєстрації подій з виникнення та доставки інформації від відповідного об'єкту чи суб’єкту до іншого відповідного об'єкта чи суб’єкта.

Для реалізації системи її побудовано на основі клієнт-серверних технологій з WEB-сервером або на базі мережевого сервісу з метою уніфікації доступу та інтерфейсів з віддаленим доступом на основі динамічної генерації контенту. Реєстрація повідомлень виконується серверною частиною, яка збирає повідомлення для клієнтів у проміжки часу між сеансами зв'язку, виконує процес ідентифікації користувача при підключенні до системи, передає йому накопичені повідомлення та приймає повідомлення від нього.

Аналіз специфічних факторів роботи агентів середовища у сільському господарстві визначив наступні можливі рішення для архітектури фізичних каналів передачі та обміну інформації:

- наявність вібрації вимагає конструктивних рішень, які полягають в тім, що конструкція повинна бути невеликих розмірів, жорсткою та з мінімально можливою кількістю механічних з'єднань та контактів;
- вплив пилу та вологи вказує на необхідність конструкції майже закритої, без великих отворів;
- попередній фактор та великий температурний діапазон вказує на необхідність використання інтегральних схем у промисловому виконанні;
- коливання напруги у системі живлення вказують на потребу додаткових стабілізаторів напруги та току.

Процес обміну даними потребує реалізації з двох елементів – програмного забезпечення, яке виконує функції з обробки даних та управління логікою процесів, та апаратного забезпечення, найважливішою з функцій якого є створення лінії фізичного зв'язку.

Для реалізації інструментів фізичного зв’язку, передачі та обміну інформацією всередині системи було використано структуру, наведену на рис.4.1.

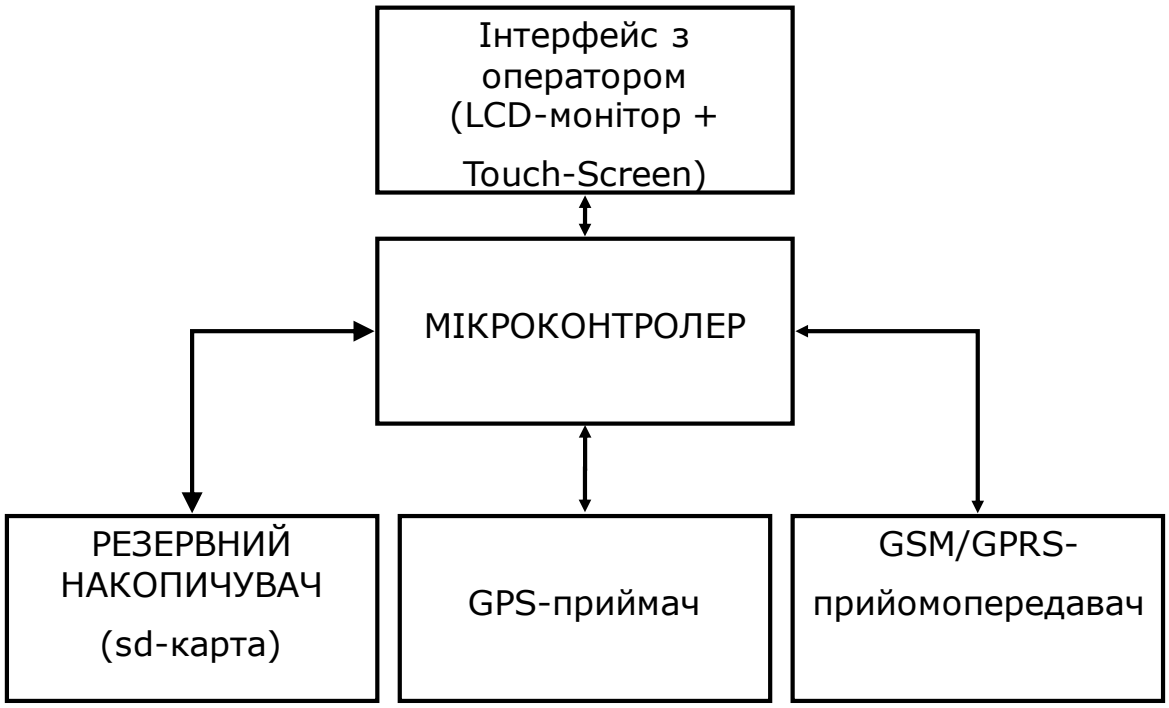


Рис. 4.1. Структура макету системи

Даний дизайн представляє собою модель, що включає в собі 4-х агентів, що взаємодіють із зовнішнім середовищем через інтерфейс. В мові інсерційних моделей маємо наступну структуру:

- Опис середовища, що включає чотирьох агентів Microcontoler, SD-card, GPS-receiver, GSM/GPRS-receiver;
- Дії, що описують обмін сигналами для Microcontoler-SD-card,Microcontoler-GPS-receiver, Microcontoler-GSM/GPRS-receiver.
- Поведінка представляє собою паралельну композицію агентів.

Для проектування моделі зв’язку використано програмне забезпечення PacketTracer від компанії Cisco. Це дозволило спроектувати топологію зв’язку за вимогами користувача, та промоделювати їх роботу. Модель зв’язку представлена на рис. 4.2.

Організація виходу до Інтернету через маршрутизатор надає серверу відповідності до вимог безпеки, бо, окрім керування процесом маршрутизації, роутер виступає у якості брандмауеру. Це забезпечує первинний захист від втручання у роботу зловмисників.

Сервер бази даних розташовується на окремій машині від веб-серверу. Прямий доступ з глобальної мережі інтернет є тільки до веб-серверу. Інші ж пристрої що знаходяться у локальній мережі, що підключені до маршрутизатора є невидимими зовні.

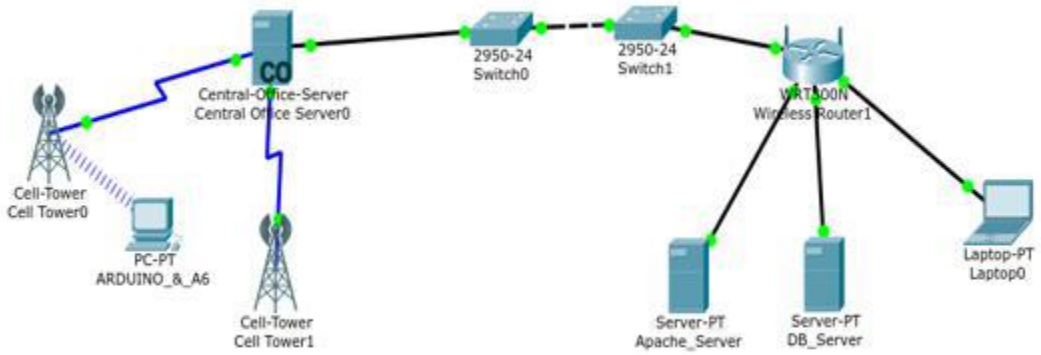


Рис. 4.2. Схема організації лінії зв'язку у системі.

Віртуальний сервер використовується для налаштування сервісів загального користування у локальній мережі. Віртуальний сервер визначається як порт сервісу, після чого запити, що приходять із глобальної мережі на цей порт, перенаправляються на визначений комп’ютер у локальній мережі. Любий комп’ютер у локальній мережі з використанням віртуальних серверів повинен мати статичну IP адресу. Прямий доступу до серверу бази даних із глобальної мережі відсутній. Але доступ до нього можливий із серверу та інших локальних пристроїв.

Таким чином в організації зв’язку створювались агенти типів Cell-Power, Switch, Server, Laptop. Кожен тип мав деяку кількість екземплярів агентів, що взаємодіяли між собою. Поведінка всієї системи представляє собою паралельну композицію взаємодіючих агентів. Обмін сигналами формалізується в діях, параметризованими екземплярами агентів.

Для верифікації властивостей таких агентів використовується символьне моделювання із застосування алгоритмів видалення *інтерлівінгу*, який визначається залежністю дій. Більшість дій агентів незалежні, тому видалення

інтерлівінгу в даному випадку полягає в ігноруванні порядку приходу сигналів в незалежних діях.

Однією із специфік перевірки властивостей та тестування є використання шарів (slices) даних, що скорочує час тестування та верифікації.

Для тестування використна модель дизайну, з якої було згенеровано тести.

4.1.3. Методи реалізації інформаційно-комунікаційного середовища та ядра системи

Логіка роботи системи з вирішення комунікаційної задачі полягає в тому, що її агенти виконують наступні дії у разі виникнення нової інформації:

- джерело інформації генерує повідомлення, заповнюючи усі необхідні поля;
- джерело інформації генерує запит до сервера ядра системи управління системою;
- серверна частина приймає запит та реєструє дані про джерело, отримувача та характеристики інформаційного повідомлення;
- серверна частина генерує значення для поточного стану, виходячи із характеристик повідомлення.

Кожен агентна початку роботи звертається до системи (реєстрація на точці призначення, генерація повідомлення про перехід до стану готовності, реєстрація поточноїгеолокації) сервер виконує пошук повідомлень у базі даних, які надійшли до системи для цього агенту з часу його останнього звернення. Якщо такі записи було знайдено, сервер виконує наступне:

- передає повідомлення на пристрій агента;
- реєструє у запису повідомлення дату та час передачі одержувачу;
- реєструє у запису повідомлення місце розташування одержувача;
- змінює статус повідомлення на стан «отримано».

Така процедура гарантує доведення усіх повідомлень у системі до одержувачів та фіксує усі необхідні дані для реєстрації дій відповідних агентів у певних ситуаціях.

Згідно з правилами обігу інформаціїпри кожному відправленні або отриманні повідомлення має бути зафіксовано:

- відправника (ідентифікатор користувача, ціле число, integer);
- дата та час відправлення (тип дата+час - timestamp);
- місце знаходження відправника (координати як набір полів, чи);
- отримувача (ідентифікатор користувача, ціле число, integer);
- дата та час отримання (тип дата+час - timestamp);
- зміст інформації (текстове поле - text);
- відповідь (ідентифікатор запису з цієї ж таблиці);
- тип повідомлення (ціле число, integer).

Поле «Тип повідомлення» введено для ідентифікації повідомлень за типом відповідної реакції на це повідомлення. Так, наприклад, можна вказати на наступні ситуації:

- повідомлення є інформаційним і не потребує відповіді чи виконання якихось робіт;
- повідомлення потребує відповіді;
- повідомлення є розпорядним до виконання робіт;
- повідомлення про поточний стан робіт (поточний звіт);
- повідомлення для встановлення нової точки призначення для мобільних об'єктів, які використовуються у системах навігації.

Загальна структура інформаційних зв’язків може бути представлена діаграмою, представленою на рис. 4.3.

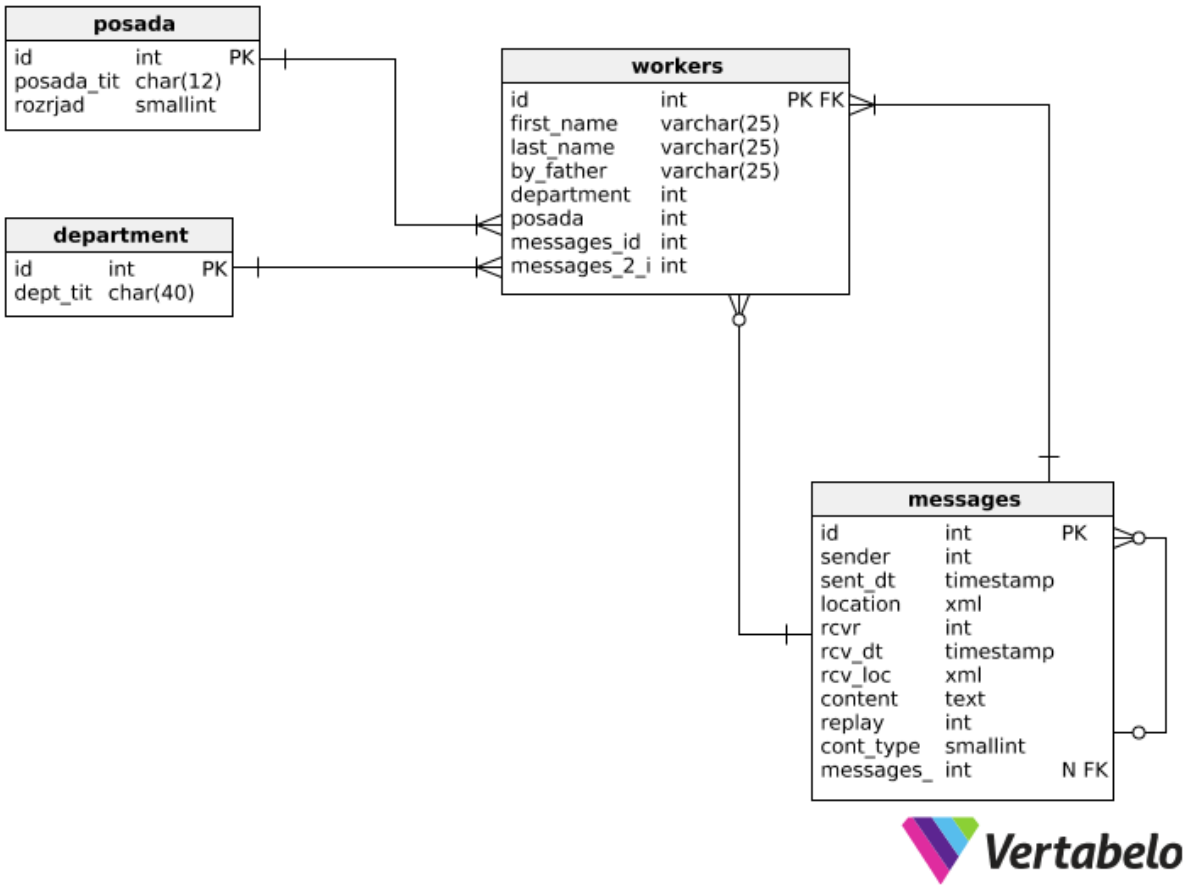


Рис. 4.3. Структура зв’язків у базі даних інформаційно-комунікаційного середовища.

4.1.4. Методи реалізації програмного забезпечення геоінформаційної підсистеми

Головною задачею підсистеми є знаходження місця розташування об'єкту моніторингу на місцевості та передачі отриманих даних до системи управління фермерським господарством. Таким чином, в структурі програмних засобів виділяють програмне забезпечення мобільної системи та серверне програмне забезпечення.

Серверне програмне забезпечення має працювати 24 години на добу та виконує, відповідно до розглянутої задачі, прийом даних від мобільної системи, реєстрацію отриманих даних, організацію інтерфейсу для виведення графічної інформації користувачам системи про поточний стан системи та розташування об'єктів моніторингу.

Загальна послідовність дій, яка виконується мікроконтролерною мобільною системою у процесі визначення положення об'єкту у просторі та їх реєстрації наведена на рис.4.4.

Рисунок 4.4. Послідовність дій у роботі контролера.

За відправку даних на сервер відповідає частина коду, котра циклічно повторюється кожні десять секунд, що дозволяє досить оперативно контролювати розташування мобільної системи.

Збір даних полягає у створенні запитів до GPS приймача з використанням протоколу NMEA та отриманні від нього певної відповіді, яка приходить у форматі текстової строки, з якої потрібно виділити потрібні дані. Для цього було створено ряд сервісних процедур та реалізовано відповідний парсер.

Після того, як ці дані отримано, їх подальша обробка полягає у кількох напрямках – реєстрації на сервері, аналізу для виконання навігаційних функцій апаратурою мобільної системи та відображенню користувачеві для оперативного інформування.

Основою серверної частини є відповідна база даних, яка формується по кожному агенту системи. Структура бази даних підсистеми наведена на рис. 4.5.

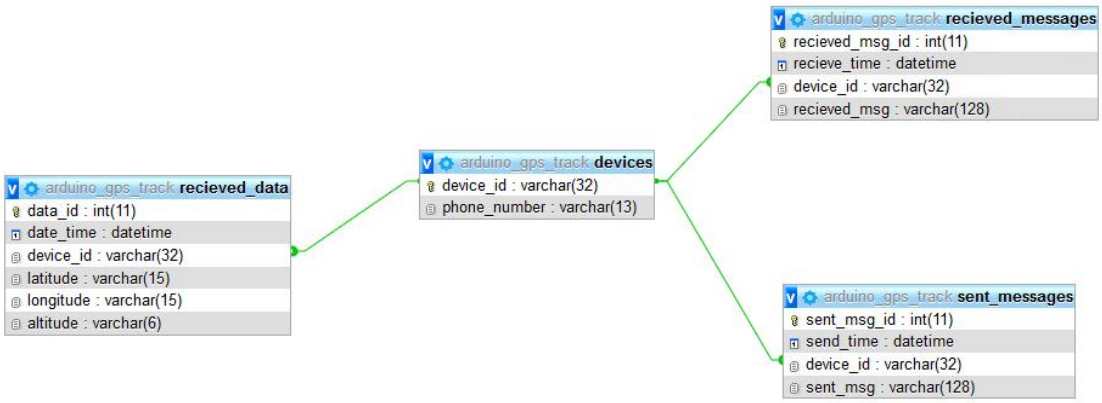


Рис. 4.5. Структура БД з реєстрації поточного стану об'єкту моніторингу.

Таким чином, база даних містить чотири таблиці:

- devices – комбінація з ідентифікатора плати та комутатора агента;
- received_data – отримані від пристрою координати, час та ідентифікатор пристрою;
- sent_messages – містить повідомлення, які передано на мобільний пристрій через серверну частину;
- received_messages – містить повідомлення, які отримано сервером з мобільного пристрою.

Структура модулів серверної части наведена на Рис.4.6.

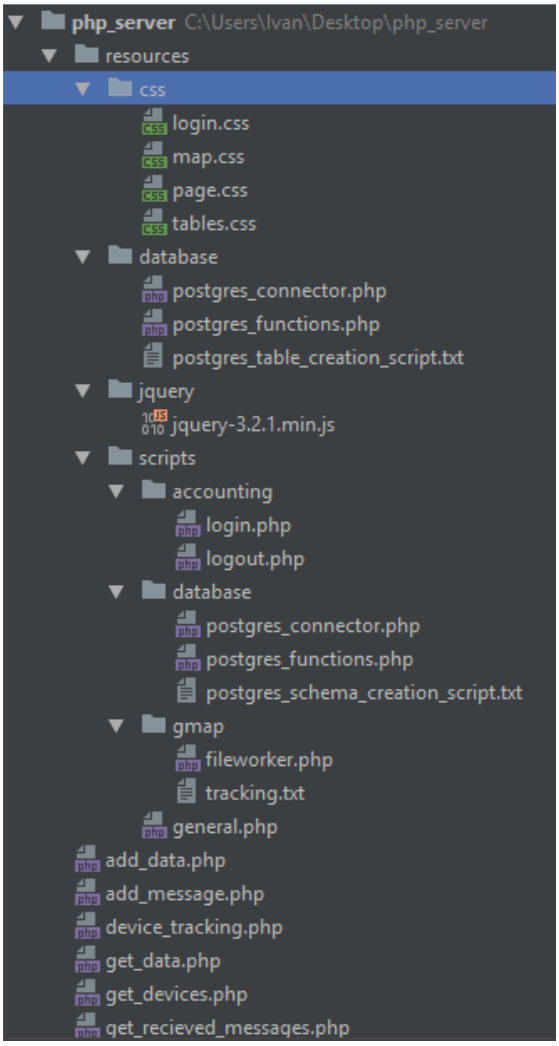


Рис. 4.6. Структура модулів серверної частини

На прикладі нашого конкретного фермерського господарства та створеної на її базі системи процес усіх робіт починається з аналізу стану земельній ділянці та готовності урожаю до збирання. Тобто стартом до початку робіт являється присвоєння певному полю статусу готовності агрономом господарства. Така інформація надається за допомогою геоінформаційної системи, яка входить до комплексу автоматизованого управління. Таким чином, інтерфейс системи агронома повинен передбачати можливості відображення картографічної інформації за допомогою ГІС-складової та можливості організації роботи з аграрною моделлю та відповідними базами даних.

У процесі виконання робіт, повинна надаватися можливість поточного стану земельних ділянок та виконуваних на них роботах. Тобто, у режимі реального часу

відображається інформація про техніку, яка знаходиться у полі, історія її пересування, відсоток виконаної роботи та таке інше.

Після команди агронома про готовність поля та початку роботи всіх агентів система безперервно організовує збір інформації, яка характеризує наступні задачі:

- технічний стан транспортного засобу;
- інформація про поточне географічне положення транспортного засобу;
- груба навігаційна інформація у разі виконання загальних транспортних задач;
- навігаційна інформація з високою розподільчою здатністю для виконання аграрних технологічних операцій у задачах PrecisionAgriculture;
- комунікаційні задачі оператора-водія з адміністраторами робіт.

4.1.5. Методи реалізації інтерфейсу навігаційної системи

Основні вимоги до інтерфейсу навігаційної системи полягають у відображенні поточного положення об'єкту у світовій системі координат та відносно якихось інших об'єктів.

Відповідно до цього, передбачається два елементи, які виводять поточні координати на екран у цифровій формі. Але, оскільки у цифровій формі положення у просторі сприймається погано, потрібно передбачити й графічний режим виводу, у якому відображається карта та у вигляді умовного символу положення об'єкту.

Окрім виводу поточного положення, у задачах навігації часто потрібно виводити також точку кінцевого призначення та маршрут.

Особливістю функціонування у цьому режимі є потреба виводу зображення карти, яке відображається у растровому форматі. Ця необхідність може бути реалізована двома шляхами:

- отримання картографічних даних у векторному форматі та растерізація їх у контролері системи, що потребує значних обчислювальних затрат;
- отримання зображення карти від системи управління фермерським господарством з його завантаженням спочатку у контролер через

комунікаційні канали, а потім з контролеру до системи монітору, що дає велике навантаження на комунікаційні канали.

Інший шлях може полягати в тім, що на етапі до проведення польових робіт у контролер завантажуються відповідні картографічні зображення, які потім використовуються контролером у процесі роботи.

Інша особливість роботи навігаційної системи полягає у тім, як буде проводитися розрахунок траєкторії руху об'єкта з поточного положення у кінцеву точку призначення. При цьому, розрахований шлях подається у вигляді сукупності координат проміжних точок, у яких слід змінювати напрямок руху. У цьому процесі також рішення можуть бути знайдені або на стороні контролеру, або на стороні серверу основної системи управління фермерським господарством, яка потім по каналам зв'язку їх передає до контролеру, який вже потім керує як процесом управляння засобом, так і процесом відображення отриманої траєкторії на екрані монітору.

Із допоміжних засобів, які використовуються у задачах навігації, ще слід зазначити компас, спідометр та підказки для підтримання напряму руху. Взагалі, сучасні GPS-приймачі видають цю інформацію у систему, оскільки мають у своєму складі магнітометри та акселерометри.

Виходячи із зазначеного, пропонується реалізація 2-х режимів роботи інтерфейсу навігаційної системи – повністю числового та графічно-числового, зовнішній вид яких представлено на рис.4.7 та 4.8, та для переходу між якими використано додаткові елементи управління.

Основною особливістю процесу обміну між TFT-монітором та мікроконтролерною системою є те, що обмін може будуватися у двох режимах – обміну даними та командами.

При обміні даними, використано спеціальний режим кодування з префіксом та суфіксом. Для переводу монітору у режим передачі даних, при виникненні подій, потрібно встановити змінну bkcmd у відповідне значення.

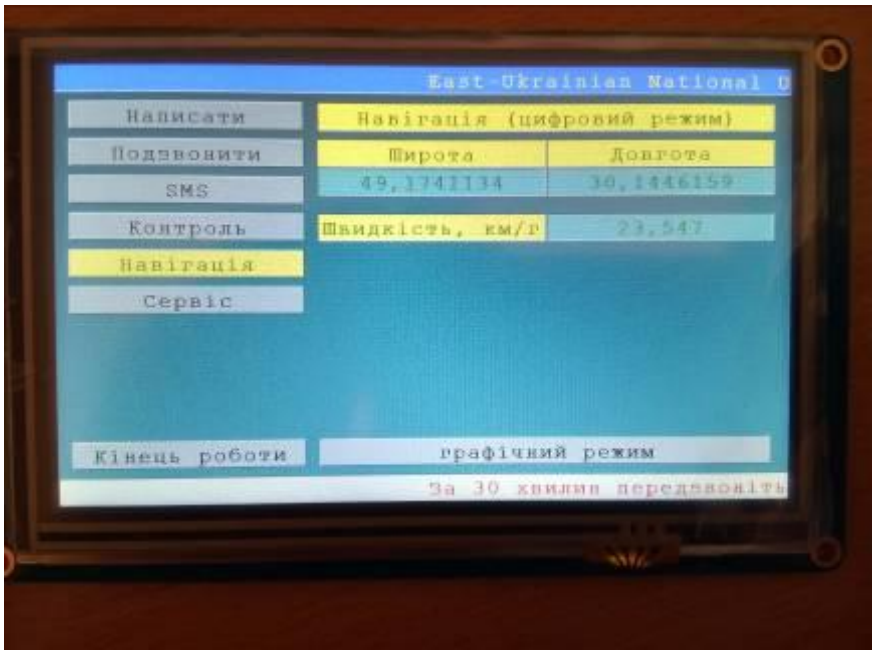


Рис. 4.7. Цифровий режим навігаційного інтерфейсу

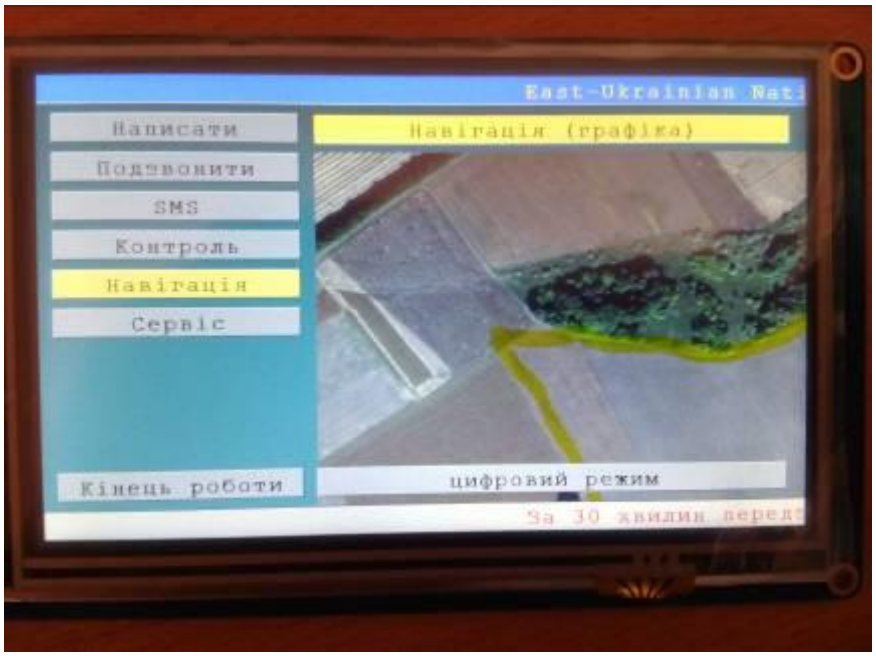


Рис. 4.8. Графічний режим навігаційного інтерфейсу

При цьому, монітор буде видавати данні у послідовний порт, які залежатимуть від режиму роботи – виконання команд, які отримано з послідовного порту, чи видача даних, які відповідають діям з інтерфейсом.

Перехід у режим виконання команд із послідовного порту здійснюється по виконанні команди com_start, а вихід з цього режиму – по виконанні команди com_stop.

Команди повинні передаватися у кодировці ASCII. Кожна команда закінчується символом «END», який кодується послідовністю 3-х байт «0XFF 0XFF0XFF».

Зворотні дані визначаються конструкцією виду – «код даних»+[«атрибут(и)»]+«END». Де квадратні лапки відповідають опційному використанню елементу, тобто він не обов'язково може бути. При цьому, виділяють 2 групи даних – данні про результат виконання команди, яка прийшла з послідовного порту, та «інші» дані.

Для першої групи, значення елементу «код даних» лежить у діапазоні від 0X00 до 0X23 і конструкція даних формується як «код даних»+«END».

Для другої групи поле «код даних» приймає значення від 0X65, а конструкція будується з трьох елементів - «код даних»+[«атрибут(и)»]+«END». Так, наприклад, послідовність «0X65 0X00 0X02 0X01 0XFF 0XFF0XFF» відповідає тому, що у вікні з індексом 0 (перший атрибут) було натиснуто (третій атрибут) на елемент з індексом 2 (другий атрибут).

Такі використовувані формати даних указують на необхідність наявності мікроконтролера, який взаємодіє з монітором селектора, який би аналізував отримані дані та передавав їх відповідним процедурам. Таким чином, мікроконтролер повинен мати програму, яка на основі отриманих даних буде ідентифікувати поточний режим роботи монітору (чи встановлювати його за необхідністю) та визначати які дії виконав оператор на основі даних про індекси елементів інтерфейсу у екранній формі.

При цьому мікроконтролер може виконувати ці дії за перериванням від вбудованого контролеру послідовного порту.

Схема організації взаємодії може бути представлена у виді, наведеному на рис. 4.9.

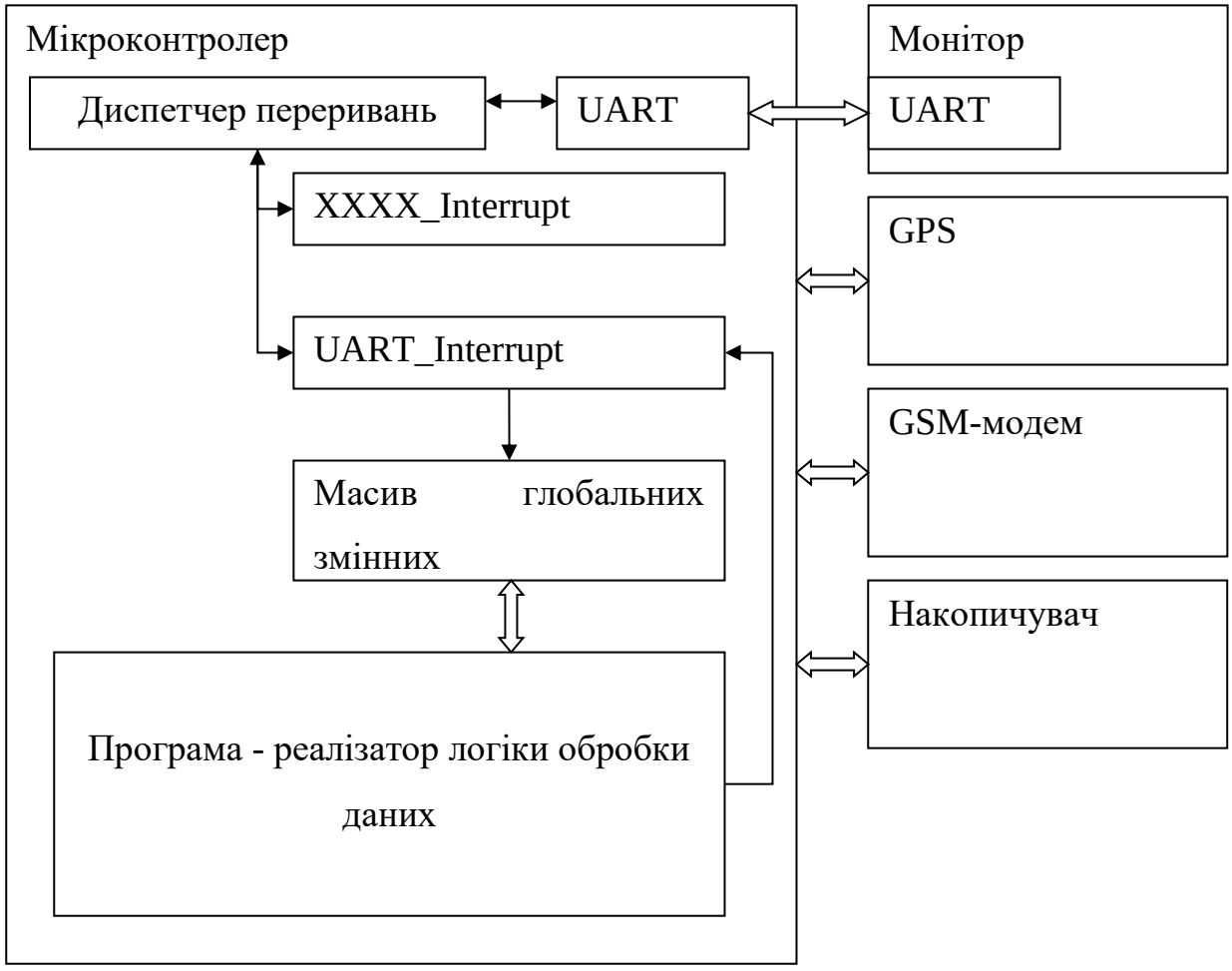


Рис. 4.9. Схема взаємодії засобів у системі.

4.2. Система моделювання та створення системи управління міжнародної відкритої логістичної системи

Розглянемо принцип побудови та моделювання міжнародної відкритої логістичної системи на прикладі реального ланцюжку постачання продуктів харчування та місце закритої логістичної системи обслуговуючого типу на прикладі ФГ «Надія», розглянутого раніше, в цій міжнародній системі [2], [3].

ФГ «Надія» займається збором урожаю та перетворення сировини (сої) на кінцевий продукт (соєва олія), готовий для експортних операцій. Для подальших операцій ФГ «Надія» передає в розпорядження продукцію своєму відокремленому юридичному підрозділу – ПрАТ «Ксібекс», який вже виступає повноцінним

агентом в ланцюжку постачання. По суті ці два агенти діють як одне ціле, однак даний розподіл доцільний з метою оптимізації у їх визначених сферах діяльності.

Розглянемо дві схеми постачання: прямий продаж, експорт та поставка продукції від виробника(експортера) до кінцевого споживача (імпортера) при застосування мономодального міжнародного перевезення на прикладі постачання автотранспортом; та складний ланцюжок постачання з використанням мультимодальних міжнародних перевезень з використанням багатьох видів транспорту та пунктів перевантажень та передачі контролем над вантажем від одного оператора до іншого. Перша схема є доволі простою та відрізняється від автоперевезень всередині країни лише більшої кількістю агентів, що беруть участь, а також посиленого документального контролю та контролю зі сторони державних органів та митниці. За другої схеми в ланцюжку також присутні агенти не тільки державного, а й міждержавного контролю, так як перевантаження відбуваються на території третіх країн та на території поза юрисдикцією будь-якої країни. Документальний та міжурядовий контроль також значно ускладнений значним документопотоком, при цьому документи вимагають гармонізації згідно міжнародним законодавством.

На рис. 4.10 показано міжнародний ланцюжок постачання при автоперевезеннях при прямій поставці.

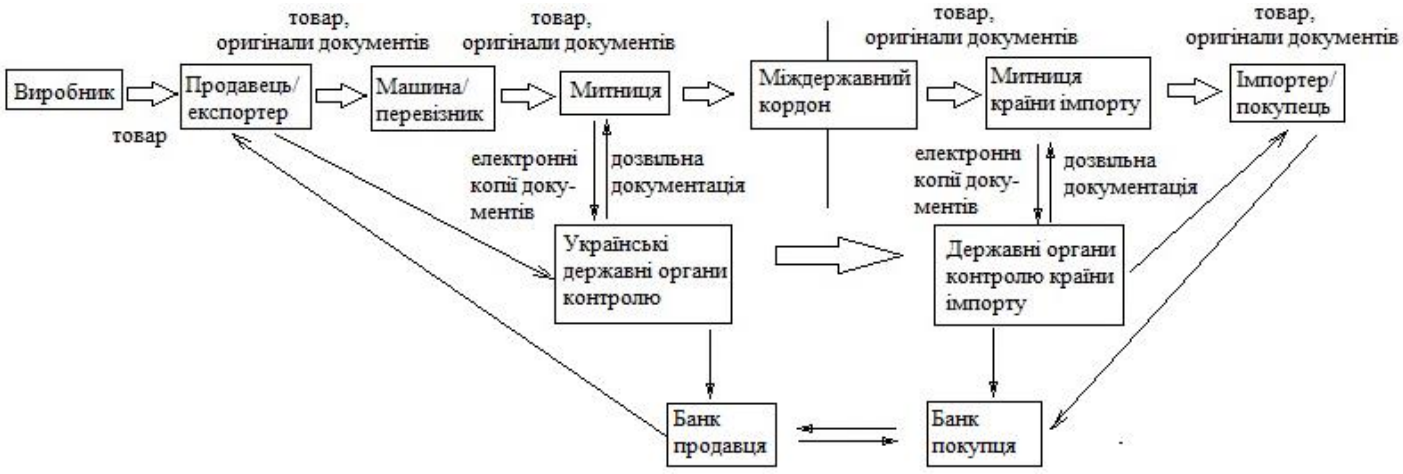


Рис. 4.10. Ланцюжок постачання при мономодальному перевезенні

На практиці документообіг не співпадає з фізичним переміщенням товару в ланцюжку постачання. В даному випадку схема наступна: виробник передає

готовий товар в розпорядження продавця (експортера), цей експортер вже створює всі документи, необхідні для продажу, експортного та митного оформлення, державного контролю, проведення процедур сертифікації та контролю якості, документи для вантажного перевезення товару автотранспортом перевізника та надає копії цих документів для банку, який на їх основі проводить контроль та взаємодіє з банком покупця (імпортера) для повної та своєчасної оплати за відвантажений товар згідно умовам продажів, передбачену договором. Тобто, по суті, продавець (експортер) є першою стартовою ланкою в документообороті в міжнародному ланцюжку постачання. Всі інші агенти цього ланцюжку зацікавлені в правдивості початкової інформації від експортера та контролю за її цілісністю, незмінністю та достовірністю при подальшому її поширенні та передачі від агенту до агенту. Смарт-контракти та захищений документообмін, що базується на технології блокчейн та розподілених системах, між усіма зацікавленими агентами в значній мірі захищають цілісність та достовірність наданої інформації та дозволяє проводити оперативний контроль зі сторони відповідних органів на кожному етапі передачі такої інформації. Злочинні дії зловмисників можуть бути спрямовані на спотворення або часткову зміну чи приховування даних на певних етапах обміну. Найпоширеніша злочинна практика – при володінні всією цілісною картиною та загальною та всеохоплюючою інформацією до розгляду відповідному агенту надається неповна інформація, приховуються деякі її аспекти, таким чином спотворюючи інформацію та отримуються якісь блага від такого контролюючого агенту, наприклад отримання швидкого позитивного висновку від державного органу сертифікації та безпеки, часто це загрожує підтриманню ключових та важливих властивостей та характеристик готової продукції.

На даній схемі при мономодальному перевезенні експортер створює оригінальні документи, передає їх паперові версії водієві транспортного засобу, з якими той проходить процедуру митного оформлення при імпорті та експорті після прибуття в країну призначення та пред’являє прикордонній службі під час фізичного перетину міждержавного кордону транспортним засобом та товаром. Однак якщо країна імпорту не межує територіально з країною експорту, тобто існує

ще країна транзиту, то таких перетинів державного кордону буде декілька. При цьому зростає вірогідність підміни та фальсифікації оригінальних паперових документів зловмисниками в країнах транзиту. Одночасно з передачею паперових документів водієві експортер (продавець) розсилає їх електронні копії усім зацікавленим агентам цього ланцюжку – митниці для попереднього документарного контролю, банку та державним контролюючим органам. При цьому з точки зору властивостей безпеки перевезень доцільно повністю бути впевненим в тому, що початкова інформація від експортера є повною, достовірною та незмінною, а також вона повинна бути ідентична в паперових документах та їх електронних копіях, що надсилаються усім агентам.

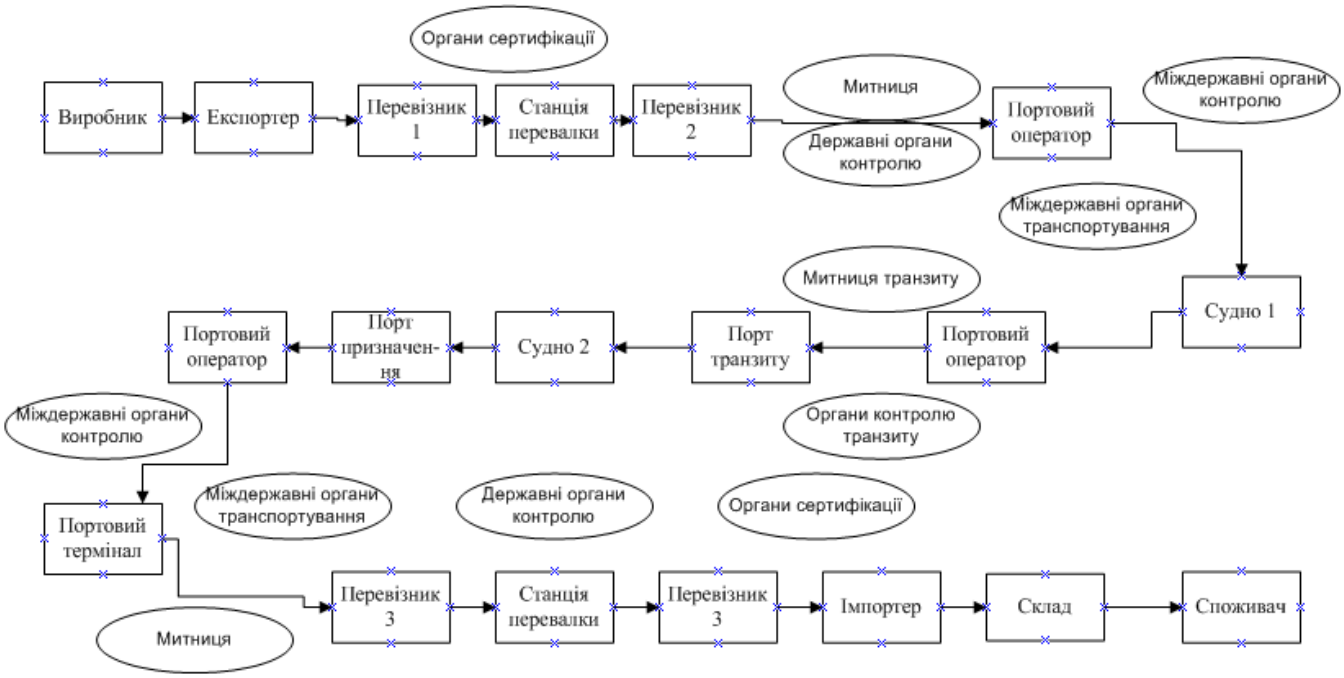


Рис 4.11. Схема фізичного руху товару при мультимодальному міжнародному перевезенні

Видно, що навіть при такому простому ланцюжку постачання можливі значні втрати часу внаслідок обміну документацією та інформацією між агентами. Значний час займає верифікація оригінальності та достовірності даних на кожній інстанції державних контролюючих органів. Також присутня ймовірність шахрайських операцій з ціллю надання недостовірної інформації для отримання комерційної вигоди. Однак за даної схеми інші ризики, зокрема стабільності та дотримання заданих параметрів транспортування продукції порівняно низькі через короткий час доставки автотранспортом та малої кількості транспортних агентів-посередників при перевантаженнях та відсутності періоду зберігання на пунктах перевалки вантажів в очікуванні прибуття наступного перевізника. Розглянемо складнішу схемуланцюжка постачання за умови мультимодального перевезення та більшої кількості агентів.

Складне мультимодальне перевезення для кращого розуміння доцільно розглянути в розрізі фізичного руху товару та в розрізі документообігу, пов’язаним з цим фізичним рухом.

Як бачимо, навіть при спрощеній подачі видно наскільки ланцюжок документообігу складніший та відрізняється від реального руху товарів між агентами системи. Фізичний товар послідовно передається від одного агенту до іншого вздовж усього ланцюжку постачання, який би складний та довгий він не був, в той час як документообіг ніяк не прив’язаний до реального місцезнаходження товару. Авторами одного документу може бути більше, ніж один агент одночасно, найбільш поширений варіант – автором документу є агент, який надає певну початкову вихідну інформацію, а потім дана інформація перевіряється, підтверджується та доповнюється наступними агентами, тобто цей документ проходить низку верифікації органами контролю, банками та іншими учасниками ланцюжку постачання. При цьому на певній початковій ланці певний агент створює оригінальний паперовий документ, який надається початковому перевізнику, та який в подальшому супроводжує товар впродовж усього шляху фізичного переміщення, та на якому ставляться фізичні відмітки усіх контролюючих органів, митниці та агентів-посередників. Після створення цей

паперовий оригінальний документ в його електронній копії надається усіх пов’язаним та зацікавленим агентам як джерело початкової інформації. При цьому очевидно, що інформація змінюється в процесі верифікації (доповнюється та розширяється або видаляється зайве) при обробці наступним агентом.

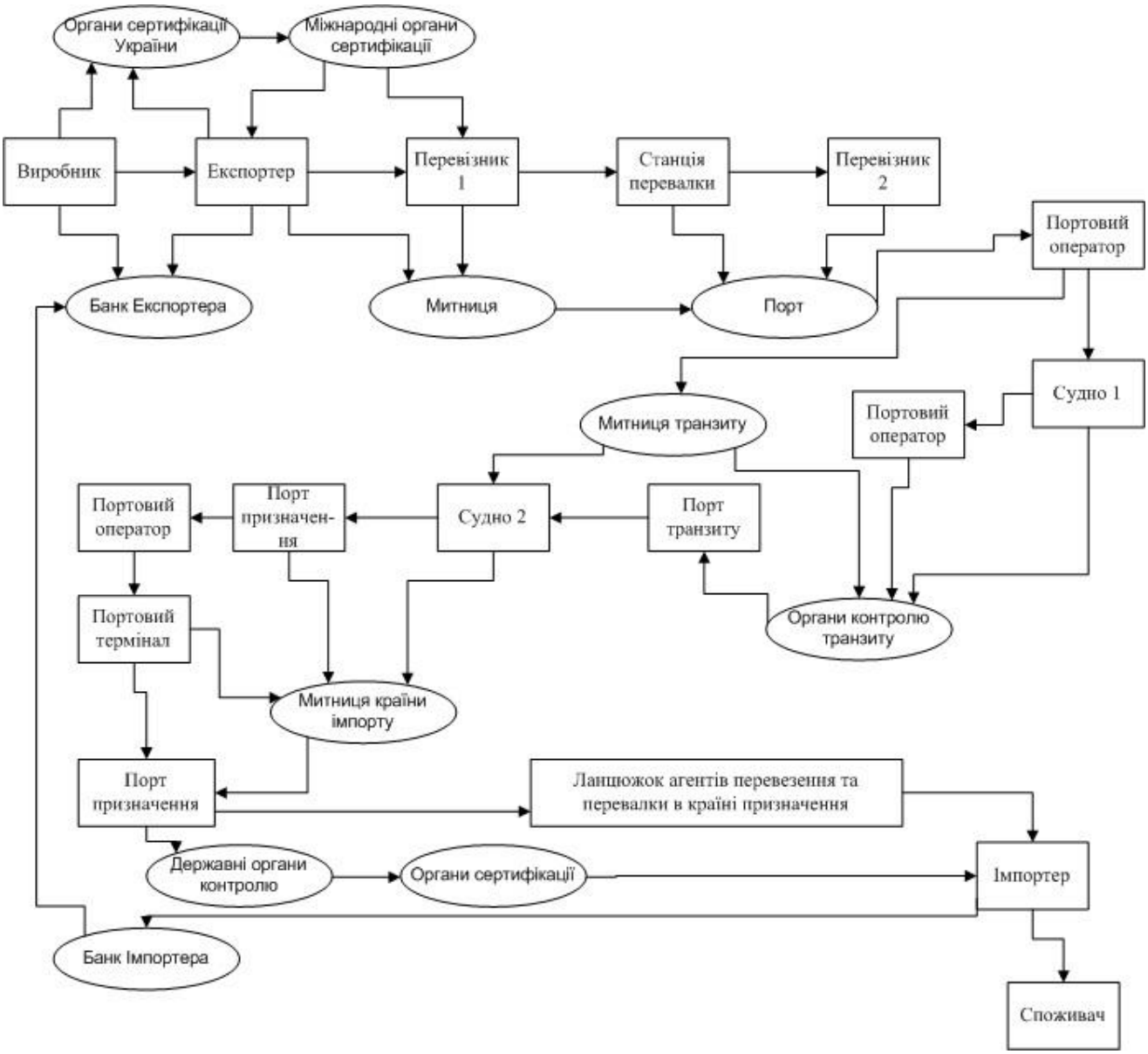


Рис 4.12. Схема документообігу при мультимодальному міжнародному перевезенні.

З однієї сторони обмін електронними копіями та інформацією в електронному виді значно прискорює обмін, обробку та перевірку інформації в порівнянні з паперовою документацією та значно економить час, але, з іншої сторони, часто буває, що початкова інформація на паперовому носії від

початкового агенту дуже сильно відрізняється від кінцевого варіанта в верифікованому електронному виді, коли вона в обох варіантах надходить до кінцевого споживача цієї інформації. Через значну розгалуженість мережі обміну документацією та велику кількість агентів і виникає ризик зловживання через спотворення інформації. Наприклад, по протоколу банк імпортера проводить платіж банку виробника-експортера за відвантажену продукцію за умови виконання ряду передумов, тобто отримання певної підтвердженої інформації від певних агентів: митниці країни імпорту та експорту, фізичного перевізника, органів сертифікації та охорони здоров'я, портів призначення та транзиту та багатьох інших. Отримана з цих багатьох джерел інформація співставляється та перевіряється на предмет взаємного підтвердження і несуперечності.

Тому і постає основне завдання оперативного контролю достовірності, цілісності і незмінності інформації, яка поширюється. Найповнішою мірою це вдається досягнути за допомогою використання смарт-контрактів та максимального залучення розподілених систем, зокрема блокчейн, для створення електронних документів та обміну інформацією між агентами.

4.3 Модельний спосіб розробки контракту для мономодального перевезення на блокчейн платформі.

Розглянемо всю модель ланцюжку постачання для мономодального перевезення, що створена на основі блокчейн платформи. Як вже зазначалося, блокчейн дає захист від втручання сторонніх осіб в документообіг та захист від шахрайства в транзакціях.

Ми будуємо модель всього ланцюжка у вигляді інсерційної моделі. Далі інсерційна модель має бути переведена в мову Solidity, що забезпечує надійне та безпечне функціонування всього ланцюжка.

Спершу розглянемо агенти та їх дії.

Агент Виробник постачає продукцію для експортера та отримує документи за якими після закінчення всього ланцюжка поставки та проходження всіх платежів отримує певну суму за поставлену продукцію.

Агент Експортер отримує продукцію від Виробника та оформлює документи, обирає та винаймає міжнародного перевізника та проводить необхідне митне оформлення.

Ми абстрагуємось від українських державних органів контролю, митниці та банків, як тими, що не є елементами блокчейну.

Агент Перевізник отримує всі супровідні документи, завантажує продукцію та поставляє її на митницю країни, в якій знаходиться імпортер.

Агент Імпортер сплачує повну вартість за продукцію експортеру. Можливо складати контракт таким чином, щоб імпортер робив передплату для попередніх затрат (перевозки, митниця).

Таким чином агенти обмінюються документами та проводять транзакції по оплаті частини ланцюжку постачання. Для захисту документообігу в блокчейні доречно зберігати цифровий образ документу (хеш функцію), а самі документи зберігати в традиційних базах даних. При підміні документів, цифрових образ суттєво змінюється і тому в розумний контракт відповідно виявить підміну в базі, якщо така буде можлива та відповідно заблокує або здійснить відповідні захисні дії.

Сам блокчейн не дуже пристосований зберігати документи, тому гарантами документообігу є цифровий образ документу та цифровий підпис відповідних гарантів та підписантів документі.

Таким чином кожен блок мережі містить наступну інформацію:

- цифровий образ документу;
- цифровий підпис та дії сторін;
- фінансові транзакції.

З точки зору інсерційних моделей ми розглянемо середовище, що включає атрибути, які визначають часові та фінансові властивості та величини. В даному середовищі взаємодіють 4 агенти, а середовище визначає дії зовнішніх агентів, такі як державні органи, банки та митниця.

Нехай T0 – початковий час дії розумного контракту, що визначає весь ланцюжок поставки, що визначається передплатою імпортера на рахунок експортера Tad

Tk- кінцевий термін дії ланцюжка поставки. Нехай він визначається кінцевою оплатою за продукцію Виробнику. Тоді запишемо умови контракту із додаванням нових часових атрибутів.

`Tst<= Tad + d1`, де Tst– час оформлення митних документів, d1 –вибраний часовий проміжок;

`Tship<= Tst + d2`, Tship – час поставки продукції на митний склад імпортера,

`Timp<= Tship + d3`, Timp – час розмитнення продукції на складі імпортера,

`Tpay_imp<= T0 + d4`, Tpay_imp – час оплати за продукцію, який може бути вибраний, як відповідна кількість днів від початку дії контракту.

`T_pay_man<= Tpay_imp + d5`, T_pay_man – відповідний час оплати виробнику, що завершує ланцюг поставки.

Дані умови контракту є властивостями безпеки контракту. Якщо вони порушуються, то відповідні дії можуть бути виконані, що визначають штрафні санкції, або неможливість поставки, або страховий випадок.

Визначимо дії агентів. При чому маємо на увазі, що кожен агент має свій рахунок, визначений атрибутом Money.

Агент Виробник – ім’я агенту A1, тип Manufacturer.

`DefinePrice: 1 -> “send(invoice,A2);” Price = Calc(X)` – дія виробника, де визначається ціна Price та надсилається рахунок до експортера;

`ShipProduct: 1-> “receive (certificate1,A2);” 1` – постачання продукції на склад експортера. В даному випадку ми абстрагуємось від кількості та типу продукції для аналізу властивостей документообігу та фінансових аспектів.

Агент Експортер – ім’я агенту A2, тип Exporter.

`ReceiveProduct : 1->“send(certificate,A1)” 1` – отримання продукції, що визначається відповідним документом про отримання.

`PrepareDocStart: 1-> “sendDoc(Env)” 1` – початок оформлення документів.

`ReceivePayment: 1 -> “receive (dP*Price, A4)” Tad = CurrentTime(); A2.Money = dP*Price` – отримання передплати на рахунок експортера.Функція CurrentTime() визначається середовищем та буде мати значення для імплементації в розумному контракті.dP–частка передплати.

`PrepareDocEnd: 1-> “sendDoc(Env)” Tst = CurrentTime()` – кінець оформлення документів.

`HireShip: 1->”sendDoc(A3),Payment(A3)” A2.Money = A2.Money - PriceShipment` – відправка передплати для перевізниката документів.

Агент перевізник A3.

`ReceiveMoneyShip: 1->”receive Doc(A2),Payment(A2)” A3.Money = A3.Money + PriceShipment` – отримання документів та передплати.

`DoShipment: 1->”send Doc(A4)” Tship = CurrentTime()` – поставка продукції імпортеру.

Агент імпортер A4.

`GetImport: 1-> “receive Doc(A3)” Timp = CurrentTime()` – отримання продукції від перевезника

`PayImport: 1->”send Payment(A2)” A4.Money = A4.Money - Price*(1 - dP)` – відправка решти оплати до експортера.

Відповідні формалізовані дії мають бути для оплати виробнику.

Поведінка даної системи визначається чіткою послідовністю дій агентів. Таким чином, було формалізовано весь ланцюжок поставки, як перший етап розробки системи, що реалізує його. Система є параметризована, зокрема атрибутами d1,d2,…, що визначаються при конкретній реалізації ланцюжка. Отже можемо перевірити за допомогою символьного моделювання досяжність порушення властивостей безпеки. При цьому отримаємо формулу для параметрівd1,d2,… та часових атрибутів, при яких ці порушення можливі. Таким чином в подальшому реалізація має перевіряти подібні порушення вже в специфікаціях іншого рівня, а саме рівня розумного контракту на мові Solidity.

Таким чином, етап вимог до ланцюжка постачання визначає властивості безпеки, що мають бути взяті до уваги при написанні коду розумного контракту.

Ми розглянули лише часові властивості безпеки, але при складанні всіх факторів і всіх вимог в ланцюжку постачання може бути сформульовано інші умови, наприклад в зв'язку із властивостями продукції, термін поставки разом із часом перебування на митних складах, має бути не більше допустимої межі.

При реалізації штрафних санкцій, додаються дії, що зв'язані із витратами на штрафи на митниці або при вичерпанні терміну поставки. В зв'язку з цим треба розглядати взагалі можливість скасування поставки.

Після аналізу та отримання розширених властивостей безпеки, код Solidity, що включає дії всіх агентів може бути згенеровано автоматично і далі деталізовано з метою мінімізації невідповідностей вимогам. На рівні мови Solidity проводиться верифікація стандартних властивостей неповноти та властивостей безпеки із пошуком відповідних вразливостей та стійкістю до атак.

Висновки до Розділу 4.

Таким чином в четвертому розділі ми розглянули систему внутрішньої обслуговуючої логістики закритого типу на рівні фермерського господарства, що створена з використанням модельного методу розробки. Дана закрита логістична система є постачальником готової продукції та стартовою точкою для міжнародного ланцюжка постачання товарів та продукції. Тобто весь ланцюг постачання та логістичні процеси, які його супроводжують та забезпечують, описано від виробника сировини (фермерського господарства) до імпортера в іншій країні. Модельний метод розробки логістичної системи для фермерського господарства базується на системному підході, тобто система розглядається, як сукупність певного непостійного числа агентів, які діють в закритому середовищі, та їх діяльність концентрована навколо незалежного захищеного ядра, яке збирає, опрацьовує та розподіляє інформацію. В цьому розділі розроблено та описано методологію створення інструментів для системи управління логістичною

системою в розрізі фізичної системи компонентів для збору та передачі навігаційної інформації, інформаційно-комунікаційної мережі та компонентів ядра системи з врахуванням відповідної специфіки використання в галузі сільського господарства.

Надалі розглянуто рух товарів на виході цієї закритої логістичної систем уздовж усього ланцюжка постачання в міжнародній логістиці. Аналіз системи фізичного переміщення товару через усі державні, контролюючі, митні та інші інстанції, а також через мережу логістичних агентів, та його порівняння з національним та міжнародним документообігом, пов'язаним з рухом цього товару, показує можливий простір дій потенційних зловмисників та вразливостей міжнародної логістичної системи відкритого типу. Це пов'язано з надзвичайно розгалуженою мережею обміну електронною документацією між великою кількістю незалежних агентів різного рівня та повною відірваністю цього електронного документообігу від фізичного транспортування товару, а також з необхідністю постійно доповнювати, перевіряти та верифікувати початкову вхідну інформацію кожним агентом на своєму рівні доступу, повноважень та компетенції. Ланцюжок постачання заснований на технології блокчейн з метою невтручання в цей документообіг та мінімізації випадків шахрайства та злочинних дій з фінансами, привласнення товару або отриманням інших неправомірних пільг чи конкурентних переваг.

Також наведено приклад модельного способу розробки смарт-контракту для міжнародного ланцюжку постачання та безпечної оплати за продукцію на блокчейн платформі та за використання мови Solidity. Для цього спочатку ланцюжок постачання описується у вигляді інсерційної моделі, а далі трансформується у мову Solidity, що забезпечує надійне та безпечне функціонування всього ланцюжку. Поведінка даної системи визначається чіткою послідовністю дій усіх її агентів при правильному виконанні певних виконаних передумов. Смарт-контракт же аналізує цифровий образ документів та їх цифровий підпис та пов'язані з ними транзакції. Інсерційне середовище агентів містить атрибути, які визначають часові та фінансові властивості та величини. За допомогою символьного моделювання

перевіряється досяжність порушення властивостей безпеки, в результаті отримаємо формулу для параметрів та часових атрибутів, при яких ці порушення можливі. В подальшому необхідно перевіряти подібні порушення вже в специфікаціях іншого рівня, а саме рівня розумного контракту на мові Solidity. Етап вимог до ланцюжку поставок визначає властивості безпеки, що мають бути взяті до уваги при написанні коду розумного контракту. На рівні мови Solidity проводиться верифікація стандартних властивостей неповноти та властивостей безпеки із пошуком відповідних вразливостей та стійкістю до атак.

Список використаних джерел до Розділу 4.

1. Husev B.S., Horbatiuk S.O., Savytska I.A., Smolii V.V., Shelestovskyi V.H. Information technology of farm management system. Monograph. – NUBaP of Ukraine, 2018. – 220 pages.

2. O.O. Letychevskyi, S.O. Gorbatuk, V.A. Gorbatuk. Algebraic modelling in international and local service logistical systems // Problems of Programming – 2020, №4. – pages 88-97.

3. Oleksandr Letychevskyi, Serhii Horbatiuk, Viktor Horbatiuk. Algebraic modelling of logistical systems equipped by wireless monitoring devices // The 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (17-18 September 2020, Dortmund, Germany)

4. Oleksandr Letychevskyi, Volodymyr Peschanenko, Sergiy Horbatiuk. Consensus Protocol Security Analysis Using an Algebraic Virtual Machine // IntelITSIS’2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security (23–25 March, 2022) – Khmelnytskyi, Ukraine - pages 484-493.

ВИСНОВКИ

У дисертаційній роботі розв’язана актуальна науково-прикладна задача створення технології побудови безпечних та надійних логістичних систем відкритого та закритого типу різного масштабу, таких як глобальних міжнародних, так і обслуговуючих закритих логістичних систем, необхідних для функціонування міжнародних ланцюжків постачання товарів та продукції. Використання технології розподілених систем, зокрема блокчейн, дає змогу виконати вимоги стандартів для систем, що критичні до безпеки, що важливо в галузі міжнародної логістики.

Використання алгебраїчного підходу та системи інсерційного моделювання в дослідженні надійності програмних систем, зокрма логістичних, показало ефективність в задачах кібербезпеки. Підхід працює із множинами сценаріїв поведінки системи на відміну від попередніх методів аналізу надійності – імітаційного моделювання та ймовірнісного підходу, та дає **доказове підтвердження** здійсненності властивостей безпеки та стійкості до атак.

При вирішенні цієї задачі отримано такі основні наукові та практичні результати:

1. Проведено аналіз передових рішень щодо використання технології розподілених систем в побудові складних логістичних систем. Виявлено та проаналізовано глобальні проблеми, які накопичились в сучасній системі міжнародної логістики, та які можуть бути вирішені за допомогою застосування технології блокчейн та методів модельної розробки систем. Розглянуто основні проблеми в ланцюжку постачання та контролю якості товарів у міжнародній логістиці, виявлено потенціал розвитку та застосування технології блокчейн та смарт-контрактів.

2. Проаналізовано випадки відомих схем кібератак, проведено аналіз властивостей безпеки та можливих атак зловмисників в логістичних системах, що побудовані на платформах блокчейн. Розглянуто методи підвищення безпеки в логістичних системах, захисту від атак зловмисника та шахрайства та методи аналізу вразливостей, зокрема в системах на основі блокчейн платформ на основі

інсерційного підходу. Розглянуто інсерційну семантику блокчейн системи та методи виявлення вразливостей.

3. Запропоновано рішення, методи кібербезпеки та алгоритми верифікації для перевірки властивостей цілісних логістичних систем, таких як безпека та життєдіяльність; зокрема, як найбільш вдалий та ефективний, запропоновано використання комбінації модельного способу розробки та тестування управляючих розподілених систем для логістики разом з алгебраїчним та інсерційним підходом на стадіях модельного способу розробки для процедур верифікації, валідації та тестування.

4. Вперше запропоновано життєвий цикл модельної розробки високонадійних програмних систем логістики. Життєвий цикл включає залучення до кожного етапу розробки процедур верифікації та тестування на основі створення формальних моделей. Визначено та налаштовано формальні методи для верифікації та модельного тестування логістичної системи, а саме – методи символьного моделювання та статичного доведення властивостей.

5. Проведено аналіз шляхів побудови логістичних систем контролю реального часу в рамках побудови системи контролю для логістичного обслуговуючого середовища закритого типу для фермерського господарства. В результаті аналізу знайдено аналоги та обрано прототипи таких систем, виявлено їх суттєві недоліки. Розроблено формальні моделі логістичної обслуговуючої системи фермерського господарства в мові алгебри поведінок, проведена верифікація та генерація тестового набору.

6. Створено інструментарій та методологію для використання розподілених систем, зокрема технології блокчейн в логістичних системах. Запропоновано архітектурно-структурні рішення, алгоритми та програмно-апаратні засоби формування і відображення взаємодії логістичних агентів та безперервного контролю за їх розміщенням та поведінками з метою дотримання безпеки функціонування.

7. Приведено повну технологію створення логістичної системи побудованої на блокчейн платформі на прикладі системи логістичних перевезень.

8. Показано приклад проектування реальної моделі діючої логістичної системи обслуговуючого типу з широким застосуванням методу модельного тестування (МВТ), як оптимального інструменту для інтерактивних інформаційних систем, та з інтеграцією процедури верифікації властивостей безпеки у стандартний процес модельної розробки програмного забезпечення. Згенеровано повний тестовий набір та сценаріїв поведінки за допомогою символьного моделювання та модельного тестування для вирішення задачі досяжності властивостей або їх порушення. Для системи логістики фермерського господарство було згенеровано набір сценаріїв із покриттям всіх ребер графу поведінки.

9. Змодельовано дії зловмисника при атаці на блокчейн систему. Проаналізовано інсерційну семантику розумних контрактів (Smart Contracts), які використовуються в розподілених системах, за допомогою алгебри поведінок. Запропоновано шляхи запобігання кібер-атакам при проектуванні блокчейн системи шляхом аналізу шаблонів кібер-атак на можливість досяжності успішної атаки на стадії модельної розробки системи.

Результати досліджень, отримані і впроваджені при проведенні науково-дослідних фундаментальних та прикладних НДР Інституту кібернетики імені В.М. Глушкова НАН України, пошукових НДР та у навчальному процесі на факультеті інформаційних технологій НУБіП України, у виробничу діяльність фермерського господарства «Надія», ТОВ «Смарт Трейдинг», ПрАТ «Ксібекс», Херсонського Державного Університету, приватного підприємства ЛітСофт.

Результати впровадження підтверджено відповідними актами. Копії актів впровадження результатів дисертації у виробничому процесі подані в Додатку Д.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Васюхін М., Касім А., Долинний В., Касім М., Шелестовський В., Горбатюк С. Метод створення класифікатора картографічної інформації для агрономічних автоматизованих систем // Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні ‘2017: збірник матеріалів V Міжнародної науково-практичної конференції (22 – 23 червня 2017 року, Київ). – К.: Компринт, 2017. – С. 47–49.

2. Васюхін М.І., Касім А.М., Долинний В.В., Касім М.М., Шелестовський В.Г., Горбатюк С.О. Геоінформаційна система для малих і середніх фермерських господарств // Безпека життєдіяльності на транспорті і виробництві – освіта, наука, практика: тези доповідей IV Міжнародної науково-практичної конференції (14-16 вересня 2017 року, Херсон). – Херсон: ХДМА, 2017. – С. 324–330.

3. Васюхін М.І., Горбатюк С.О., Касім М.М., Шелестовський В.Г. комп’ютерні системи. Навчальний посібник. – К.: Компринт, 2017. – 270 сторінок. ISBN 978-966-929-552-1

4. Гусєв Б.С., Горбатюк С.О., Савицька Я.А., Смолій В.В., Шелестовський В.Г. Інформаційна технологія системи управління фермерським господарством. Монографія. – НУБіП України, 2018.- 220 сторінок.

5. С. Горбатюк. Блокчейн в логістиці та формальна верифікація властивостей безпеки // Ідеї академіка В.М. Глушкова і сучасні проблеми штучного інтелекту. 8-ма Всеукраїнська науково-практична конференція: “Глушковські читання” (29 жовтня 2019, Київ). – Київ, Видавничий дім Ліра-К, 2019. – С. 57–60.

6. О.О. Летичевський, С.О. Горбатюк. Децентралізовані системи в логістиці: огляд використання та проблеми безпеки // Проблеми програмування – 2020, №1. – С. 55-73. DOI: 10.15407/pp2020.01.055

7. О.О. Летичевський, С.О. Горбатюк, В.О. Горбатюк. Алгебраїчне моделювання в системах міжнародної та місцевої обслуговуючої логістики // Проблеми програмування – 2020, №4. – С. 88-97. DOI: 10.15407/pp2020.04.088

8. Oleksandr Letychevskyi, Serhii Horbatiuk, Viktor Horbatiuk. Algebraic modelling of logistical systems equipped by wireless monitoring devices // The 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (17-18 September 2020, Dortmund, Germany)

9. Oleksandr Letychevskyi, Volodymyr Peschanenko, Maksym Poltoratskyi, Serhii Horbatiuk, Horbatiuk Viktor and Yuliia Tarasich. One Approach to Formal Verification of Distributed Ledger Technologies on the Blockchain Technologies Example // Conference on Mathematical Foundations of Informatics: Proceedings MFOI-2020 (12–16 January 2021) – Taras Shevchenko National University of Kyiv - pages 227-242.

10. Oleksandr Letychevskyi, Volodymyr Peschanenko, Sergiy Horbatiuk. Consensus Protocol Security Analysis Using an Algebraic Virtual Machine // IntelITSIS'2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security (23–25 March, 2022) – Khmelnytskyi, Ukraine - pages 484-493.

11. Економічна енциклопедія: У трьох томах. Т. 2. / Редкол.: …С. В. Мочерний (відп. ред.) та ін. – К.: Видавничий центр "Академія", 2000. – 864с.

12. Крикавський Є.В., Чорнописька Н.В. Логістичні системи: Навч. посібник. – Львів: Видавництво Національного університету "Львівська політехніка", 2009. – 264 с.

13. Крикавський Є.В. Логістичне управління: Підручник. – Львів: Видавництво Національного університету "Львівська політехніка",2005. – 684 с.

14. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)

15. Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V.: Blockchain technology: beyond bitcoin. Appl. Innov. 2(6), 6–10 (2016)

16. McPhee, C., Ljutic, A.: Blockchain. Technol. Innov. Manag. Rev. 7(10), 3–5 (2017)

17. Electricity journal homepage: blockchain technology: will it make a difference? Electr. J. 30 (3), 86–87 (2017)

18. Finextra: Marine Transport International Applies Blockchain to Shipping Supply Chain (2016). URL: https://www.finextra.com/pressarticle/66223/marine-transportinternational-applies-blockchain-to-shipping-supply-chain

19. Solesvik, M.Z.: Interfirm collaboration in the shipbuilding industry: the shipbuilding cycle perspective. Int. J. Bus. Syst. Res. 5(4), 388–405 (2011)

20. Official website of International Maritime Organization –IMO. URL: http://www.imo.org

21. Den Norske Veritas—DNV GL: Making your Asset Smarter with the Digital Twin. URL: https://www.dnvgl.com/article/making-your-asset-smarter-with-the-digital-twin-63328

22. The Economist. The data deluge, 25 Feb 2010. URL: http://www.economist.com/node/15579717

23. Tien, J.M.: Internet of connected servgoods: considerations, consequences and concerns. J. Syst. Sci. Syst. Eng. 24(2), 130–167 (2015)

24. Lacey, M., Lisachuk, H., Giannopoulos, A., Ogura, A.: Shipping smarter: IoT opportunities in transport and logistics. The Internet of Things in Shipping. Dupress-Deloitte (2015)

25. Offshore Energy Today: Meet Onboard, the Maritime Internet of Things. URL: http://www.offshoreenergytoday.com/oeec-meet-onboard-the-maritime-internet-of-things/?utm_source=emark&utm_medium=email&utm_campaign=daily-update-offshore-energy-today-2017-10-02&uid=207087

26. Maritime: Six Maritime Start-Ups That Are Changing the GAME. URL: https://knect365.com/talentandtraining/article/1149354e-68d9-4e74-9f91-a900ac869526/6-maritime-startupsthat-are-changing-the-game

27. Den Norske Veritas—DNV GL: Certificates in the blockchain. URL: https://www.dnvgl.com/assurance/certificates-in-the-blockchain.html

28. Den Norske Veritas—DNV GL: Blockchains in the shipping world. URL: https://www.dnvgl.com/expert-story/maritime-impact/Blockchains-in-the-shipping-world.html

29. Splash 24: Dr. Martin Stopford on the Future of Shipping. URL: http://splash247.com/dr-martinstopford-future-shipping

30. Kondratenko, Y.P., Kozlov, O.V., Korobko, O.V., Topalov, A.M.: Internet of things approach for automation of the complex industrial systems. In: ICTERI-2017, CEUR Workshop Proceedings Open Access, vol. 1844, pp. 3–18 (2017). URL: https://pdfs.semanticscholar.org/3ff6/4e4a07be1e8c2f0b16f4736397be1405218a.pdf

31. Tachizawa, E.M., Alvarez-Gil, M.J., Montes-Sancho, M.J.: How “smart cities” will change supply chain management. Supply Chain Manag. Int. J. 20(3), 237–248 (2015)

32. Hahn, G.J., Packowski, J.: A perspective on applications of in-memory analytics in supply chain management. Decis. Support Syst. 76(1), 45–52 (2015)

33. Maersk and IBM Introduce TradeLens Blockchain Shipping Solution. Aug. 9, 2018. URL: https://newsroom.ibm.com/2018-08-09-Maersk-and-IBM-Introduce-TradeLens-Blockchain-Shipping-Solution

34. TradeLens – official website. URL: https://www.tradelens.com

35. Forbes. IBM-Maersk Blockchain Platform Adds 92 Clients As Part Of Global Launch. URL: https://www.forbes.com/sites/michaeldelcastillo/2018/08/09/ibm-maersk-blockchain-platform-adds-92-clients-as-part-of-global-launch-1/#c469d5068a4a

36. Coindesk. IBM and Maersk Struggle to Sign Partners to Shipping Blockchain. Oct 26, 2018. URL: https://www.coindesk.com/ibm-blockchain-maersk-shipping-struggling

37. Computerworld. Maersk adds two big shipping firms to its blockchain ledger. 29 травня 2019р. URL: https://www.computerworld.com/article/3398923/maersk-adds-two-big-shipping-firms-to-its-blockchain-ledger.html

38. TC. IBM-Maersk blockchain shipping consortium expands to include other major shipping companies. URL: https://techcrunch.com/2019/05/28/ibm-maersk-blockchain-shipping-consortium-expands-to-include-other-major-shipping-companies/

39. TradeLens official brochure. Solution brief. Edition two.

40. SupplyChain Dive. 9 ocean carriers, terminal operators join new blockchain initiative to rival TradeLens. Nov. 7, 2018. URL:

https://www.supplychaindive.com/news/ocean-carriers-new-blockchain-cosco-cma-cgm/541630/

41. CargoSmart. Global Shipping Business Network. URL: https://www.cargosmart.ai/en/

42. Globe news wire. Top Ocean Carriers and Terminal Operators Initiate Blockchain Consortium. November 06, 2018. URL: https://www.globenewswire.com/news-release/2018/11/06/1646014/0/en/Top-Ocean-Carriers-and-Terminal-Operators-Initiate-Blockchain-Consortium.html

43. Supply Chain Dive. Maersk blockchain solution TradeLens adds ZIM. April 22, 2019. URL: https://www.supplychaindive.com/news/maersk-blockchain-solution-tradelens-adds-zim/553146/

44. Ball, B.: Reducing Global Logistics Cost with Benchmarking and Shipping Container Pricing Transparency. Aberdeen Group (2016)

45. Dhanji, T.: Blockchain—Where Oil and Gas Traders Dare to Trade. Ernest Young Publications (2017). URL: https://www.linkedin.com/pulse/blockchain-where-oilgas-traders-dare-tread-talib-dhanji

46. Freight Waves.: Swiss firm brings blockchain to the biopharmaceutical cold chain. 02/23/2018. URL: https://www.freightwaves.com/news/blockchain/skycellblockchaincoldchain

47. John G. Smith.: Block by Block: How blockchain will transform trucking. 18.01.2018. URL: https://www.todaystrucking.com/block-block-blockchain-will-transform-trucking/

48. Ana Alexandre.: New Blockchain-Based Supply Chain System Is Presented by Microsoft and Adents. URL: https://cointelegraph.com/news/new-blockchain-based-supply-chain-system-is-presented-by-microsoft-and-ardents

49. Suku. The future of supply chain is here. URL: https://www.suku.world/

50. DHL Trend Research. URL: https://www.logistics.dhl/content/dam/dhl/global/core/documents/pdf/glo-core-blockchain-trend-report.pdf

51. OriginTrail. URL: https://origintrail.io

52. Shipchain. THE END-TO-END LOGISTICS PLATFORM OF THE FUTURE: TRUSTLESS, TRANSPARENT TRACKING. URL: https://shipchain.io/

53. Nick Szabo - Smart Contracts: Building Blocks for Digital Markets. URL: www.fon.hum.uva.nl

54. Nick Szabo. Formalizing and Securing Relationships on Public Networks.

55. Welcome to ERights.Org. URL: erights.org

56. A Formal Language for Analyzing Contracts. URL: vwh.net

57. Gideon Greenspan.: Smart contracts make slow blockchains. URL: https://www.multichain.com/blog/2015/11/smart-contracts-slow-blockchains

58. StackEdit Viewer / Benoit Schweblin. URL: https://stackedit.io/viewer#!url=https://gist.githubusercontent.com/gavofyork/31b35cd2252a00d0d057/raw/16de06189d2175d2e31b300f1f8531e20c927635/solidity-original

59. Solidity — Solidity 0.2.0 documentation. URL: https://solidity.readthedocs.io/en/latest/

60. Zigurat. Blockchain Success cases: Supply Chain and Logistics. URL: https://www.e-zigurat.com/innovation-school/blog/blockchain-success-cases/

61. Robert Hackett.: Walmart and 9 Food Giants Team Up on IBM Blockchain Plans. URL: https://fortune.com/2017/08/22/walmart-blockchain-ibm-food-nestle-unilever-tyson-dole/

62. Карта випадків африканської чуми свиней в Україні та зони карантину. URL: http://www.asf.vet.ua/index.php/asfinukraine

63. Державна служба України з питань безпечності харчових продуктів та захисту споживачів: Африканська чума свиней? URL: http://www.asf.vet.ua/index.php/purpose-project/about-asf/124-african-swine-fever

64. Укрінформ. Нашестя метеликів – розплата за людську необачність. 03.06.2019 URL: https://www.ukrinform.ua/rubric-economy/2713807-nasesta-metelikiv-rozplata-za-ludsku-nenazerlivist.html

65. Petropavlivka.City. Миколаївку атакує гусінь. 05.06.2019 URL: https://petropavlivka.city/read/experiance/33845/mikolaivku-atakue-gusin

66. Carnet Maritime. Collision CMA CGM Verlaine et Odessa Star. 04.04.2010 URL: http://carnet-maritime.com/accidents-naufrages/collision-cma-cgm-verlaine-et-odessa-star.html

67. Информационный портал «Транспортный бизнес Украины». К ситуации столкновения контейнеровоза Verlaine и теплохода Odessa Star 4 апреля в Мраморном море - Международная юридическая служба. URL: http://tbu.com.ua/news/k_situatsii_stolknoveniia_konteinerovoza_verlaine_i_teplohoda_odessa_star_4_aprelia_v_mramornom_more___mejdunarodnaia_uridicheskaia_slujba_foto.html

68. Cristina Commendatore.: Blockchain in trucking: What about the middlemen? 20.10.2017 URL: https://www.fleetowner.com/electronic-security/blockchain-trucking-what-about-middlemen

69. Robert Hackett.: IBM and Maersk Are Creating a New Blockchain Company. 16.01.2017 URL: https://fortune.com/2018/01/16/ibm-blockchain-maersk-company/

70. Winnesota.: HOW BLOCKCHAIN IS REVOLUTIONIZING THE WORLD OF TRANSPORTATION AND LOGISTICS. URL: https://www.winnesota.com/blockchain

71. Temperature Measurement in the Fish Industry. URL: http://www.fao.org/3/x5992e/X5992e01.htm

72. Yang, J., Chen, T., Wu, M., Xu, Z., Liu, X., Lin, H., Yang, M., Long, F., Zhang, L., Zhou, L. 2009. MODIST: transparent model checking of unmodified distributed systems. URL: https://www.usenix.org/legacy/events/nsdi09/tech/full_papers/yang/yang_html/index.html

73. Spin. URL: http://spinroot.com/spin/whatispin.html

74. Killian, C., Anderson, J. W., Jhala, R., Vahdat, A. 2006. Life, death, and the critical transition: finding liveness bugs in system code. URL: http://www.macesystems.org/papers/MaceMC_TR.pdf

75. Lamport, L., Yu, Y. 2011. TLC—the TLA+ model checker. URL: http://research.microsoft.com/en-us/um/people/lamport/tla/tlc.html

76. Wilcox, J. R., Woos, D., Panchekha, P., Tatlock, Z., Wang, X., Ernst, M. D., Anderson, T. 2015. Verdi: a framework for implementing and formally verifying distributed systems. Proceedings of the ACM SIGPLAN 2015 Conference on Programming Language Design and Implementation: 357-368. URL: https://homes.cs.washington.edu/~mernst/pubs/verify-distsystem-pldi2015-abstract.html

77. Towards Formal Reliability Analysis of Logistics Service Supply Chains using Theorem Proving Waqar Ahmed , Osman Hasan , and Sofi`ene Tahar EPiC Series in Computing Volume 40, 2016, Pages 1–14 IWIL-2015. 11th International Workshop on the Implementation of Logics

78. Y. Li and H. Yi. Research on the Inherent Reliability and the Operational Reliability of the Supply Chain. u- and e-Service, Science and Technology, 7(1):104–112, 2014.

79. BPMN (Business Process Model and Notation). www.bpmn.org

80. UML (Unified Modelling Language). www.uml.org

81. SysML (Systems Modelling Language). www.sysml.org

82. VDM (Vienna Development Method). www.vienna.cc/e/evdm.htm

83. SPIN. http://spinroot.com/spin/whatispin.html

84. APS (Algebraic Programming System).www.apsystem.org.ua

85. A. Letichevsky, O. Letychevskyi, and V. Peschanenko, “Insertion Modeling and Its Applications,”Computer Science Journal of Moldova, vol. 24, no. 3, 2016, pp. 357-370.

86. A. Letichevsky and D. Gilbert, “Interaction of agents and environments,”inRecent Trends in Algebraic Development Technique, LNCS 1827,Springer-Verlag, 1999.

87. Letychevskyi O,, Letichevsky A., Peschanenko V., and Weigert T., “Insertion modeling and symbolic verification of large systems,”I LNCS 9369 SDL 2015: Model-Driven Engineering for Smart Cities. Springer, 2015, pp. 3-18.

88. «Свойства предикатного трансформера системы VRS» / Летичевский А.А., Годлевский А.Б., Песчаненко В.С., Потиенко С.В. // Кибернетика и системный анализ – 2010. – № 4. – С. 3–16.

89. Щербина В.В. Проблеми та завдання розвитку портової логістики України // Розвиток методів управління та господарювання на транспорті: Зб. наук. праць, 2019. № 2 (67). С. 89-101. DOI: 10.31375/2226-1915- 2019-2-89-101.

90. Анісімов А.В., Заславський В.А., Фаль О.М. Основи інформаційної безпеки та захисту інформації у контексті євроатлантичної інтеграції України. // ДП «НВЦ» Євроатлантикінформ Київ, 2006. – 103 с.

91. ТЕХНОЛОГИЯ BLOCKCHAIN В ЛОГИСТИКЕ. 07.07.2017. URL: https://logist.fm/publications/tehnologiya-blockchain-v-logistike

92. Українські морські порти впровадять блокчейн-рішення на базі системи e-Port. 18 СЕРПНЯ 2018. URL: https://mind.ua/news/20187830-ukrayinski-morski-porti-vprovadyat-blokchejn-rishennya-na-bazi-sistemi-e-port

93. В.О. Горбатюк, С.О. Горбатюк. Methods of detection of HTTP attacks on a smart home using the algebraic methods // Проблеми програмування. 2022. № 3-4. Спеціальний випуск – С. 396-402. DOI: 10.15407/pp2022.03-04.396

94. В.О. Горбатюк, С.О. Горбатюк. Методи перевірки алгебраїчним співставленням спротиву HTTP-атакам на розумний будинок // Control systems and computers, 2022, № 4 – С. 13-23. DOI: 10.15407/csc.2022.04.013

ДОДАТКИ

Додаток А

Список публікацій здобувача

Основні наукові результати дисертації

1. О.О. Летичевський, С.О. Горбатюк. Децентралізовані системи в логістиці: огляд використання та проблеми безпеки // Проблеми програмування – 2020, №1. – С. 55-73. DOI: 10.15407/pp2020.01.055

2. О.О. Летичевський, С.О. Горбатюк, В.О. Горбатюк. Алгебраїчне моделювання в системах міжнародної та місцевої обслуговуючої логістики // Проблеми програмування – 2020, №4. – С. 88-97. DOI: 10.15407/pp2020.04.088

3. В.О. Горбатюк, С.О. Горбатюк. Methods of detection of HTTP attacks on a smart home using the algebraic methods // Проблеми програмування. 2022. № 3-4. Спеціальний випуск – С. 396-402. DOI: 10.15407/pp2022.03-04.396

4. В.О. Горбатюк, С.О. Горбатюк. Методи перевірки алгебраїчним співставленням спротиву HTTP-атакам на розумний будинок // Control systems and computers, 2022, № 4 – С. 13-23. DOI: 10.15407/csc.2022.04.013

5. С. Горбатюк. Блокчейн в логістиці та формальна верифікація властивостей безпеки // Ідеї академіка В.М. Глушкова і сучасні проблеми штучного інтелекту. 8-ма Всеукраїнська науково-практична конференція: “Глушковські читання” (29 жовтня 2019, Київ). – Київ, Видавничий дім Ліра-К, 2019. – С. 57–60.

6. Васюхін М., Касім А., Долинний В., Касім М., Шелестовський В., Горбатюк С. Метод створення класифікатора картографічної інформації для агрономічних автоматизованих систем // Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні ‘2017: збірник матеріалів V Міжнародної науково-практичної конференції (22 – 23 червня 2017 року, Київ). – К.: Компринт, 2017. – С. 47–49.

7. Васюхін М.І., Касім А.М., Долинний В.В., Касім М.М., Шелестовський В.Г., Горбатюк С.О. Геоінформаційна система для малих і середніх фермерських господарств // Безпека життєдіяльності на транспорті і виробництві – освіта, наука,

практика: тези доповідей IV Міжнародної науково-практичної конференції (14-16 вересня 2017 року, Херсон). – Херсон: ХДМА, 2017. – С. 324–330.

8. Васюхін М.І., Горбатюк С.О., Касім М.М., Шелестовський В.Г. комп'ютерні системи. Навчальний посібник. – К.: Компринт, 2017. – 270 сторінок. ISBN 978-966-929-552-1

9. Гусєв Б.С., Горбатюк С.О., Савицька Я.А., Смолій В.В., Шелестовський В.Г. Інформаційна технологія системи управління фермерським господарством. Монографія. – НУБіП України, 2018.- 220 сторінок.

10. Oleksandr Letychevskyi, Serhii Horbatiuk, Viktor Horbatiuk. Algebraic modelling of logistical systems equipped by wireless monitoring devices // The 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (17-18 September 2020, Dortmund, Germany). DOI: 10.1109/IDAACS-SWS50031.2020.9297092

11. Oleksandr Letychevskyi, Volodymyr Peschanenko, Maksym Poltoratskyi, Serhii Horbatiuk, Horbatiuk Viktor and Yuliia Tarasich. One Approach to Formal Verification of Distributed Ledger Technologies on the Blockchain Technologies Example // Conference on Mathematical Foundations of Informatics: Proceedings MFOI-2020 (12–16 January 2021) – Taras Shevchenko National University of Kyiv - pages 227-242.

12. Oleksandr Letychevskyi, Volodymyr Peschanenko, Sergiy Horbatiuk. Consensus Protocol Security Analysis Using an Algebraic Virtual Machine // IntelITSIS'2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security (23–25 March, 2022) – Khmelnytskyi, Ukraine - pages 484-493.

Апробація матеріалів дослідження
на конференціях, семінарах, круглих столах тощо

Рівень заходу та його назва	**Місце й дата проведення**	**Форма участі**
Міжнародна наукова конференція «Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні»	Червень 2017р., Київ	Очна, 1 публікація в матеріалах конференції
Міжнародна наукова конференція «Безпека життєдіяльності на транспорті і виробництві – освіта, наука, практика»	Вересень 2017р., Херсон	Очна, 1 публікація в матеріалах конференції
Міжнародна наукова конференція «Ідеї академіка В.М. Глушкова та сучасні проблеми штучного інтелекту»	Листопад 2019р., Київ	Очна, 1 публікація в матеріалах конференції
Міжнародна наукова конференція «The 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems»	Вересень 2020р., Дортмунд, Німеччина	Очна, 1 публікація в науковому виданні, що індексується Scopus
Міжнародна наукова конференція «Conference on Mathematical Foundations of Informatics: Proceedings MFOI-2020»	Січень 2021р., Київ	Очна, 1 публікація в матеріалах конференції
Міжнародна наукова конференція «IntelITSIS’2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security»	Березень 2022р., Хмельницький	Очна, 1 публікація в матеріалах конференції

Формалізація блокчейну

Додаток Б

BN: real

N: real

Node: (real) -> NODE

TIME_SLOT: real

- Атрибути середовища

Початкові умови середовища:

N == 5 &&

Forall (i:real) ((1 <= i <= N) && (Node(i).CHAINS == 1) && (Node(i).HANG(1) == 0) && (Node(i).MAX == 1) && (Node(i).MIN == 1) && (Node(i).LEN(1) == 0) && (BN == 0))

Поведінка:

rs(i, b)(

B0 =((B(1) || B(2) || B(3) || B(4) || B(5));(nextTimeSlot.B0 + !nextTimeSlot.Delta)),

B(i) = (CreateMode(i);ReceiveMode(i)),

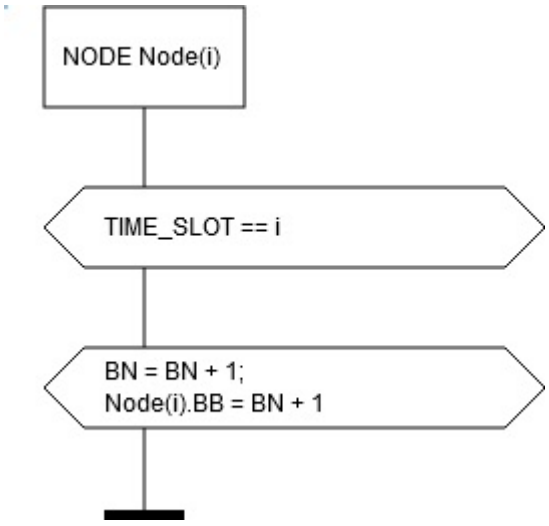
CreateMode(i) = createBlock(i).sendBlock(i),

ReceiveMode(i) = ((receiveBlock(i).checkForking(i).(doForking(i) + doHang(i)));(checkMin(i).checkMax(i));(ReceiveMode(i) + sendBlock(BB(i),i));(ReceiveMode (i) + end(i))),

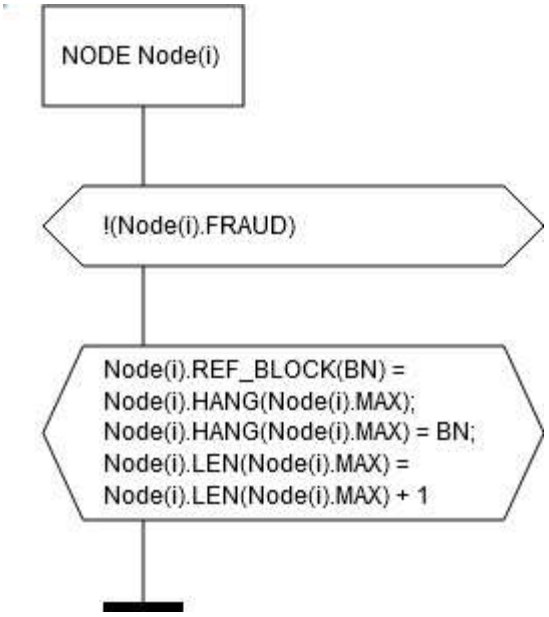
CreateBlock(i) = createBlock(i).((createReferenceFair(i).sendBlock(BN,i) + createReferenceFraud(i));(checkMin(i).checkMax(i)))

)

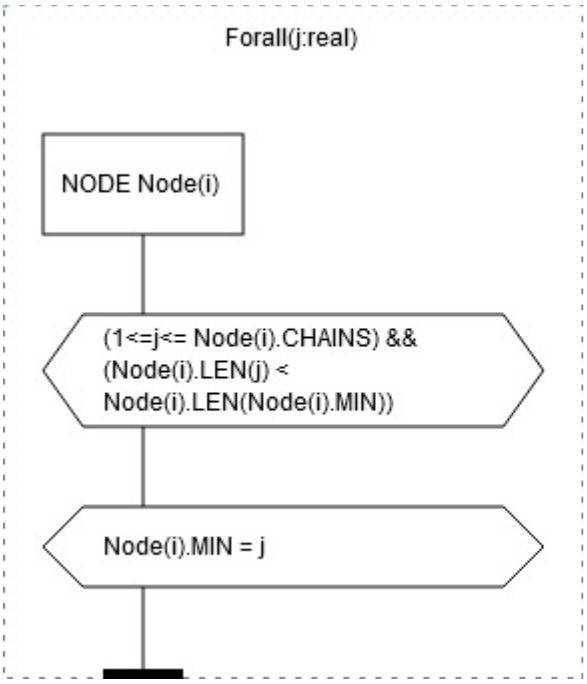
Базові протоколи



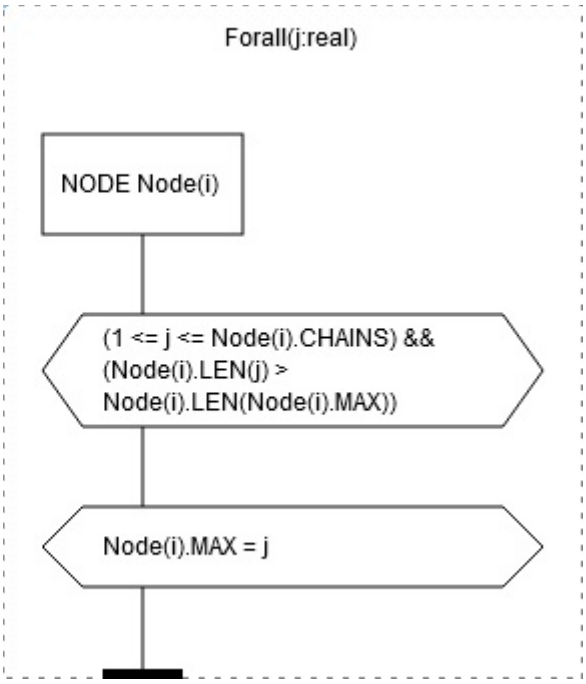
протокол дії createBlock



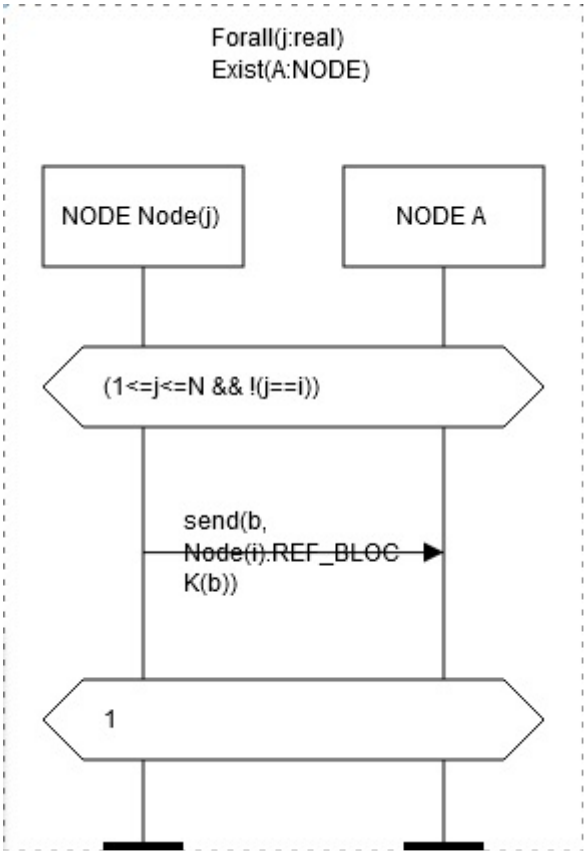
протокол дії createReferenceFair



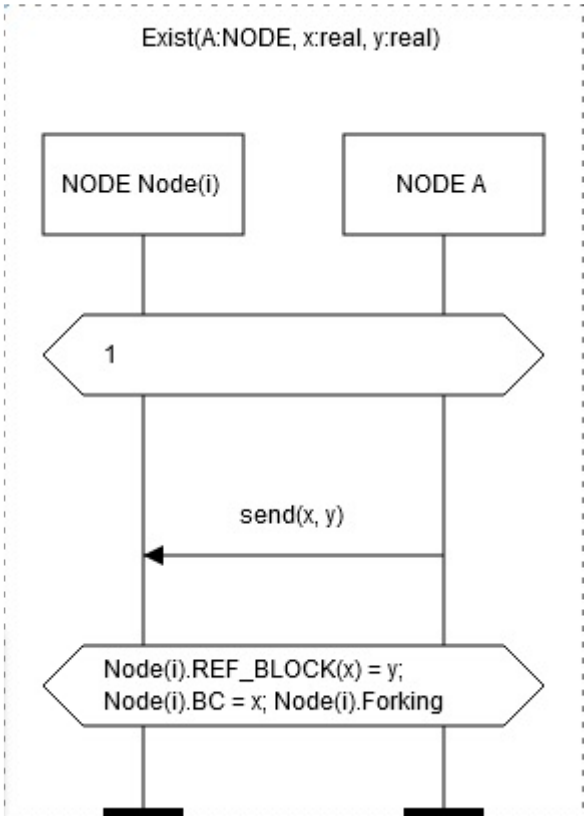
протокол дії checkMin



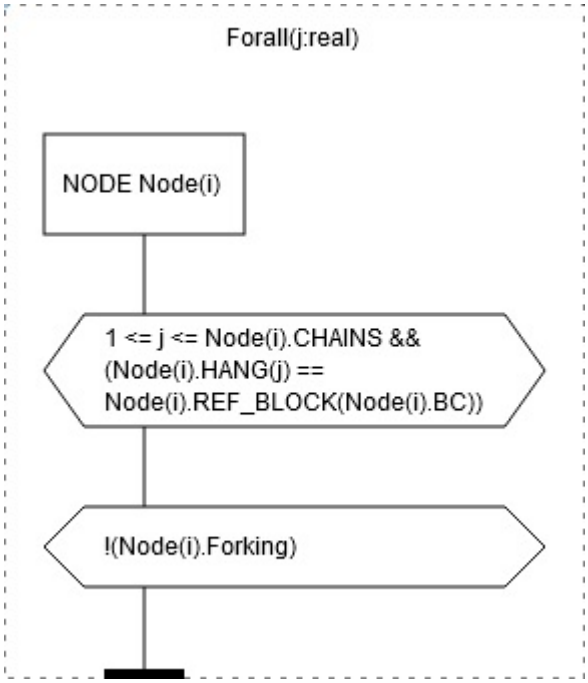
протокол дії checkMax



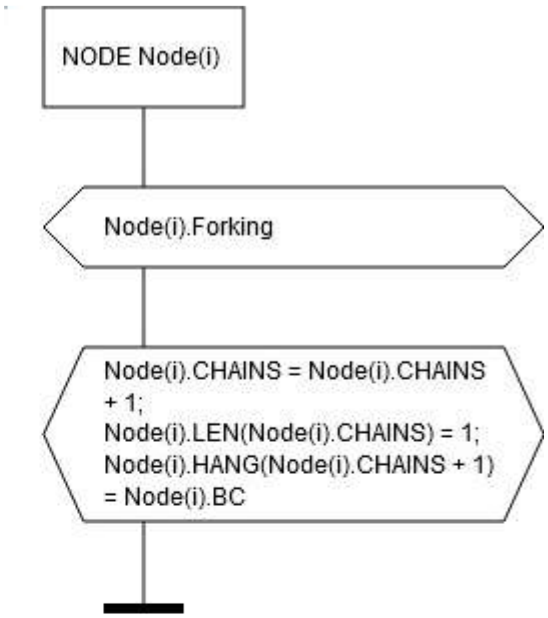
протокол дії sendBloc



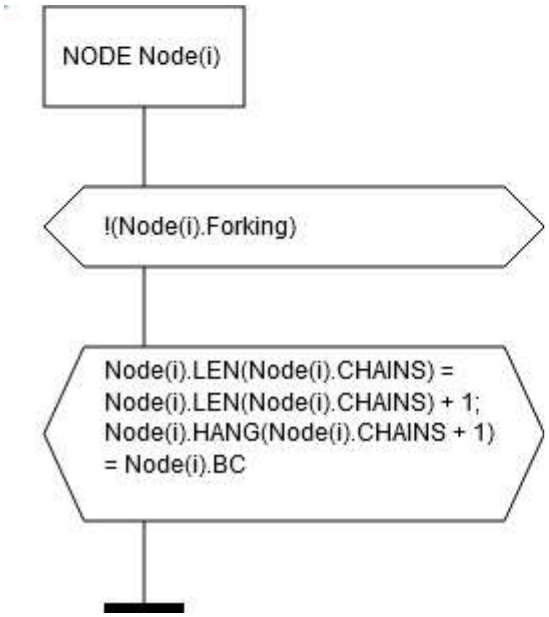
протокол дії receiveBlock



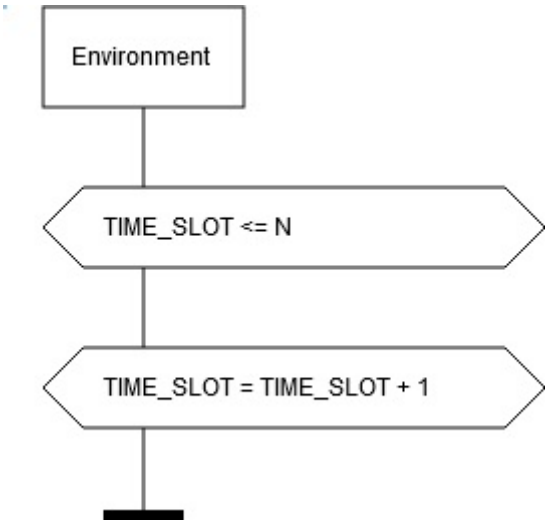
протокол дії checkForking



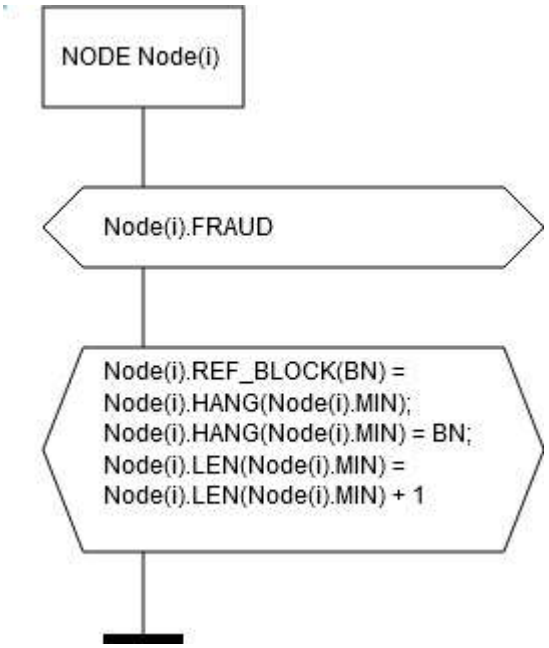
протокол дії doForking



протокол дії doHang



протокол дії nextTimeSlot



протокол дії createReferenceFraud

User: (real) -> User - атрибут середовища

Початкові умови середовища:

Forall (i:int) (0 < i <= 2 && User(i).money == 20000 && SimpleDAO.credit(i) == 0)

Поведінка:

rs(x, value)(

SP = User1(1, 10000) || User2(2, 10000),

User1(x, value) = donate(x, value).donate(x, value).withdraw1(x, value).fallback1(x, value).withdraw2(x, value),

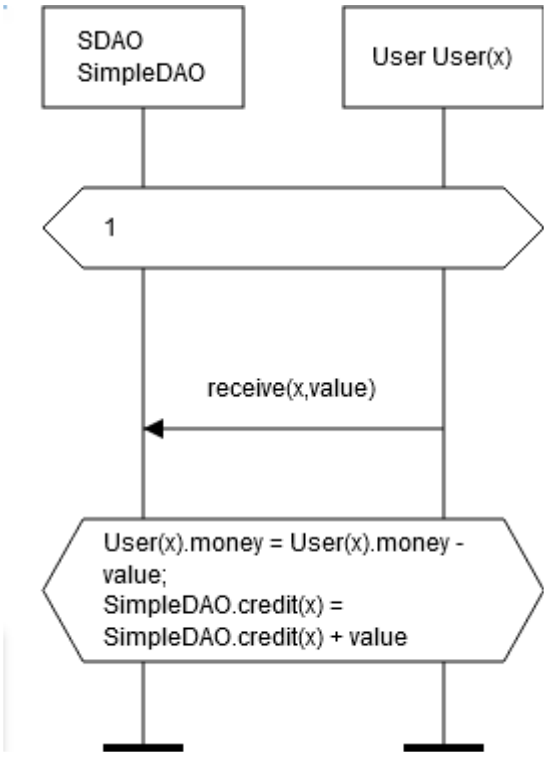
User2(x, value) = donate(x, value).withdraw1(x, value).fallback2(x, value).withdraw2(x, value),

fallback1(x, value) = empty,

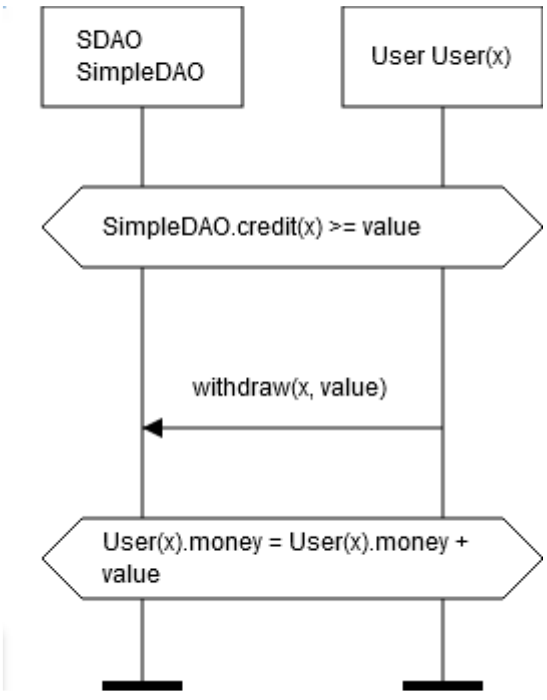
fallback2(x, value) = withdraw1(x, value).withdraw1(x, value).withdraw1(x, value)

)

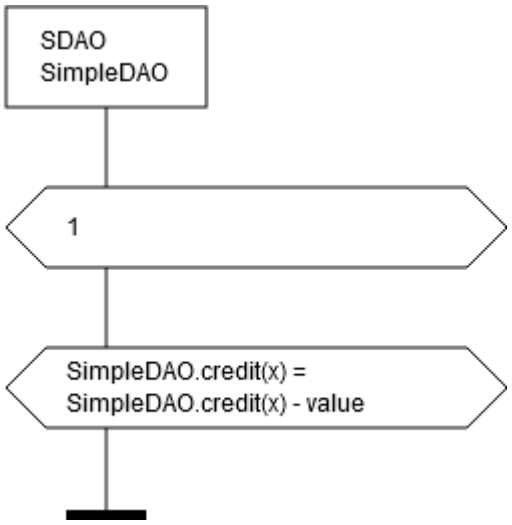
Базові протоколи



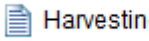
протокол дії donate



протокол дії withdraw1



протокол дії withdraw2



- атрибут середовища

Початкові умови середовища:

!Harvesting &&

Forall (f:Field) (f.state == fGROW && f.X < f.Xe && f.Y < f.Ye) &&

Forall (c:Combine) (c.state == cSTOP && c.X == W.X && c.Y == W.Y &&

c.tankCap == 365 && c.tankLevel == c.tankCap &&

c.Sp == 6 && c.Fc == 0.01 && c.Wd == 1) &&

Forall (t:Truck) (t.state == tSTOP && t.X == W.X && t.Y == W.Y &&

t.tankCap == 300 && t.tankLevel == t.tankCap &&

t.loadCap == 9000 && t.loadLevel == 0 &&

t.Sp == 36 && t.Fc == 0.004 && t.Wd == 1) &&

Forall (ft:FuelTruck) (ft.state == ftSTOP && ft.X == W.X && ft.Y == W.Y &&

ft.tankCap == 300 && ft.tankLevel == ft.tankCap &&

ft.fuelCap == 1500 && ft.fuelLevel == ft.fuelCap &&

ft.Sp == 36 && ft.Fc == 0.004) &&

Forall (m:Mechanic) (m.state == mSTOP && m.X == W.X && m.Y == W.Y &&

m.tankCap == 80 && m.tankLevel == m.tankCap &&

m.Sp == 50 && m.Fc == 0.0001) &&

W.capacity == 300000 && W.stored == 0 && W.fuelUsed == 0 &&

W.oilLoad == 0 && W.cycternLoaded == 0 &&

PL.state == pSTOP

Поведінка:

rs(c1, f1, t1)(

/* Environment behavior */

SP = FieldsGotReady.FieldReady.

AgentsGetReady.CgetReady.TgetReady.FTgetReady.MgetReady.

(Link(C1, F1, T1) || Link(C2, F2, T2) || Link(C3, F3, T3) || Link(C4, F4, T4)),

Link(c1, f1, t1) = CombineToField(c1, f1).CgetTask(t1).TgetTask.Conway.Tonway.Tonplace.Conplace.TstartWork.(CWork || TWork).CfinishField.FieldDone.TfinishField.(ConwayBack.Creturned || TonwayBack.Treturned).Delta,

/* Combine behavior */

Conway = Cmove.Conway + Cmove.Crepair.Conway + Cmove.Crefuel.Conway,

Cmove = ConwayX.Cmove + ConwayY.Cmove,

ConwayBack = Creturn.ConwayBack + Creturn.Crepair.ConwayBack + Creturn.Crefuel.ConwayBack,

Creturn = CreturnX.Creturn + CreturnY.Creturn,

CWork = Cgath.CWork + Cgath.Crepair.CWork + Cgath.Crefuel.CWork + Cgath.Cawait.CWork,

Cgath = Cwork.Cgath + Cwork.CchangeWd.Cgath,

Crepair = (CneedRepair.Mwork.CgetFixed + CneedRepairOnField.TgetAwait.Mwork.CgetFixedOnField.TgetContinue) || M,

Crefuel = (CneedFuel.FTwork.CgetFueled + CneedFuelOnField.TgetAwait.FTwork.CgetFueledOnField.TgetContinue) || FT,

Cawait = (TneedFuelOnField + TneedRepairOnField + TneedUnload).CgetAwait.((Mwork + FTwork).(TgetFixedOnField + TgetFueledOnField) + TUnload).CgetContinue,

/* Truck behavior */

Tonway = Tmove.Tonway + Tmove.Trepair.Tonway + Tmove.Trefuel.Tonway,

Tmove = TonwayX.Cmove + TonwayY.Tmove,

TonwayBack = Treturn.TonwayBack + Treturn.Trepair.TonwayBack + Treturn.Trefuel.TonwayBack,

Treturn = TreturnX.Treturn + TreturnY.Treturn,

TWork = Tgath.TWork + Tgath.Trepair.TWork + Tgath.Trefuel.TWork + Tgath.Tawait.TWork + Tgath.TneedUnload.CgetAwait.TUnload.CgetContinue.TWork,

Tgath = Twork.Tgath + Twork.TchangeWd.Tgath,

Trepair = (TneedRepair.Mwork.TgetFixed + TneedRepairOnField.CgetAwait.Mwork.TgetFixedOnField.CgetContiunue) || M,

Trefuel = (TneedFuel.FTwork.TgetFueled + TneedFuelOnField.CgetAwait.FTwork.TgetFueledOnField.CgetContinue) || FT,

Tawait = (CneedFuelOnField + CneedRepairOnField).TgetAwait.(Mwork + FTwork).(CgetFixedOnField + CgetFueledOnField).TgetContinue,

TUnload = TonwayBack.Treturned.Tunload.WgetCrop.WgiveFuel.Tonway.Tonplace.TmoveToCombine.TresumeWork,

TmoveToCombine = Taproach.TmoveToCombine + Taproach.Trepair.TmoveToCombine + Taproach.Trefuel.TmoveToCombine,

Taproach = TmoveToCombineX.Taproach + TmoveToCombineY.Taproach,

/* Mechanic behavior */

M = Mreturn.(Mreturned + Mreturned.Mwork + Mwork),

Mwork = MgetCall.Monway.Monplace.Mfixed,

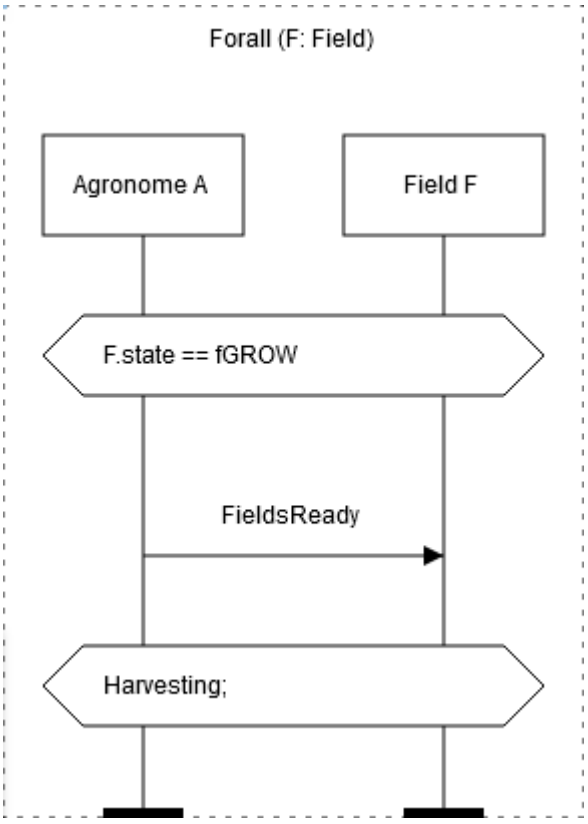
MgetCall = MgetCallC + MgetCallT + MgetCallFT + MgetCallPL,

Mfixed = MfixedC + MfixedT + MfixedFT + MfixedPL,

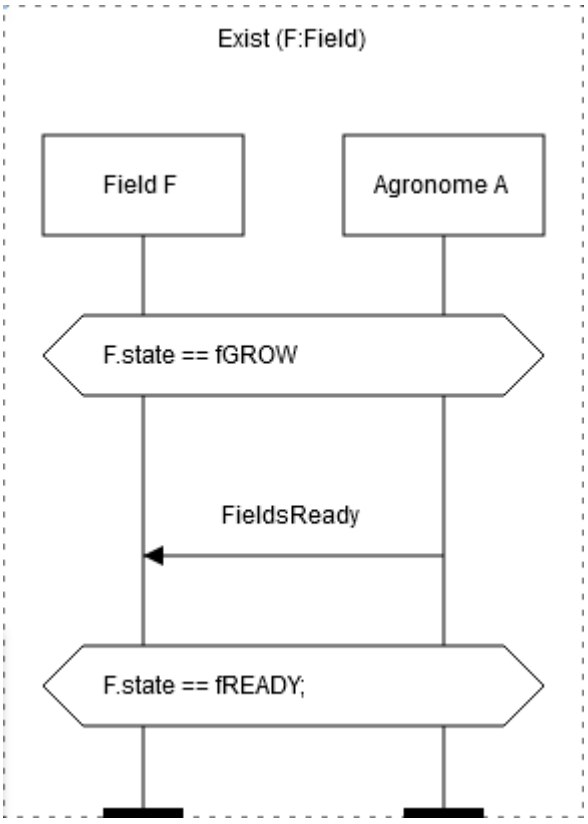
Monway = Mmove.Monway + Mmove.Mrefuel.Monway + Mmove.Mrepair.Monway,

```
Mreturn = Mmove.Mreturn + Mmove.Mrepair.Mreturn +
Mmove.Mrefuel.Mreturn,

Mmove = MonwayX.Mmove + MonwayY.Mmove,

Mrepair = MneedRepair.MgetFixed,

Mrefuel = MneedFuel.FTwork.MgetFueled,

/* FuelTruck behavior */

FT = FTreturn.(FTreturned.Wgivefuel + FTreturned.WgiveFuel.FTwork + FTwork),

FTwork = FTgetCall.FTonway.FTonplace.FTfueled,

FTgetCall = FTgetCallC + FTgetCallT + FTgetCallM,

FTfueled = FTfueledC + FTfueledT + FTfueledM,

FTonway = FTmove.FTonway + FTmove.FTrefuel.FTonway +
FTmove.FTrepair.FTonway,

FTreturn = FTmove.FTreturn + FTmove.FTrepair.FTreturn +
FTmove.FTrefuel.FTreturn,

FTmove = FTonwayX.FTmove + FTonway.FTmove,

FTrepair = FTneedRepair.Mwork.FTgetFixed,

FTrefuel = FTneedFuel.FTgetFueled,

/* Warehouse behavior */

W = (WgetCrop + WgetOil + WgiveFuel + WstartProd + WoilToDeliver +
WoilDelivered).W,

Wprod = WcropToProd.Wprod,

WstartProd.Wprod.WstopProd

/* ProductionLine behavior */

PL = PLstartWork.PLwork.PLstopWork,

PLwork = PLgetCrop.PLwork + PLgetCrop.PLneedRepair.PLgetFixed.PLwork

)
```

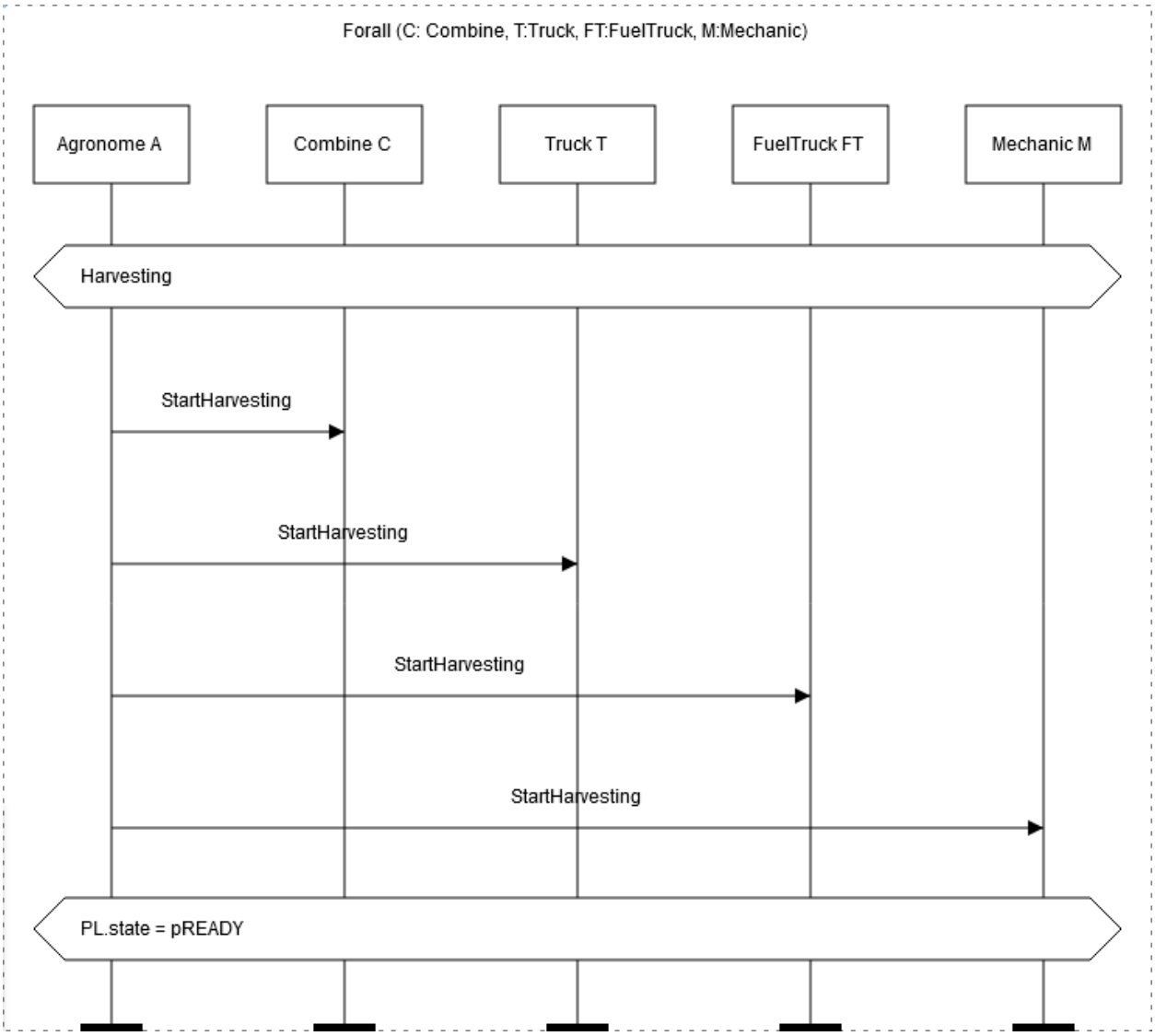
Базові протоколи



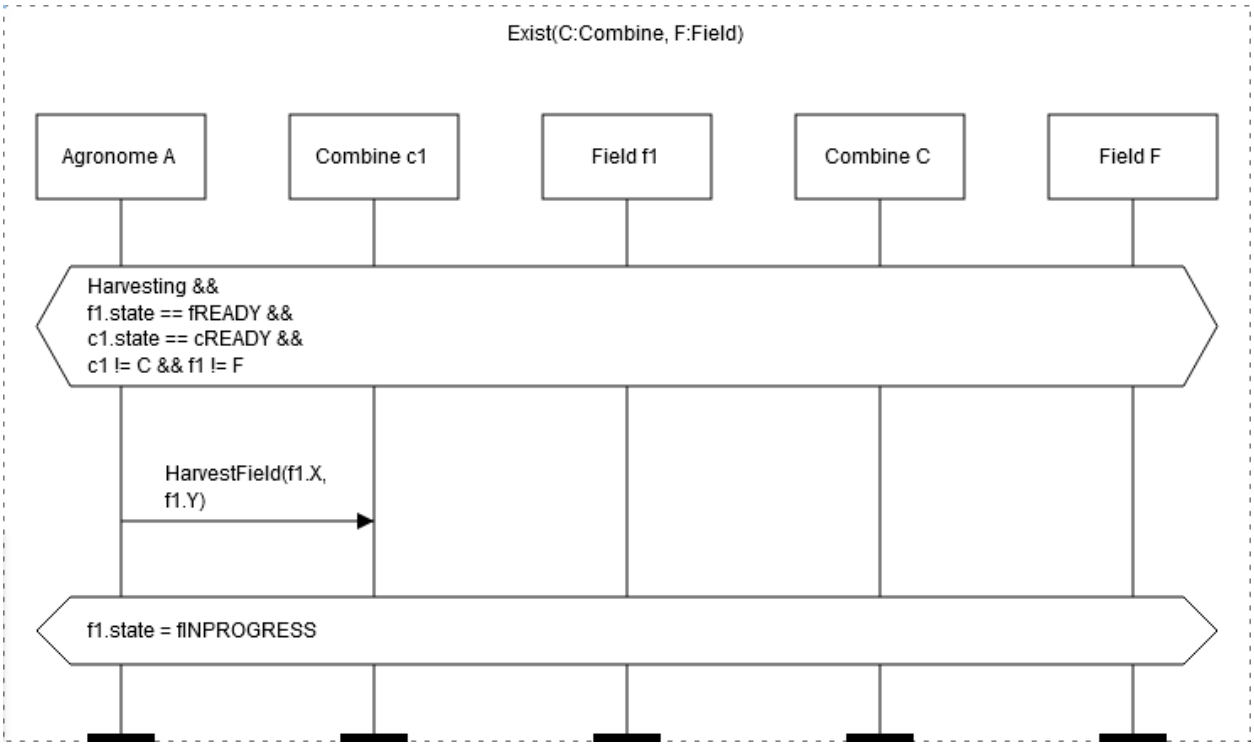
протокол дії FieldsGotReady



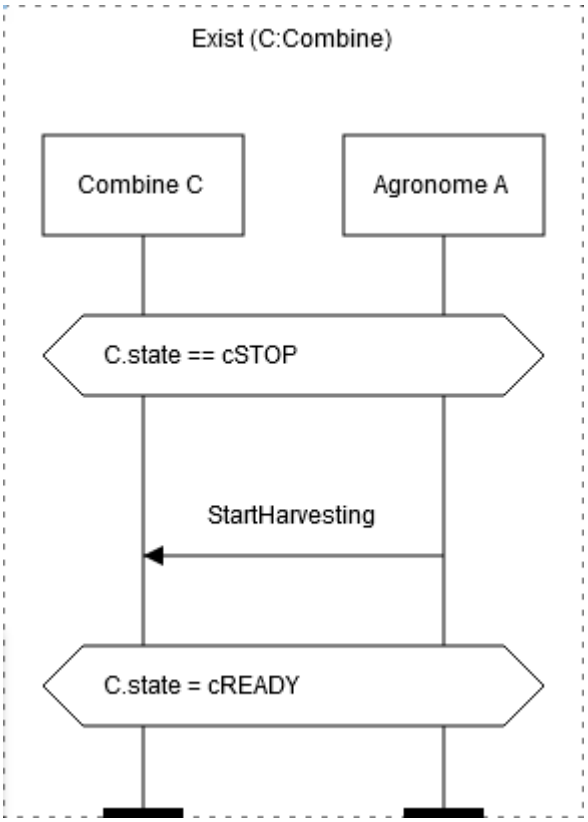
протокол дії FieldReady



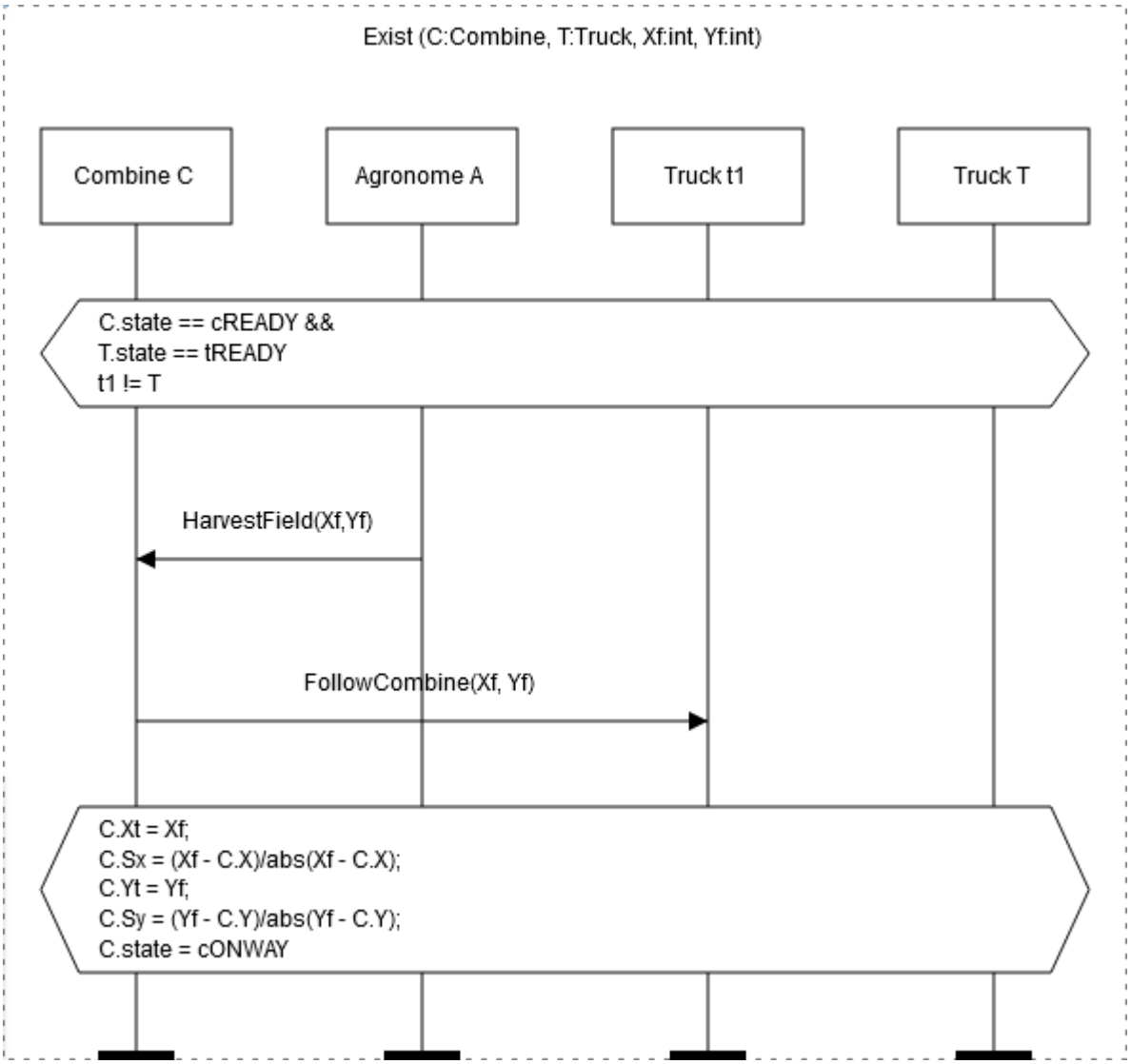
протокол дії AgentsGetReady



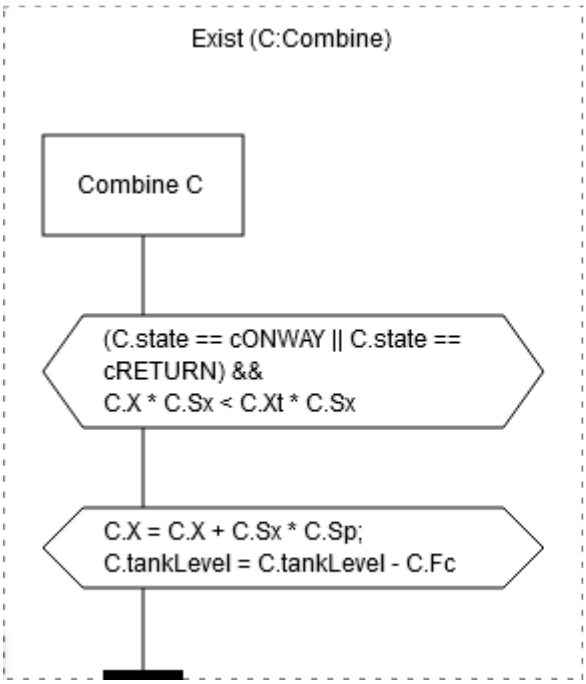
протокол дії CombineToField



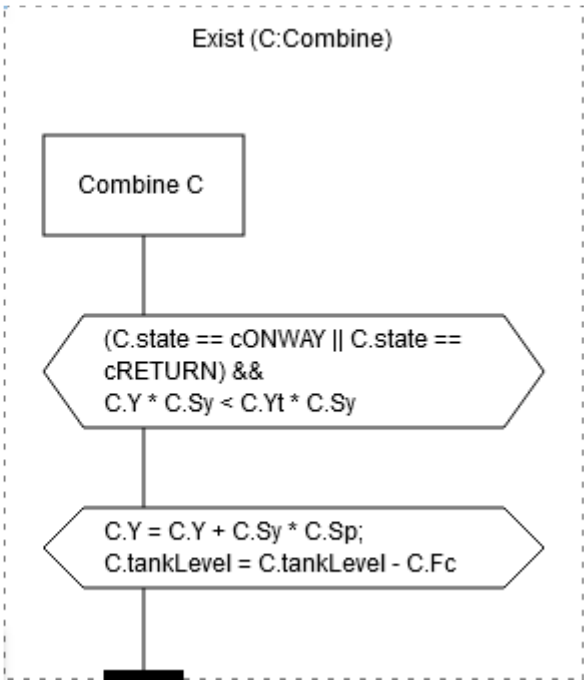
протокол дії CgetReady



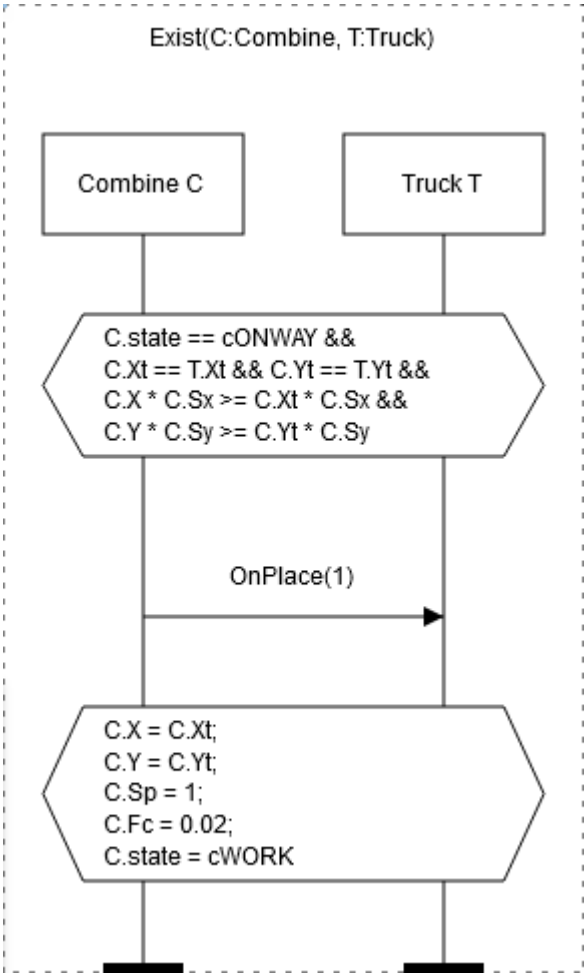
протокол дії CgetTask



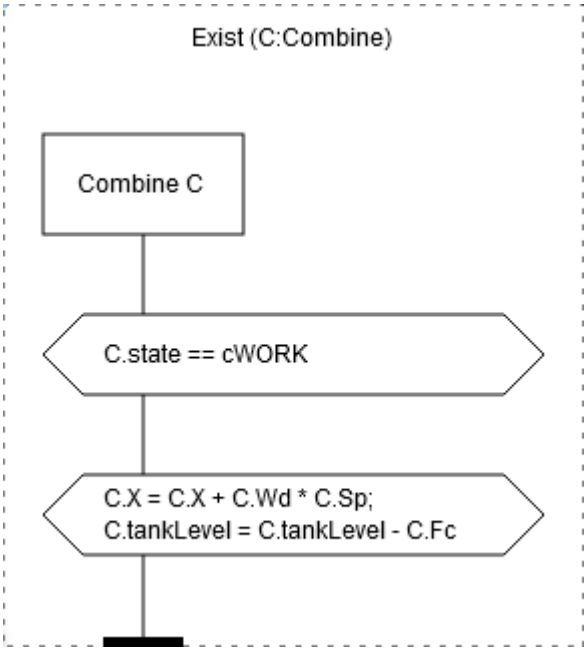
протокол дії ConwayX



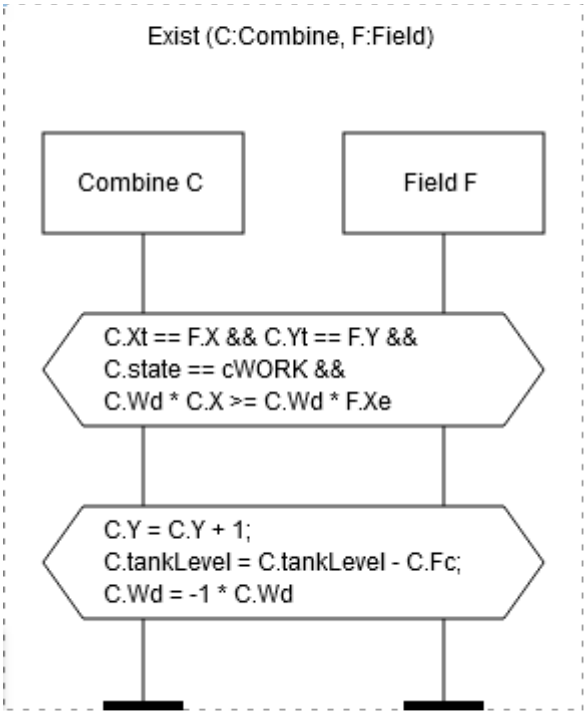
протокол дії ConwayY



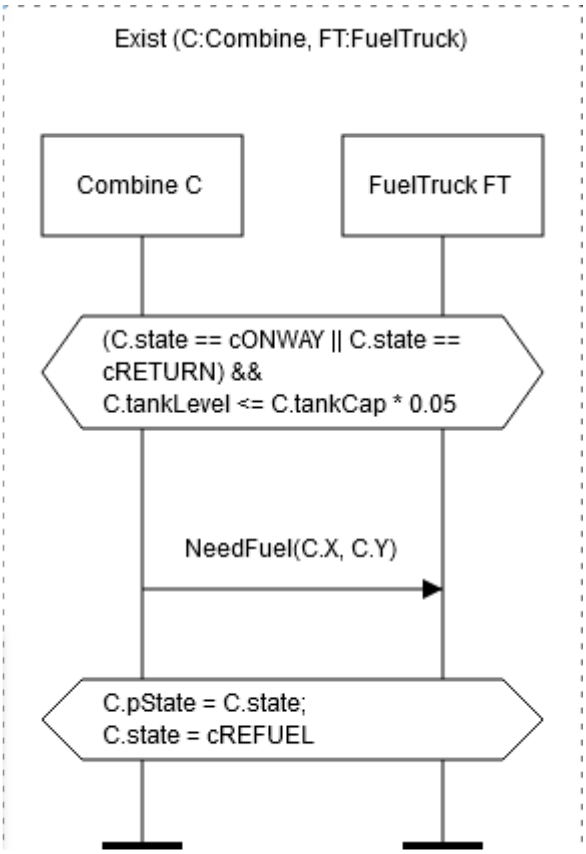
протокол дії Conplace



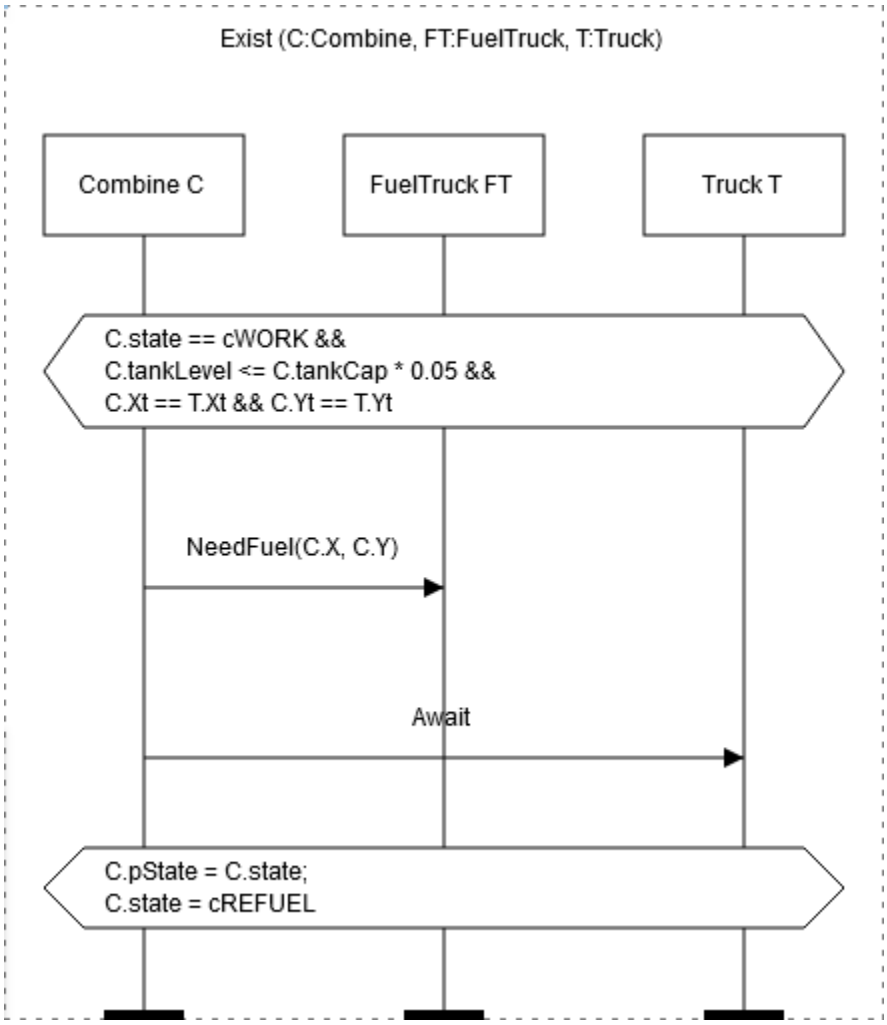
протокол дії Cwork



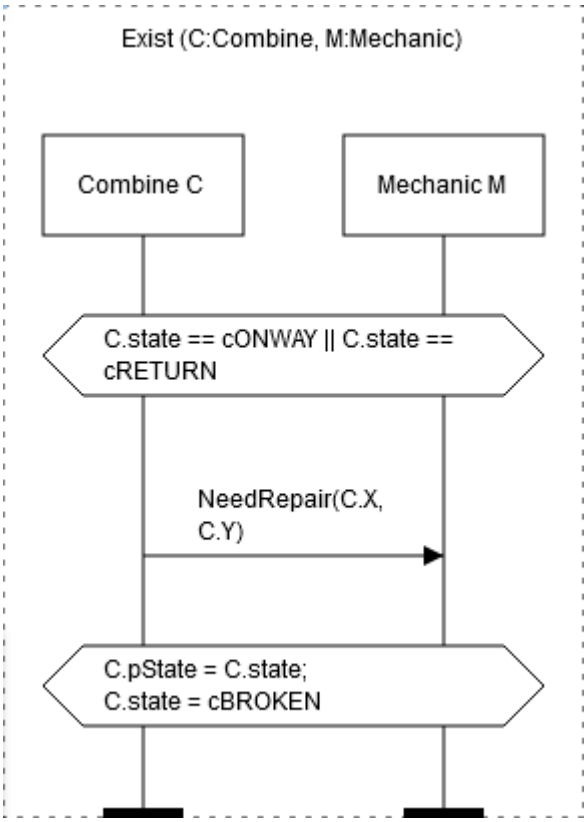
протокол дії CchangeWd



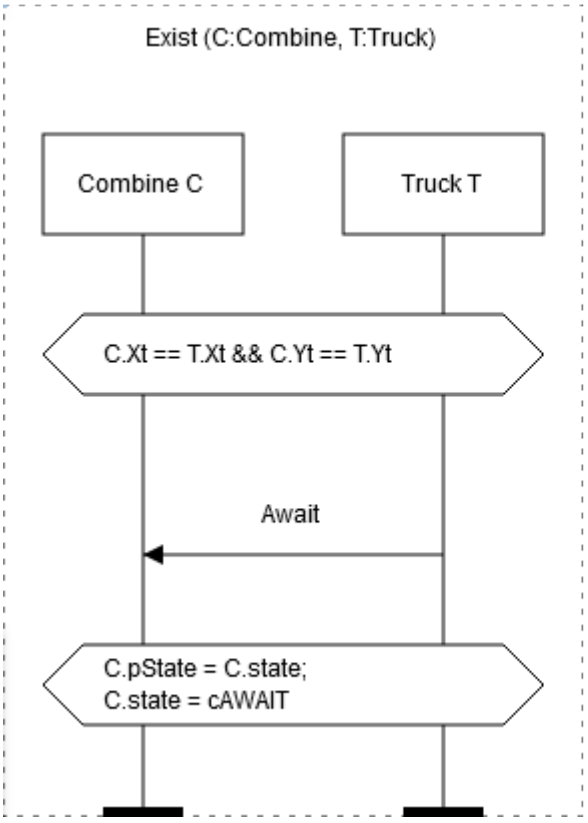
протокол дії CneedFuel



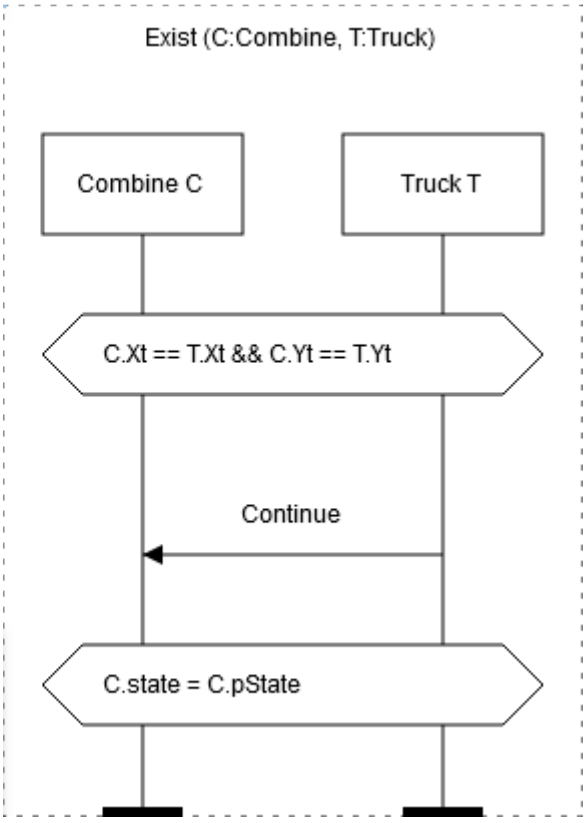
протокол дії CneedFuelOnField



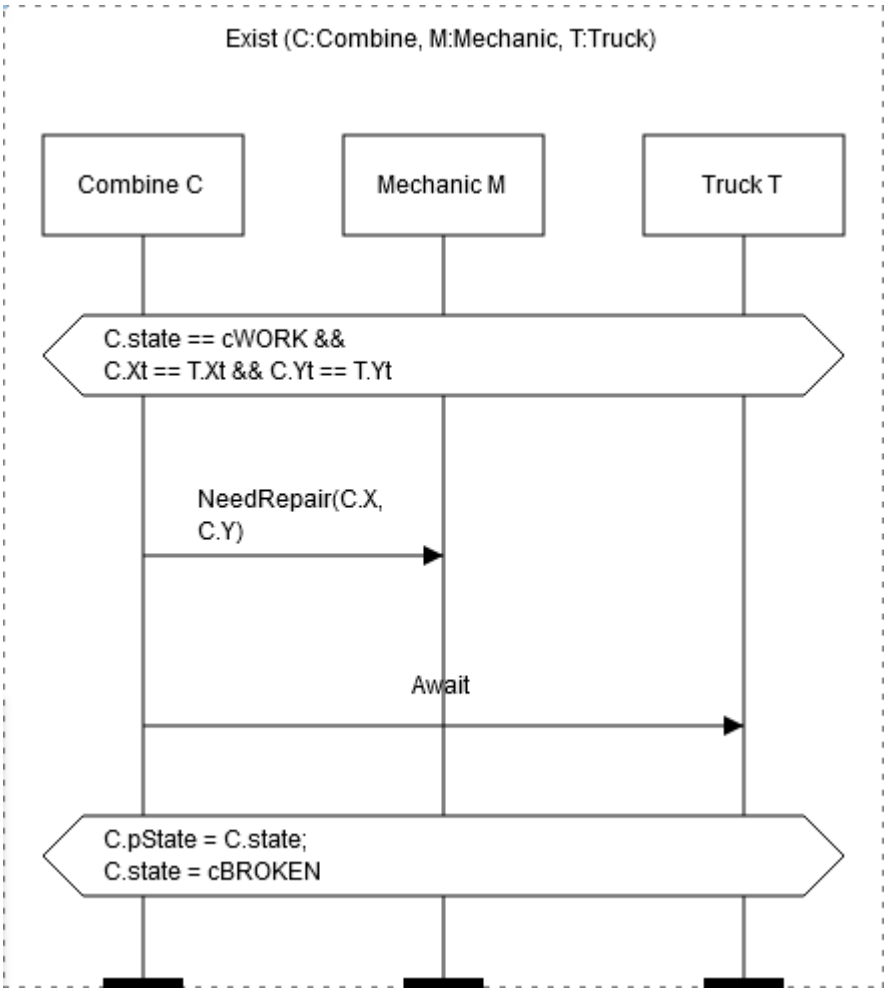
протокол дії CneedRepair



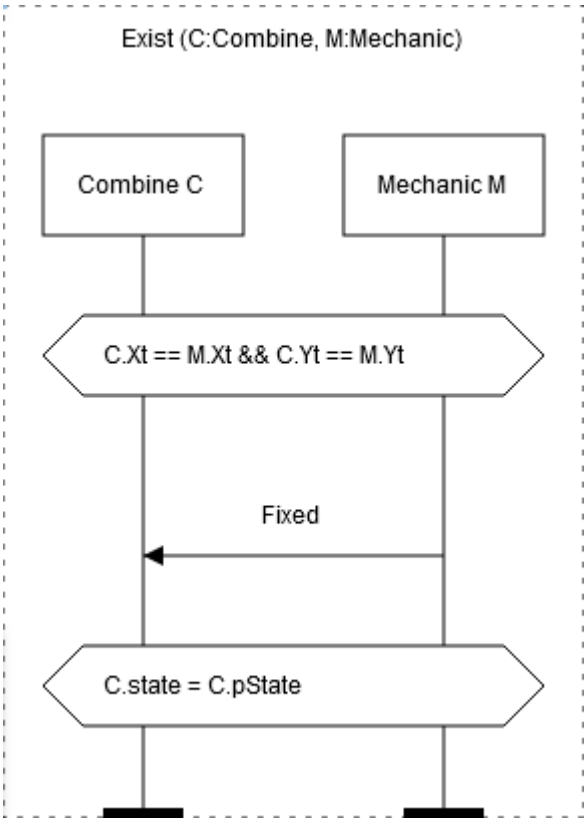
протокол дії CgetAwait



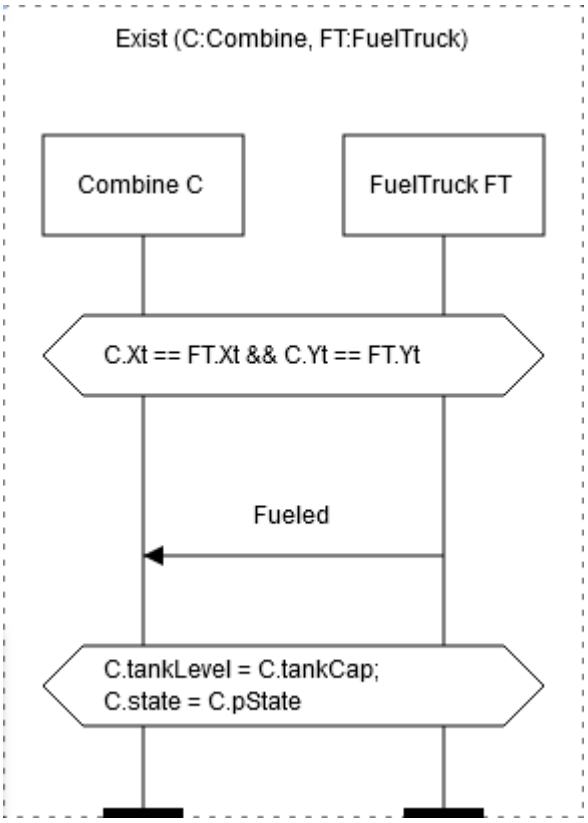
протокол дії CgetContinue



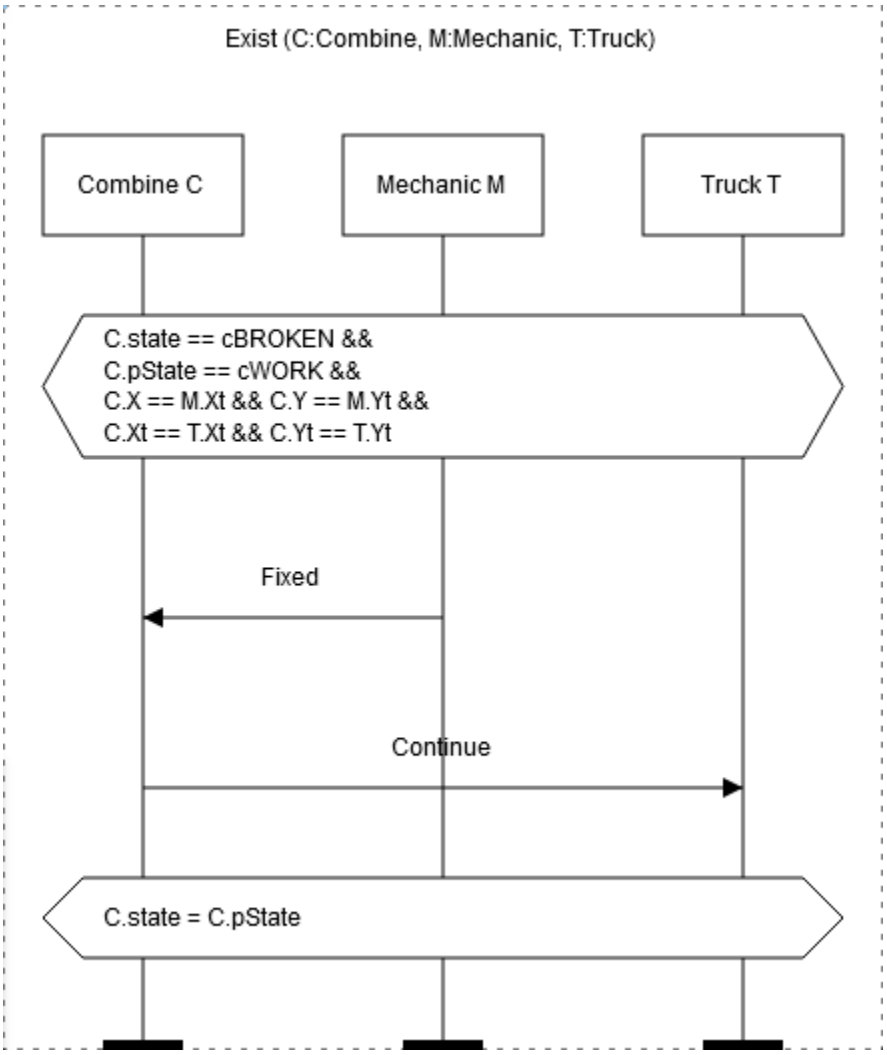
протокол дії CneedRepairOnField



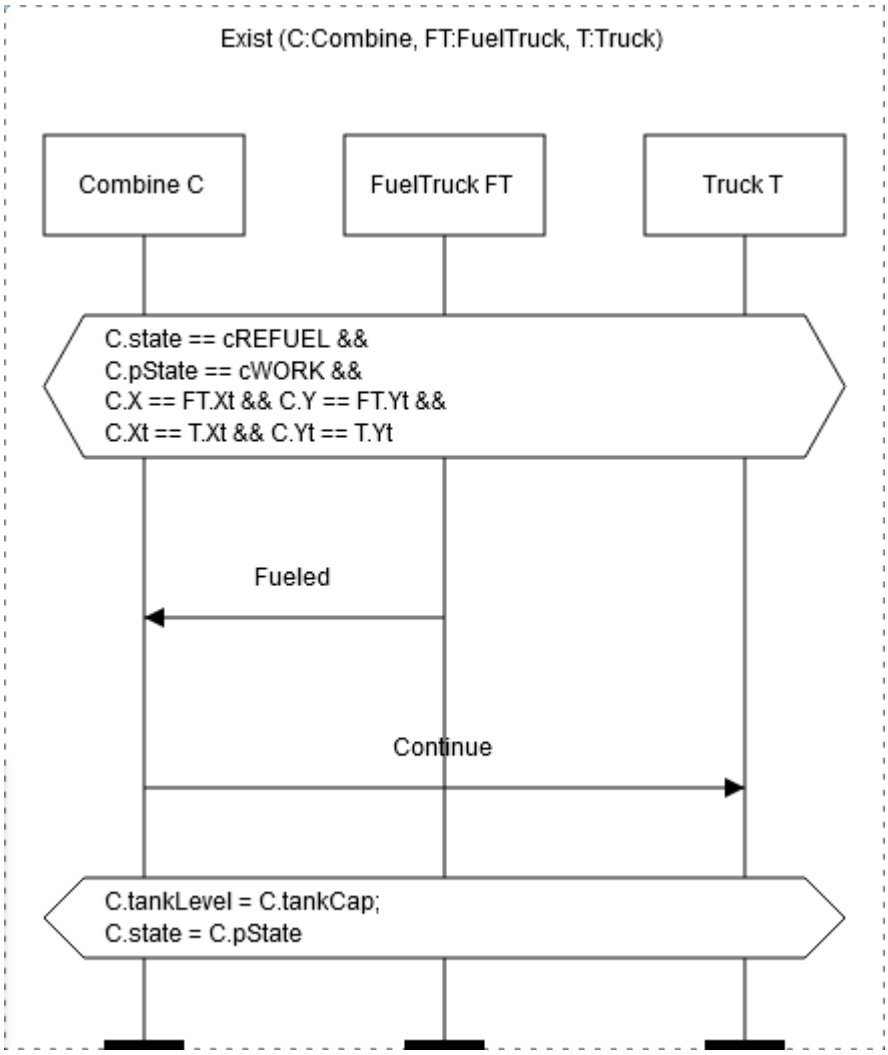
протокол дії CgetFixed



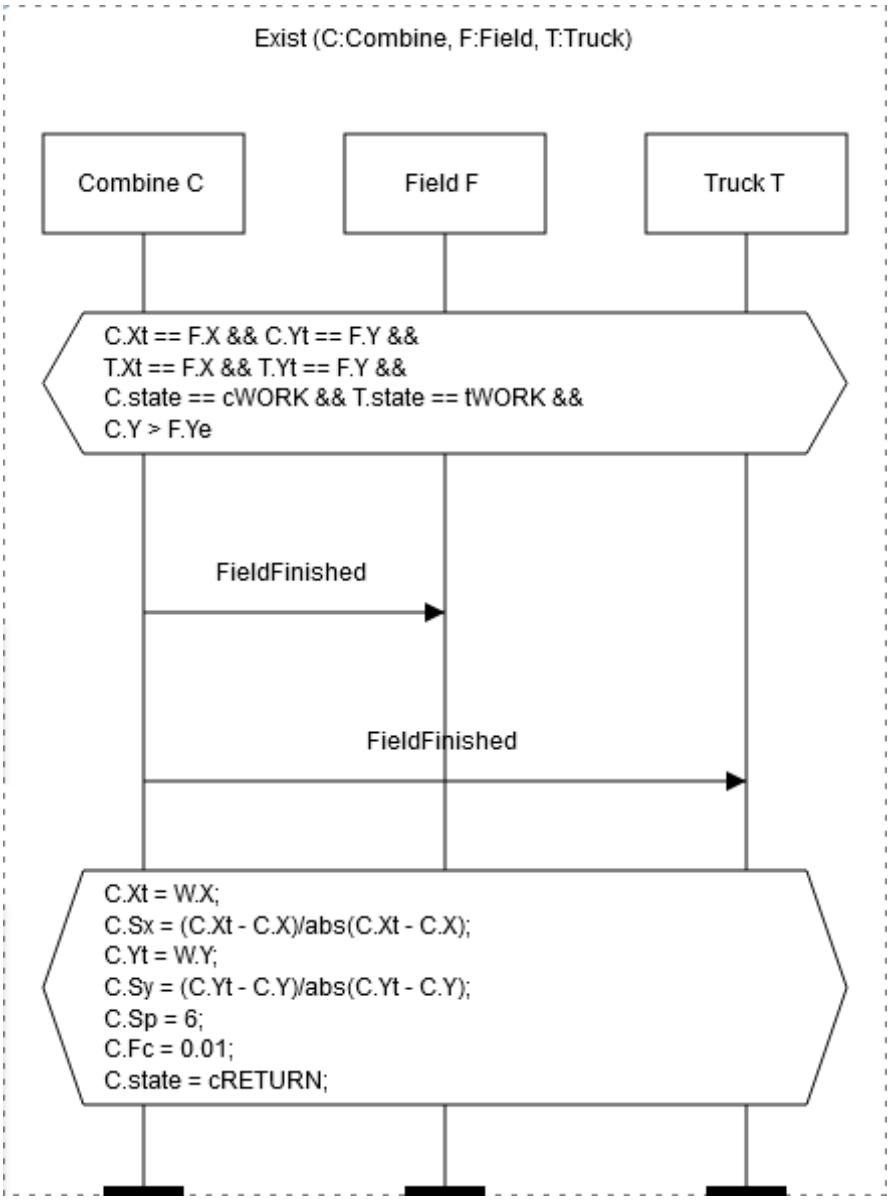
протокол дії CgetFueled



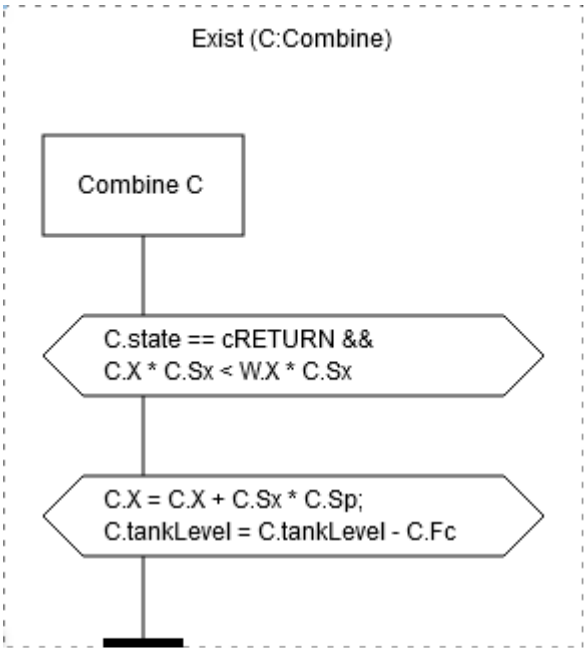
протокол дії CgetFixedOnField



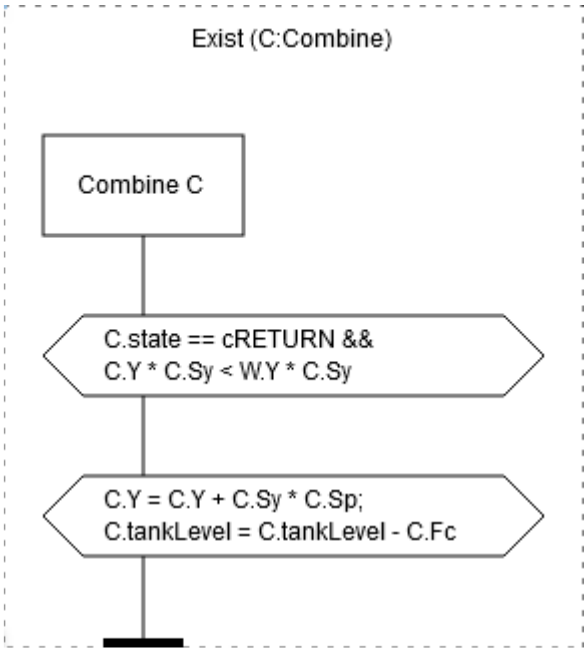
протокол дії CgetFueledOnField



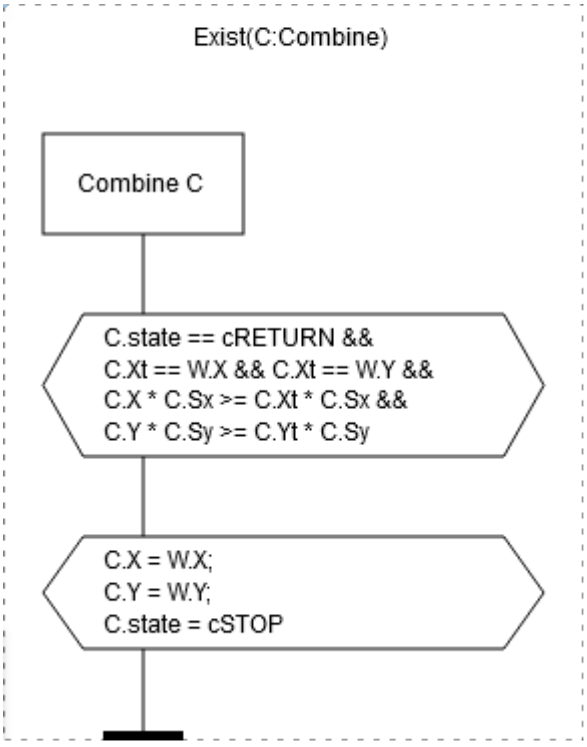
протокол дії CfinishField



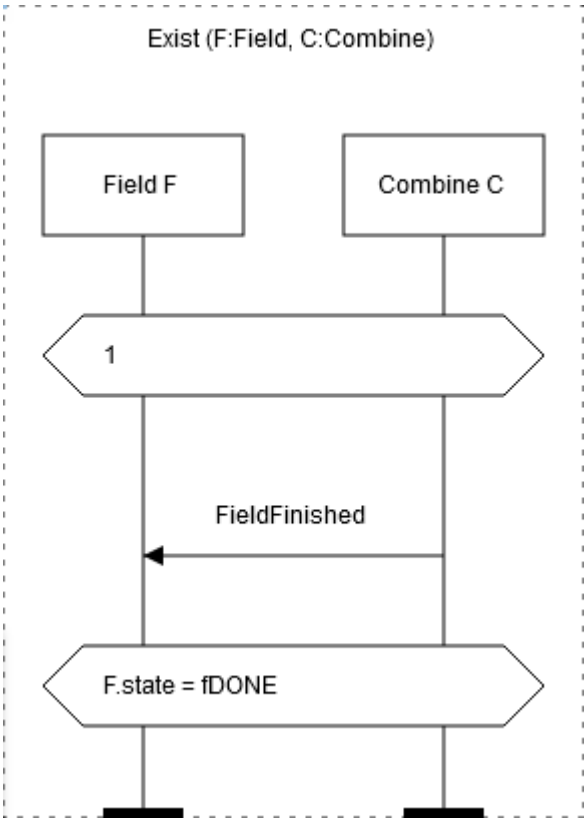
протокол дії CreturnX



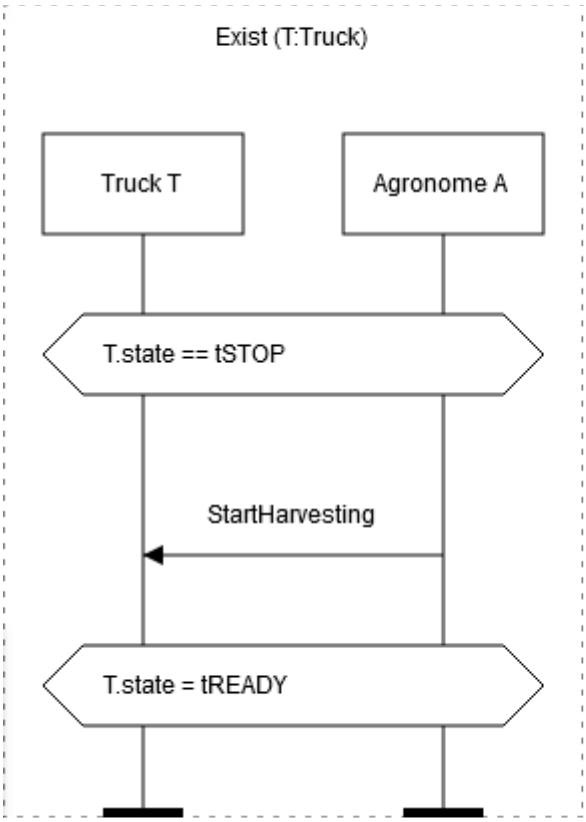
протокол дії CreturnY



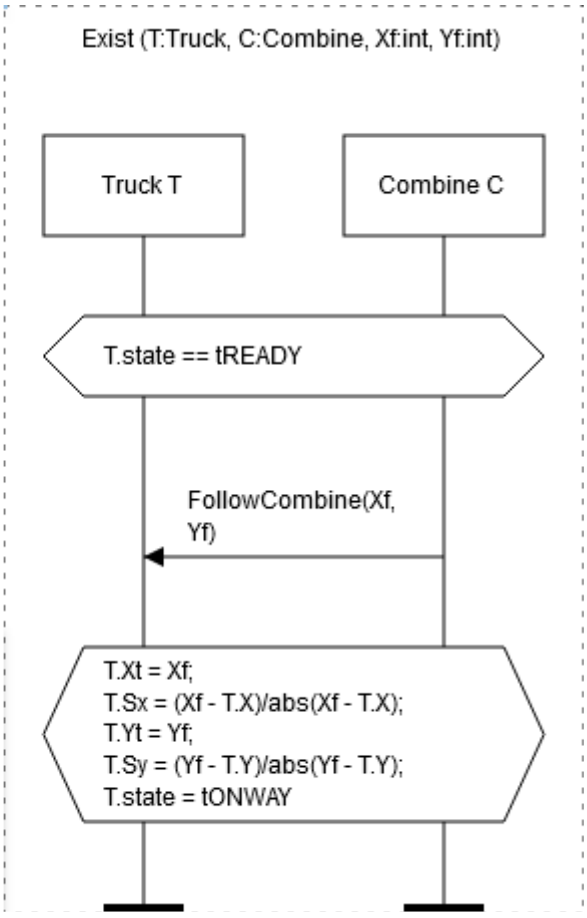
протокол дії Creturned



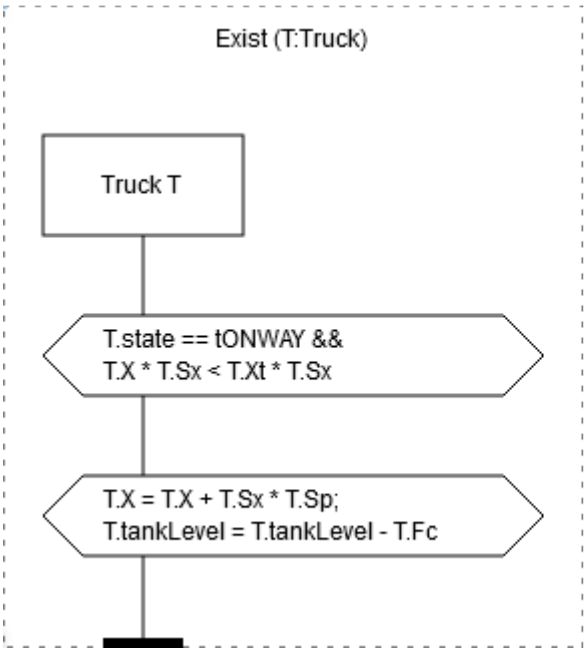
протокол дії FieldDone



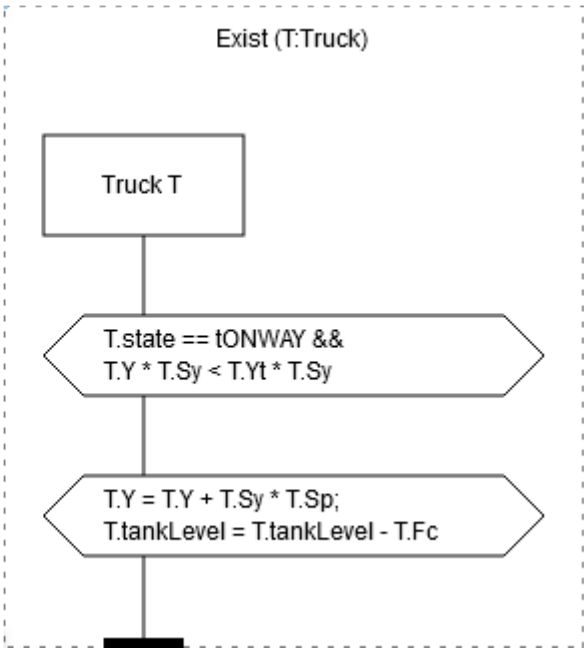
протокол дії TgetReady



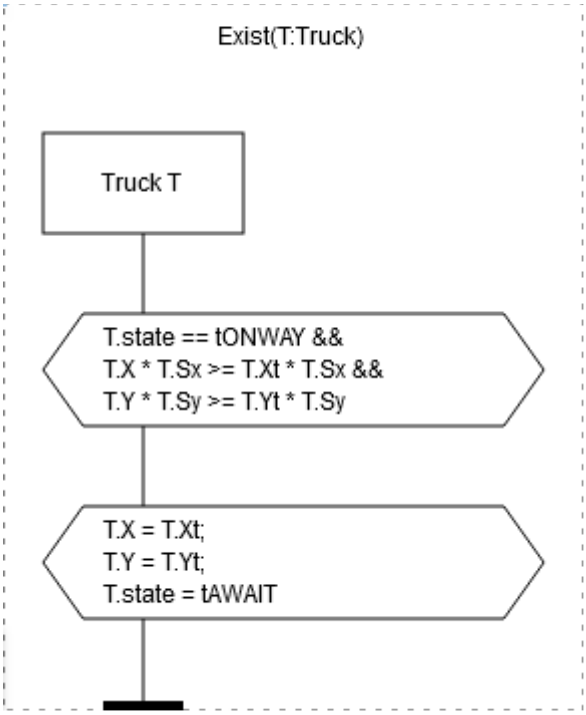
протокол дії TgetTask



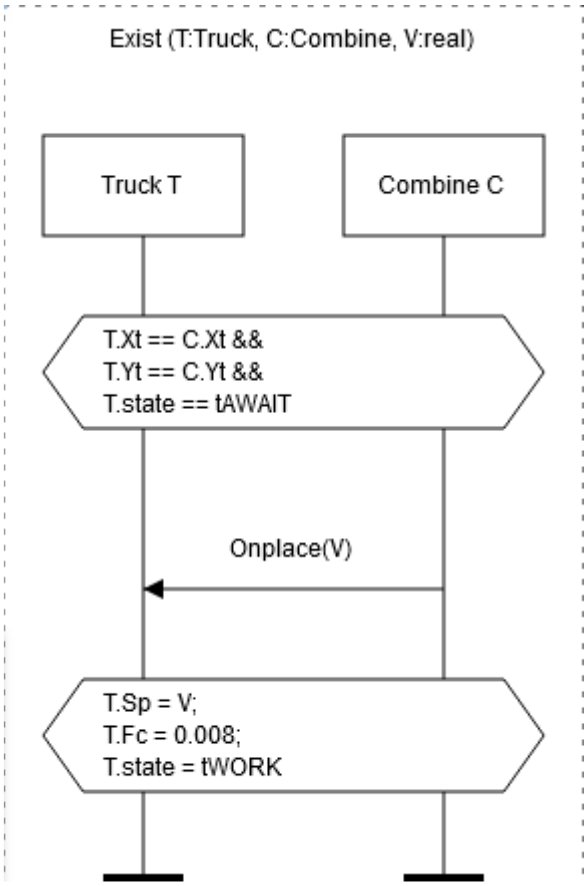
протокол дії TonwayX



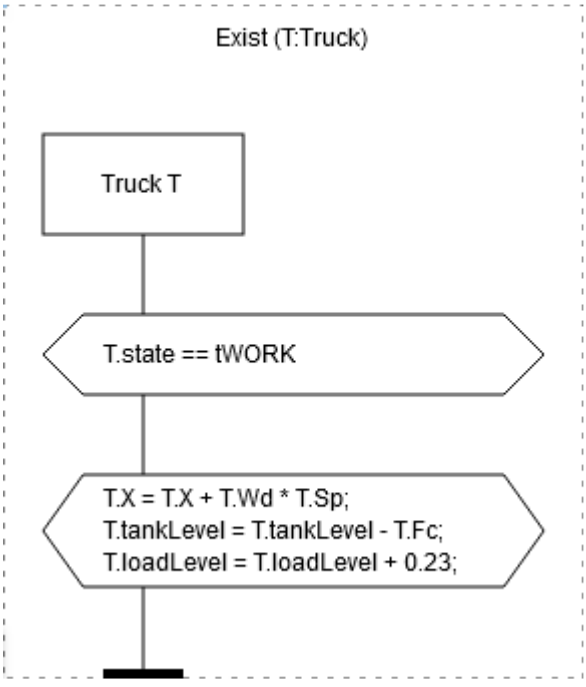
протокол дії TonwayY



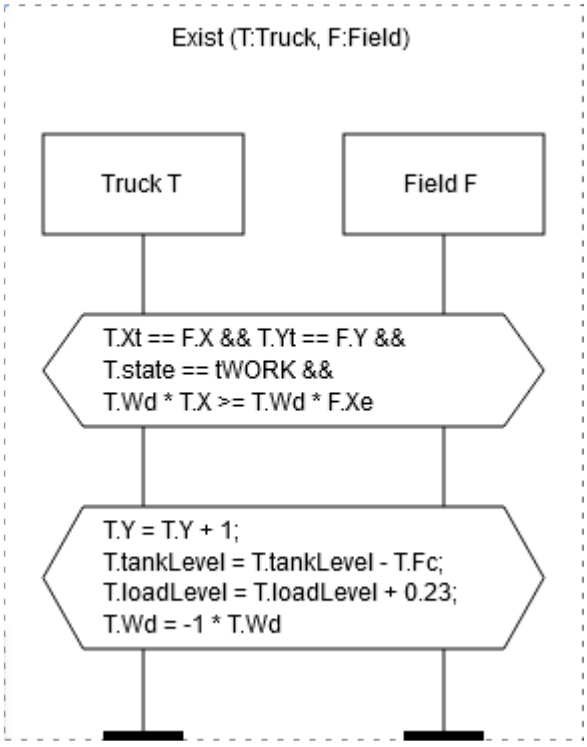
протокол дії Tonplace



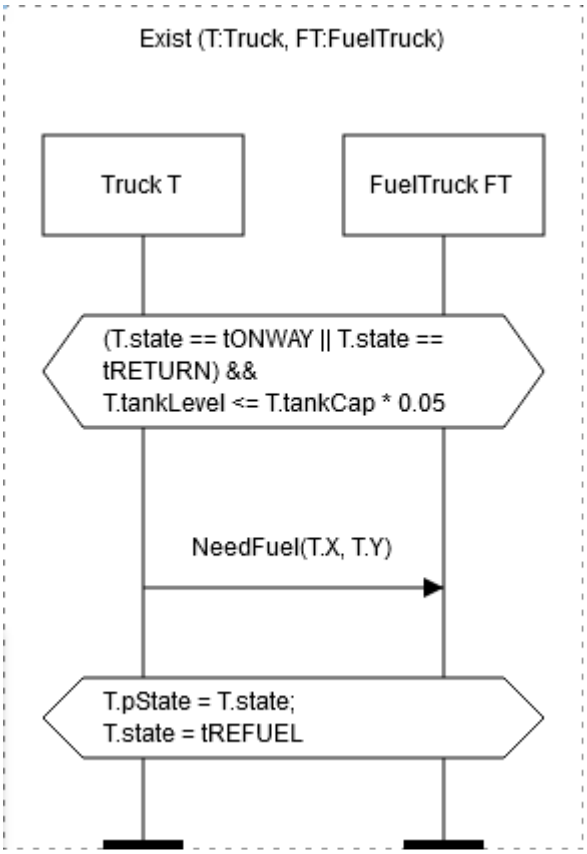
протокол дії TstartWork



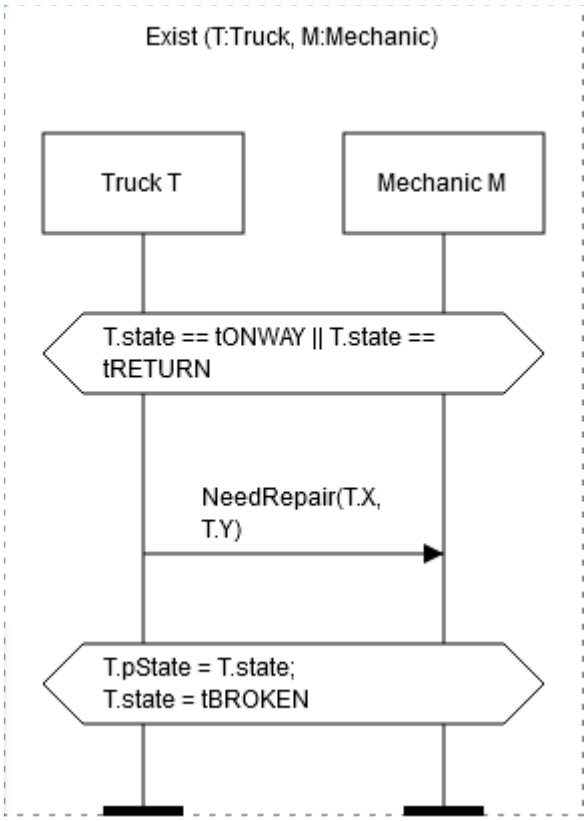
протокол дії Twork



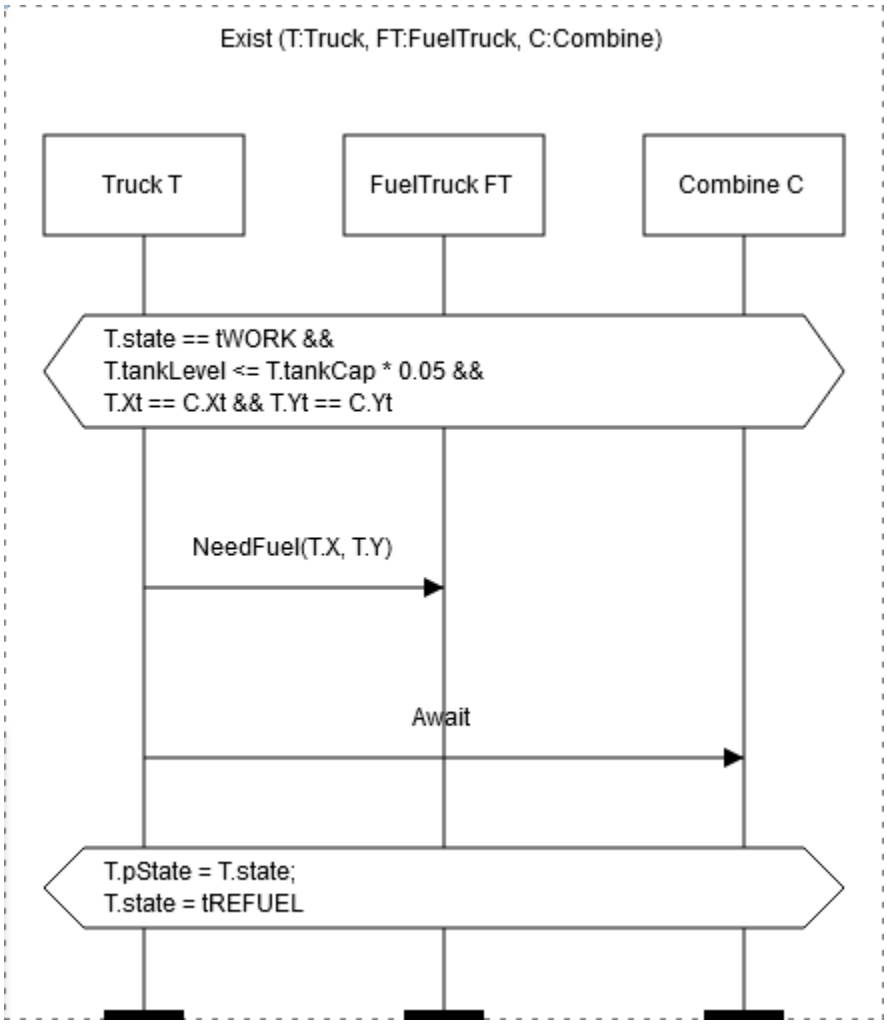
протокол дії TchangeWd



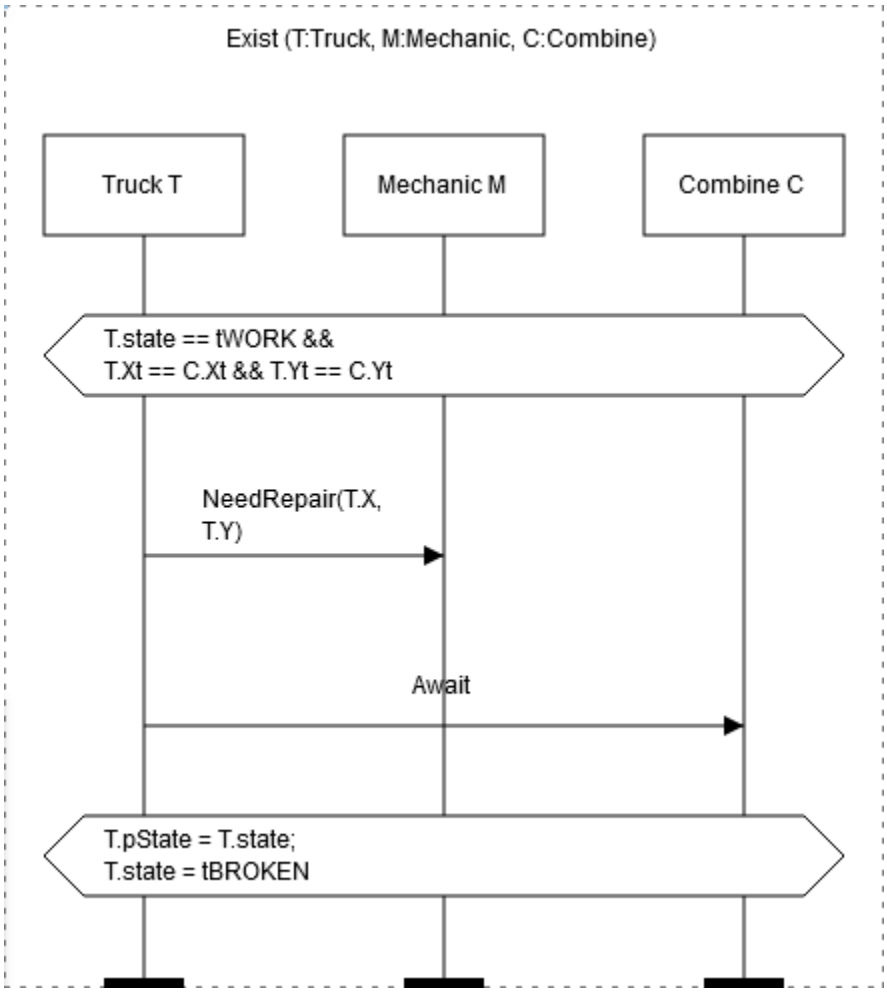
протокол дії TneedFuel



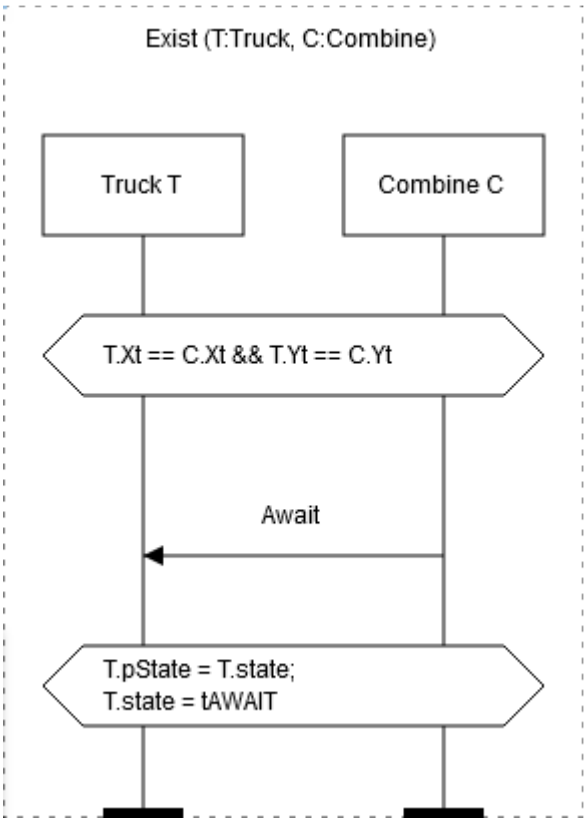
протокол дії TneedRepair



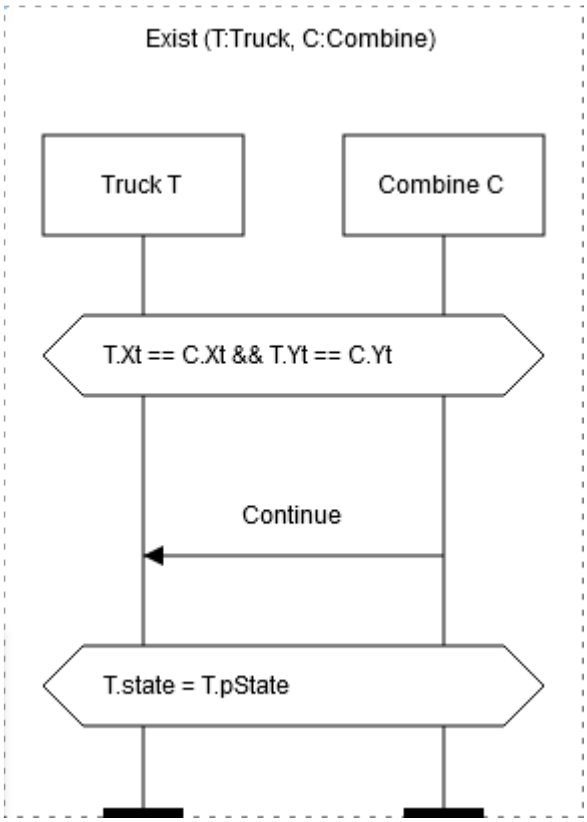
протокол дії TneedFuelOnField



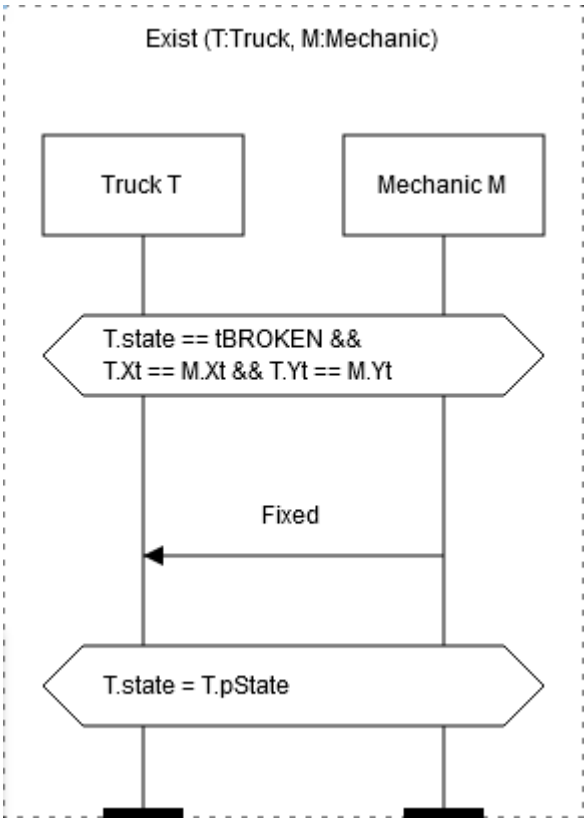
протокол дії TneedRepairOnField



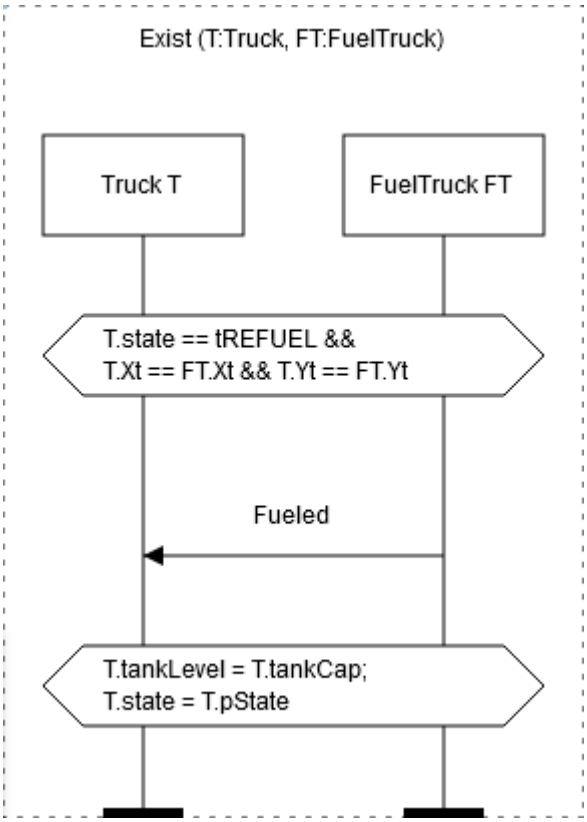
протокол дії TgetAwait



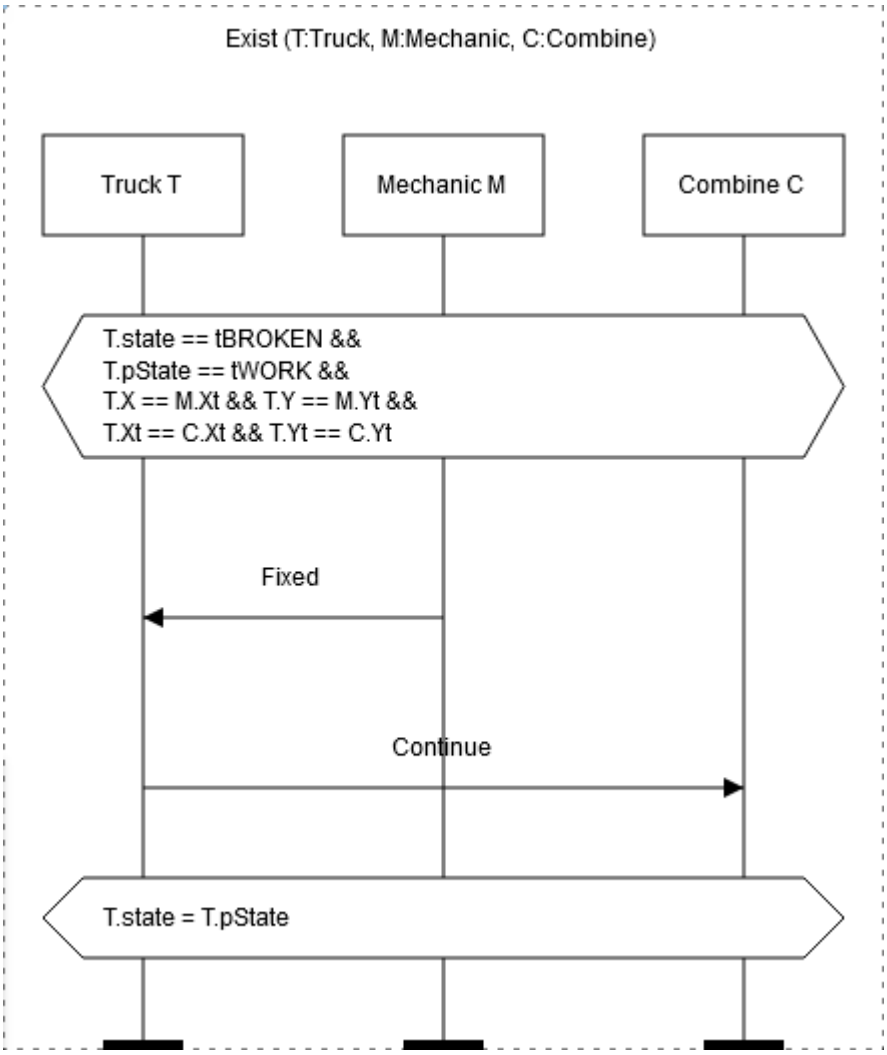
протокол дії TgetContinue



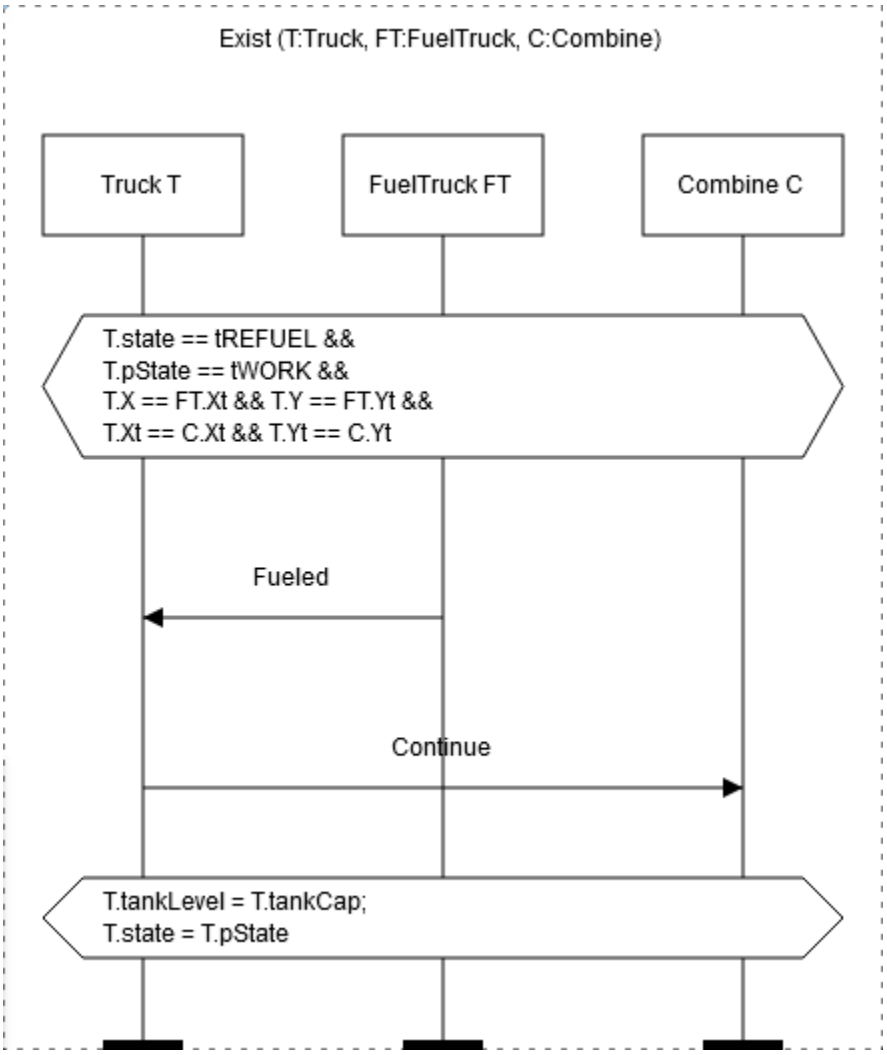
протокол дії TgetFixed



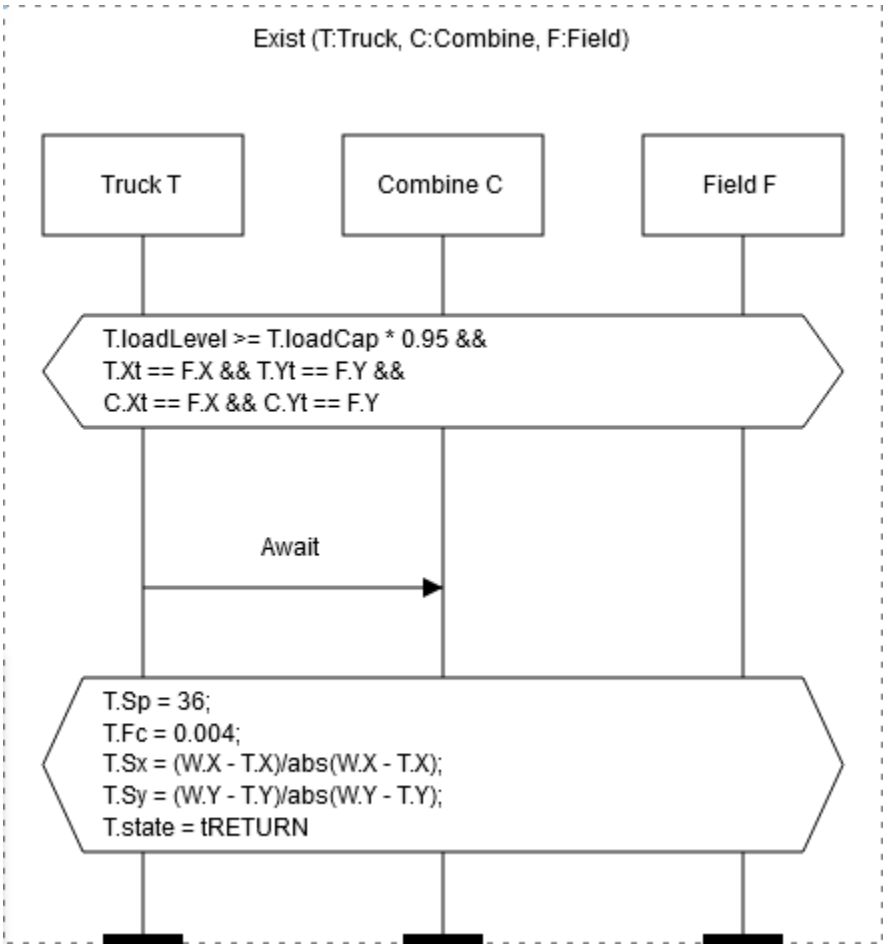
протокол дії TgetFueled



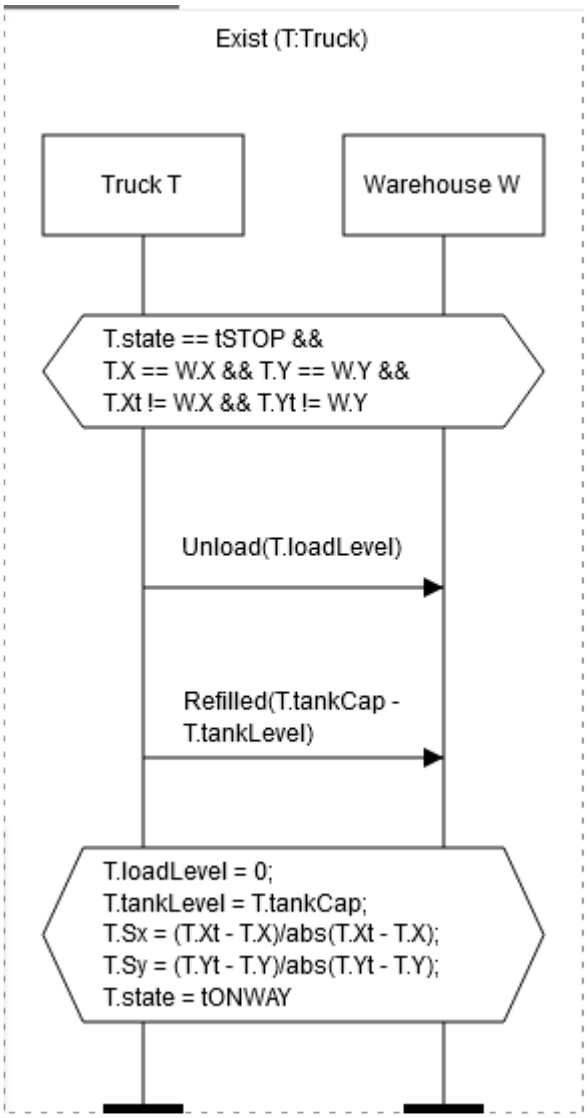
протокол дії TgetFixedOnField



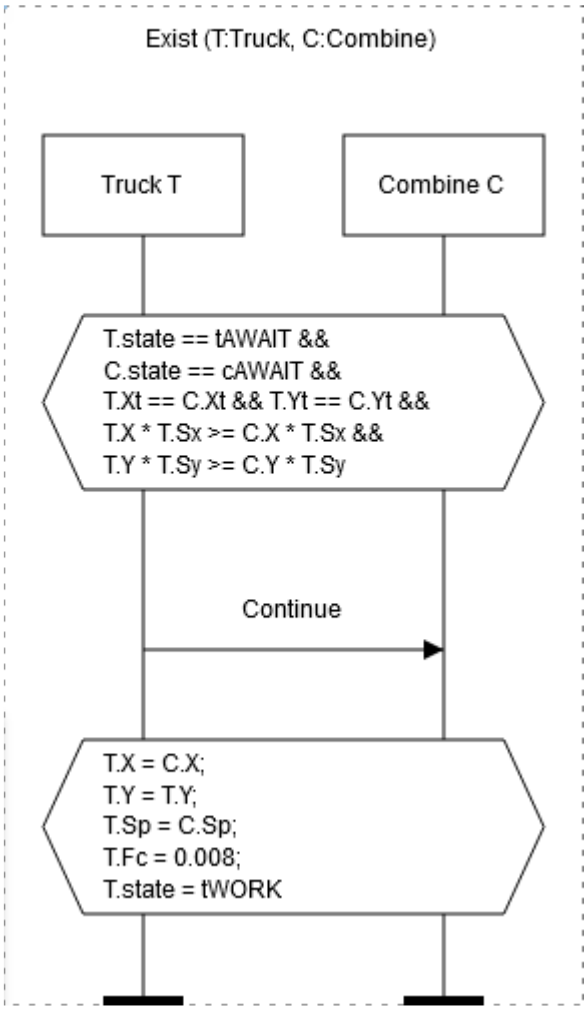
протокол дії TgetFueledOnField



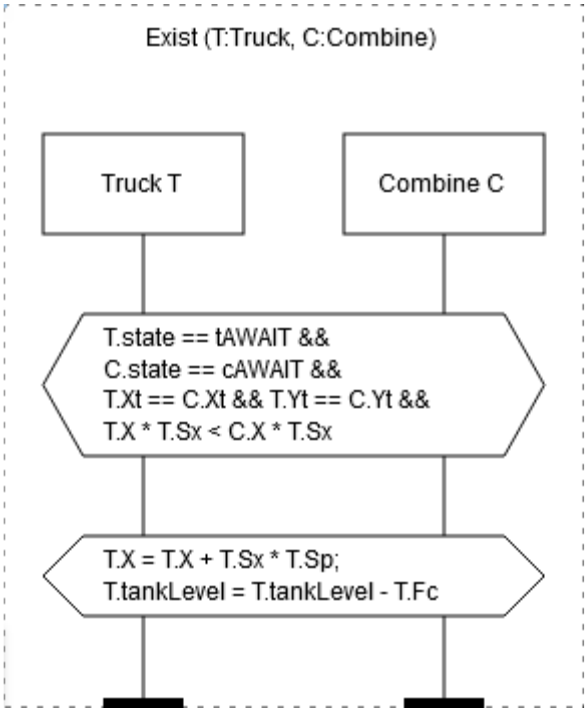
протокол дії TneedUnload



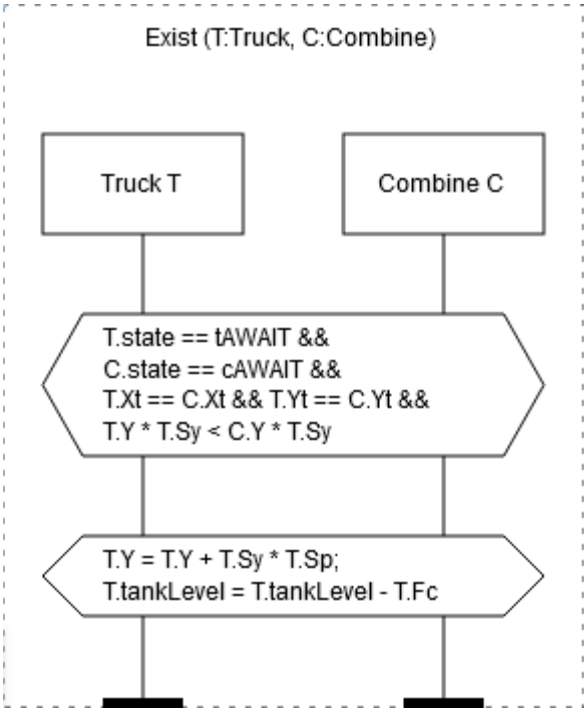
протокол дії Tunload



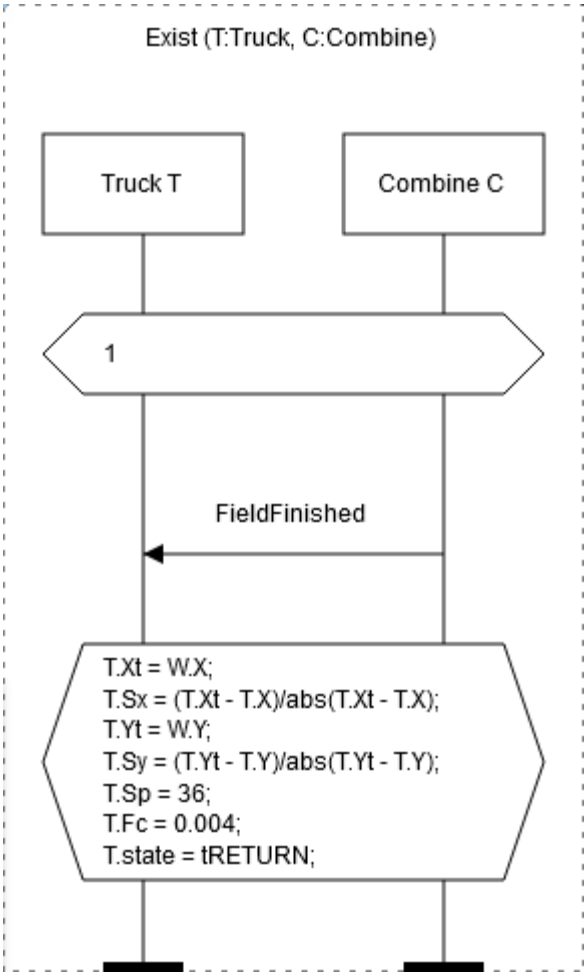
протокол дії TresumeWork



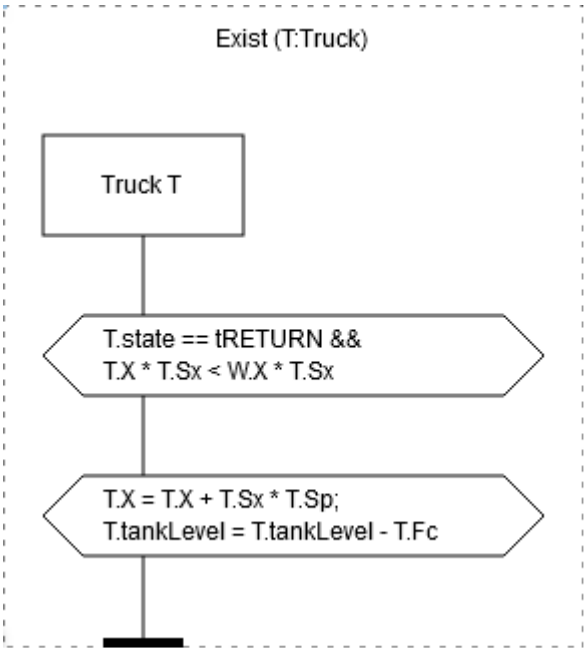
протокол дії TmoveToCombineX



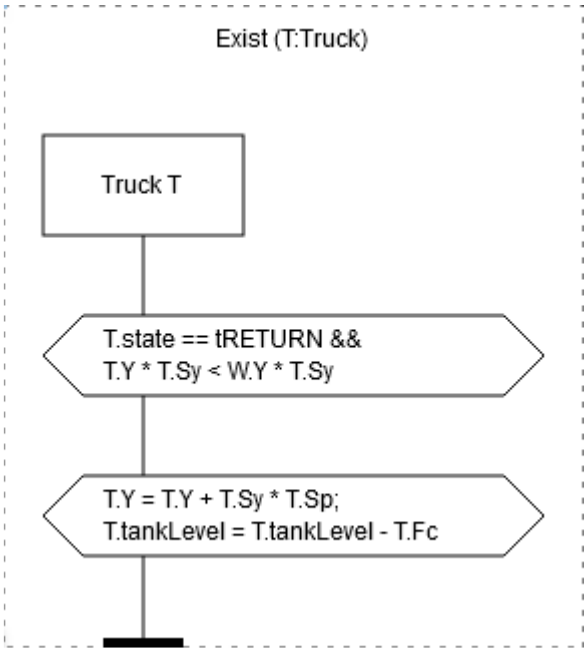
протокол дії TmoveToCombineY



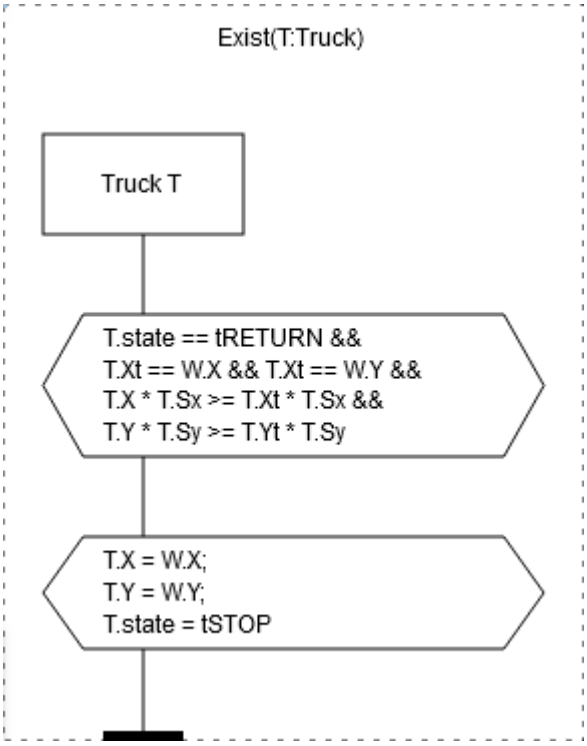
протокол дії TfinishField



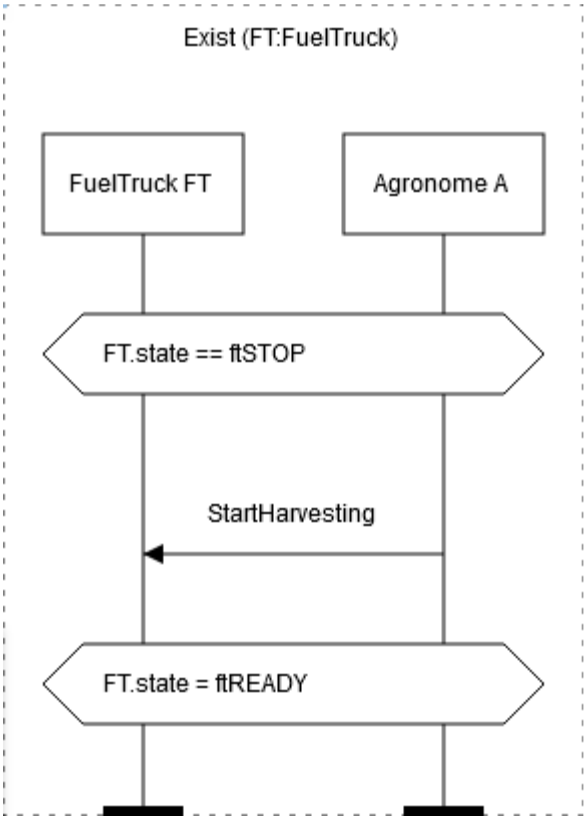
протокол дії TreturnX



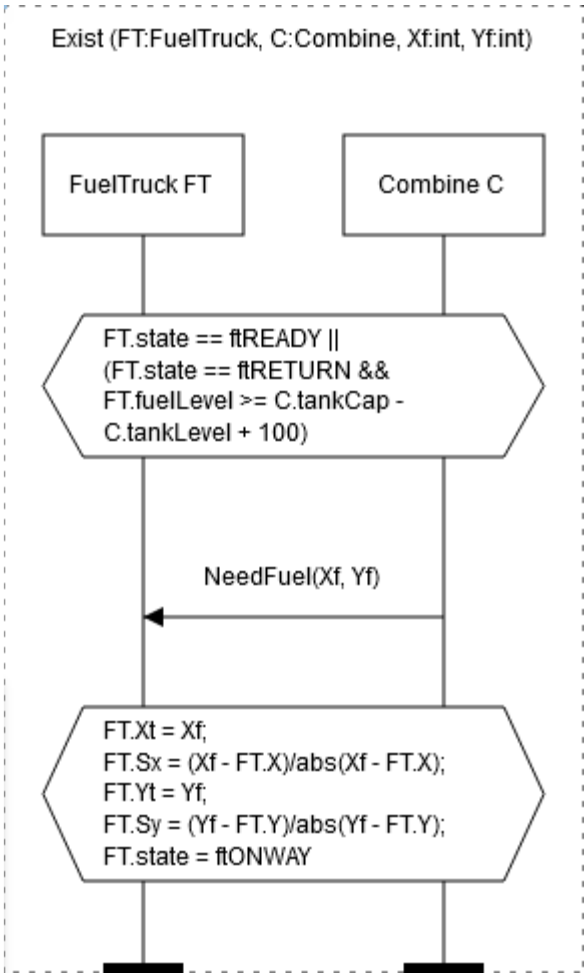
протокол дії TreturnY



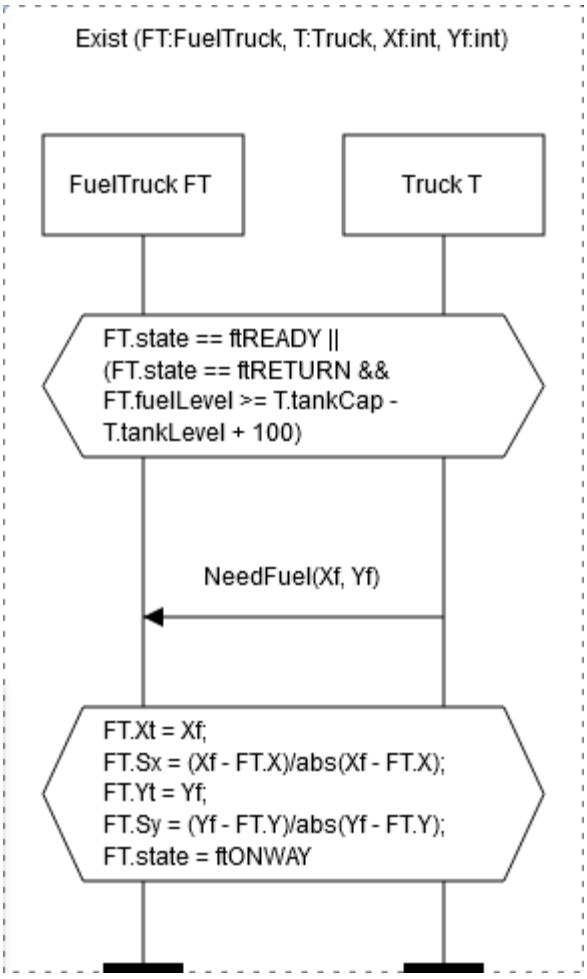
протокол дії Treturned



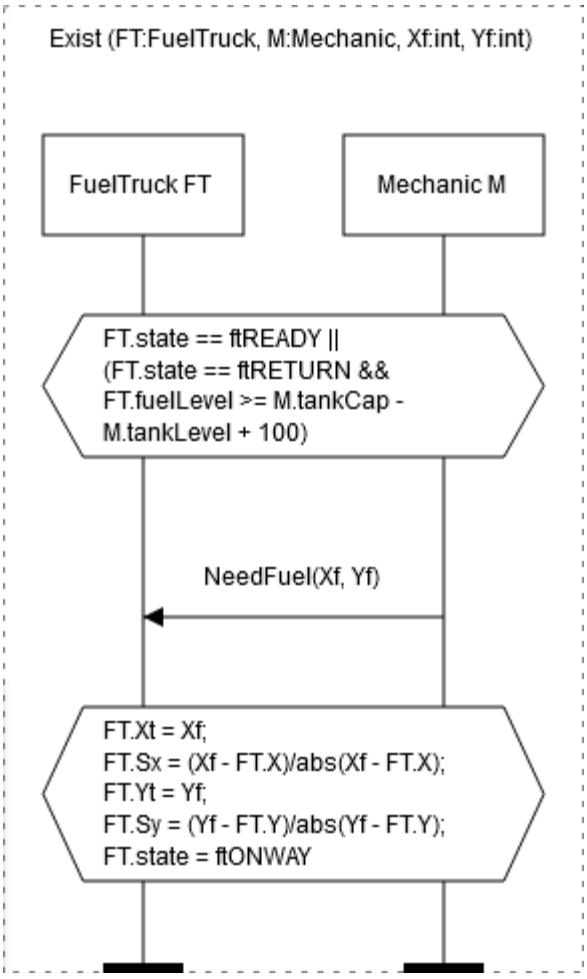
протокол дії FTgetReady



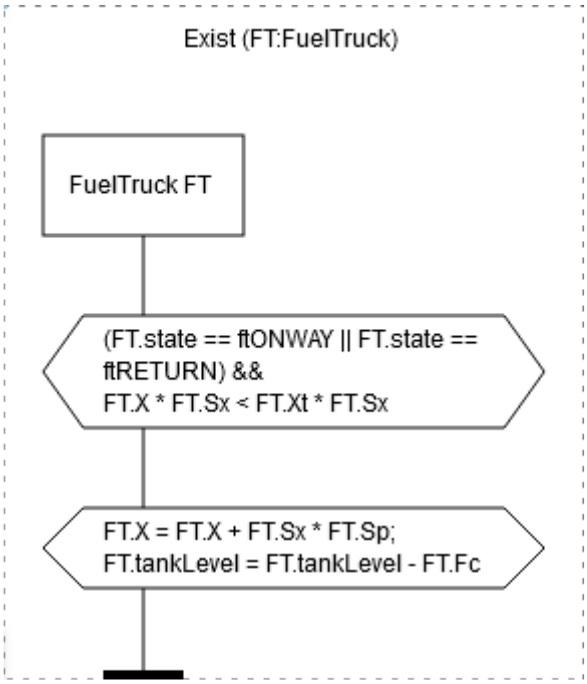
протокол дії FTgetCallC



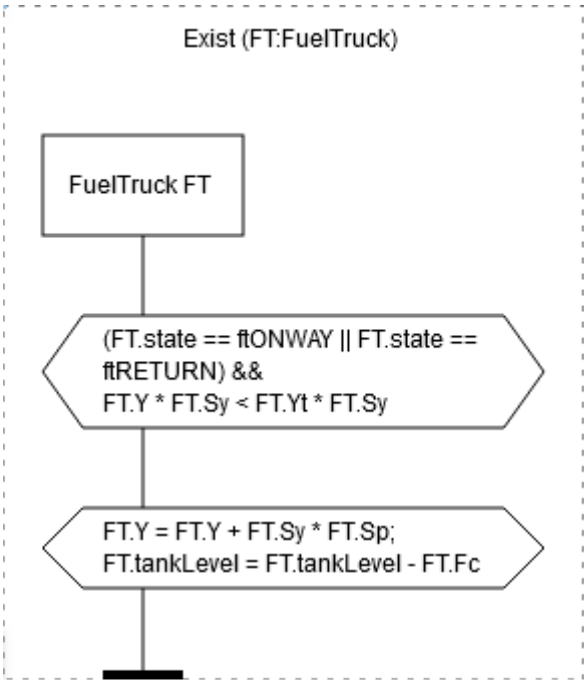
протокол дії FTgetCallT



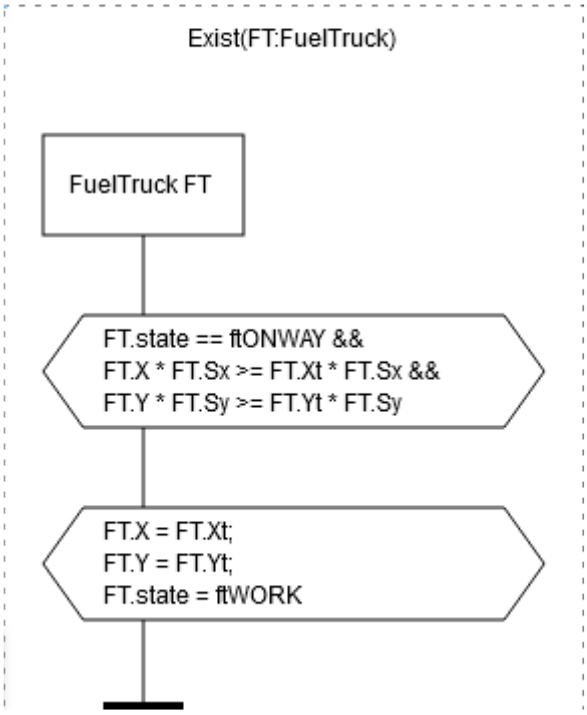
протокол дії FTgetCallM



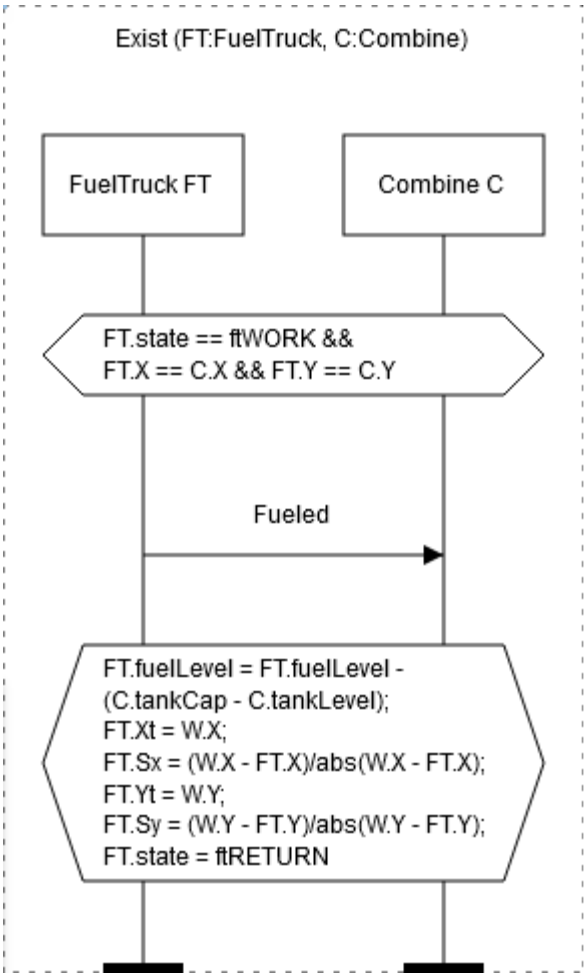
протокол дії FTonwayX



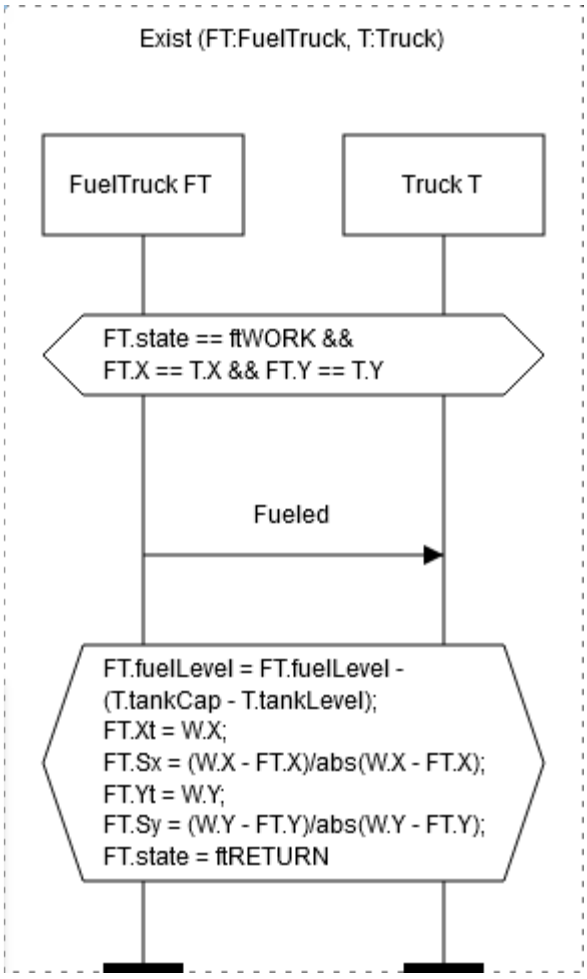
протокол дії FTonwayY



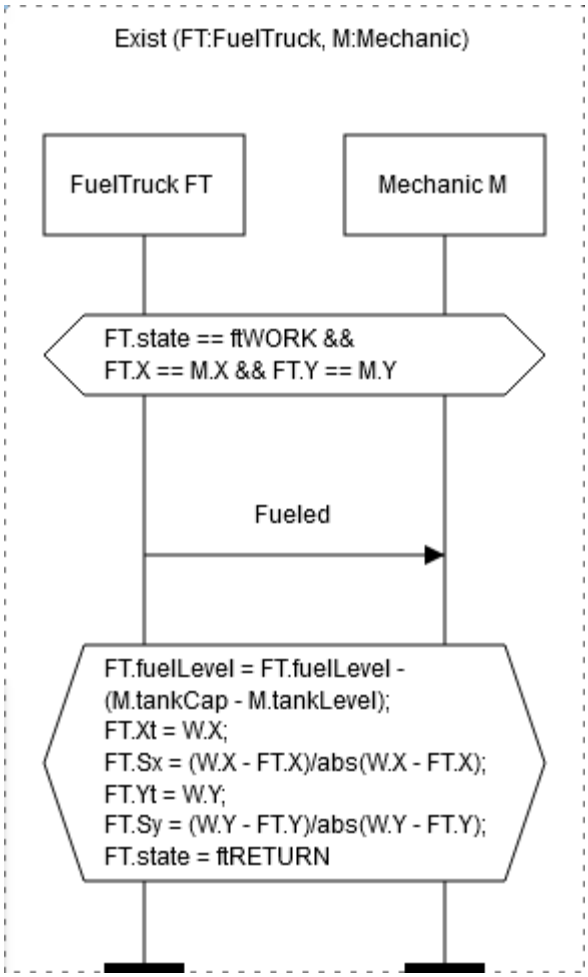
протокол дії FTonplace



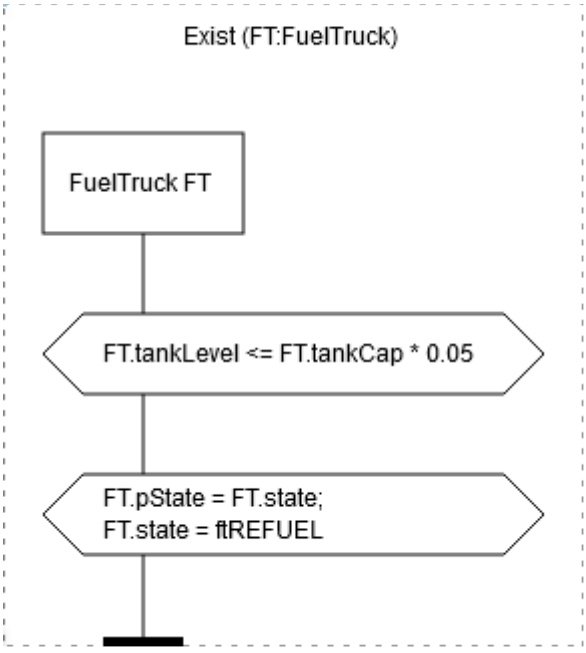
протокол дії FTfueledC



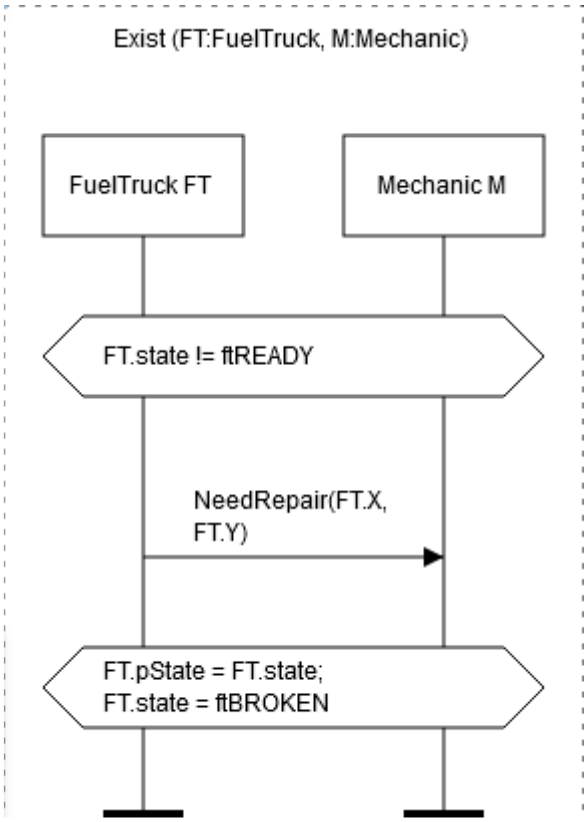
протокол дії FTfueledT



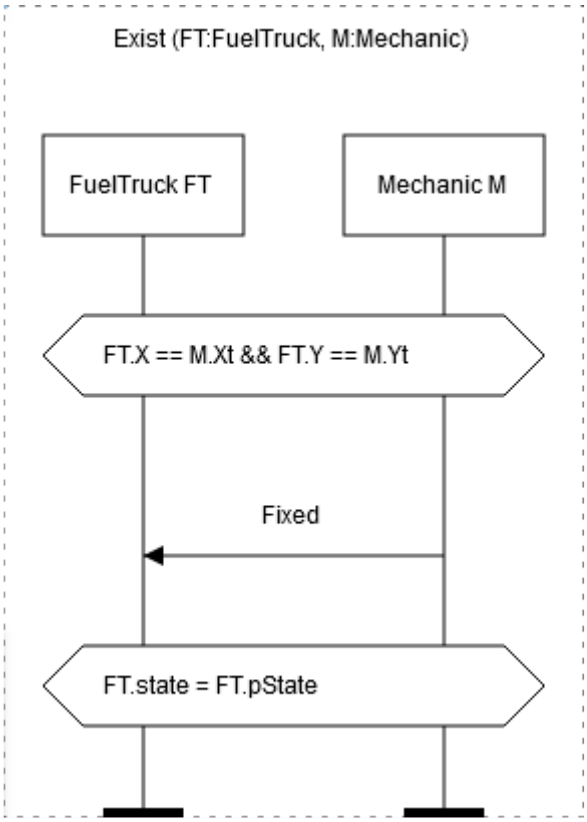
протокол дії FTfueledM



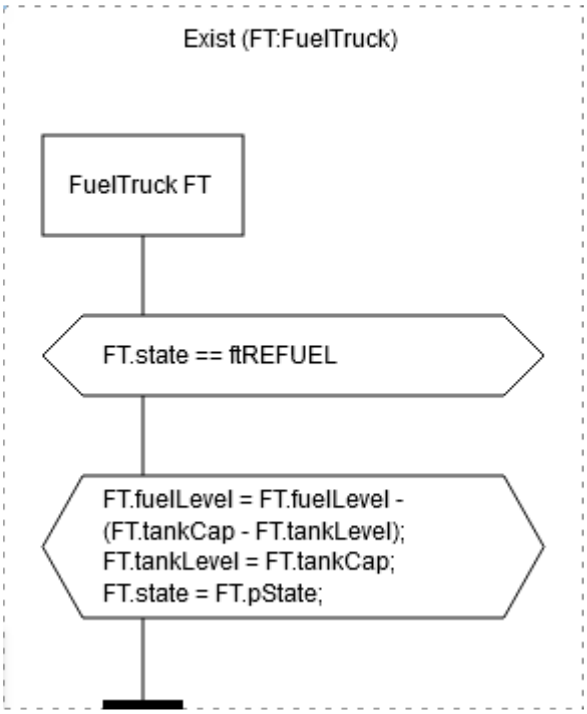
протокол дії FTneedFuel



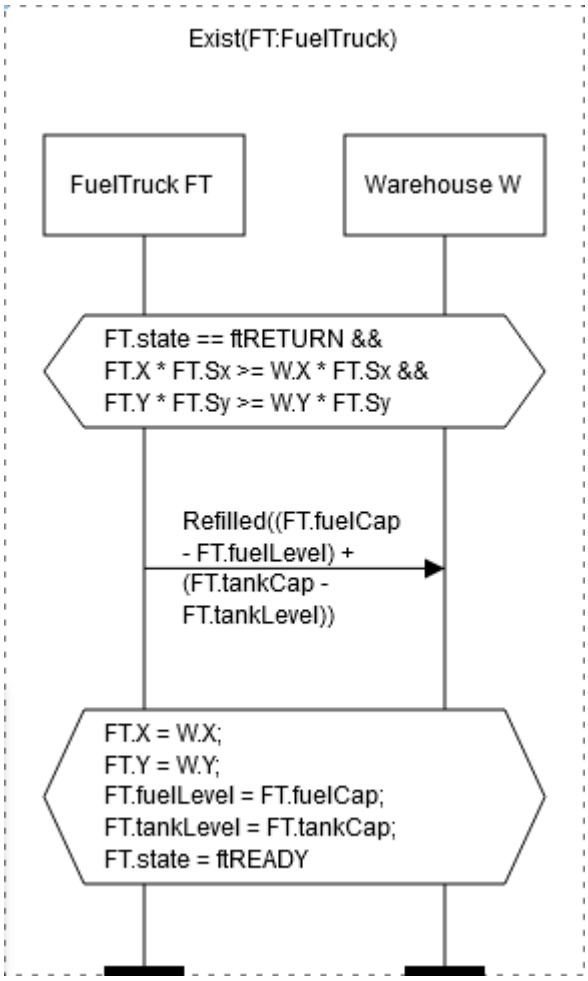
протокол дії FTneedRepair



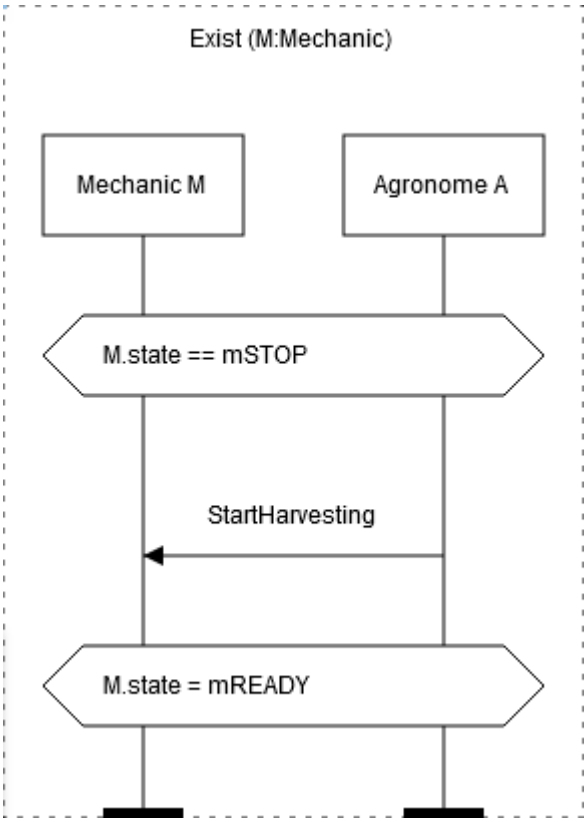
протокол дії FTgetFixed



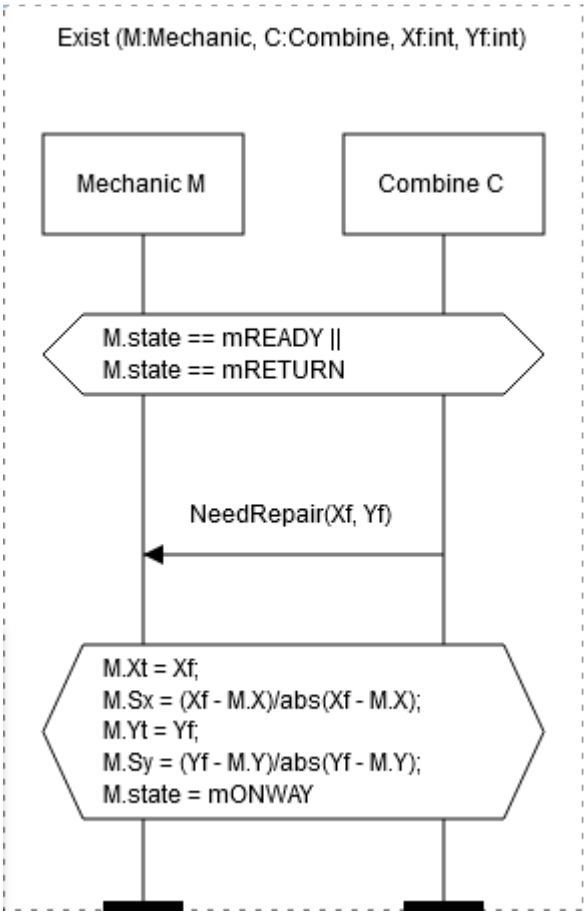
протокол дії FTgetFueled



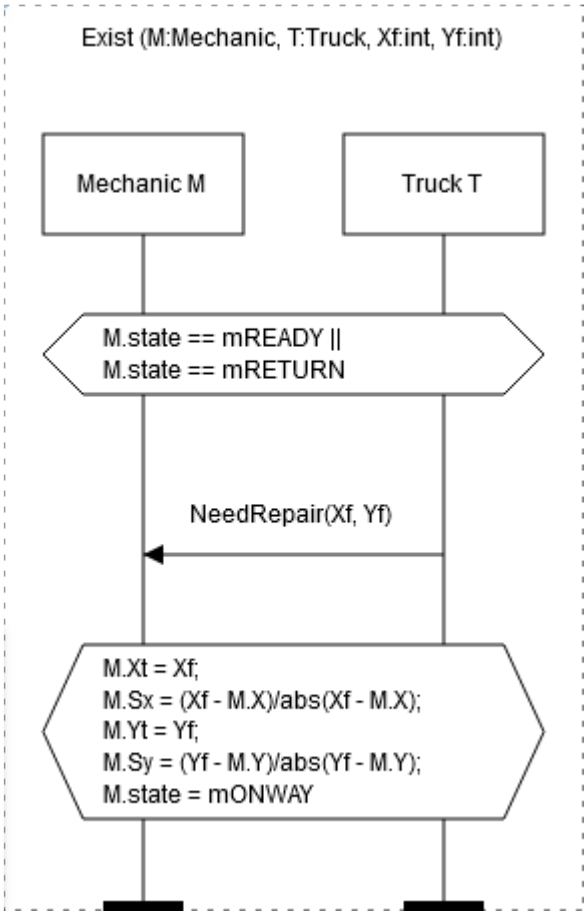
протокол дії FTreturned



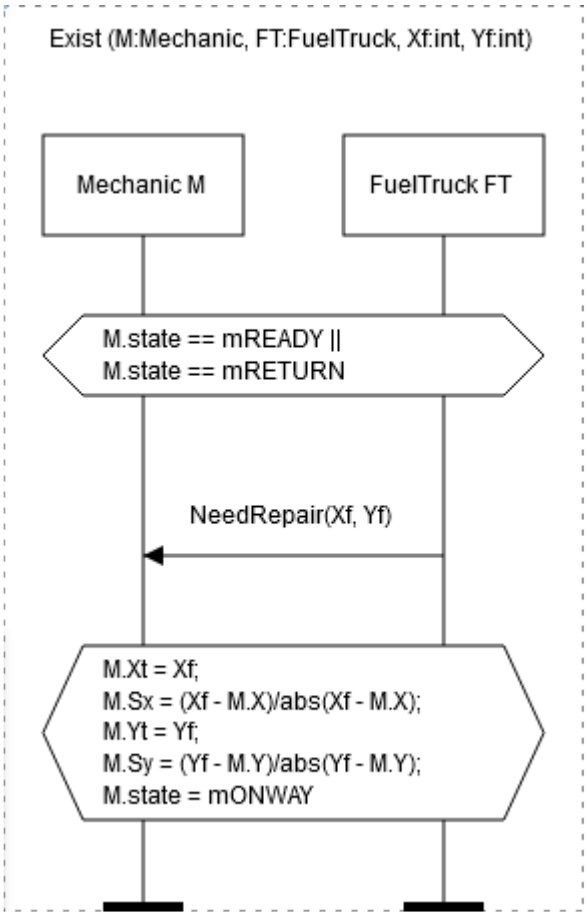
протокол дії MgetReady



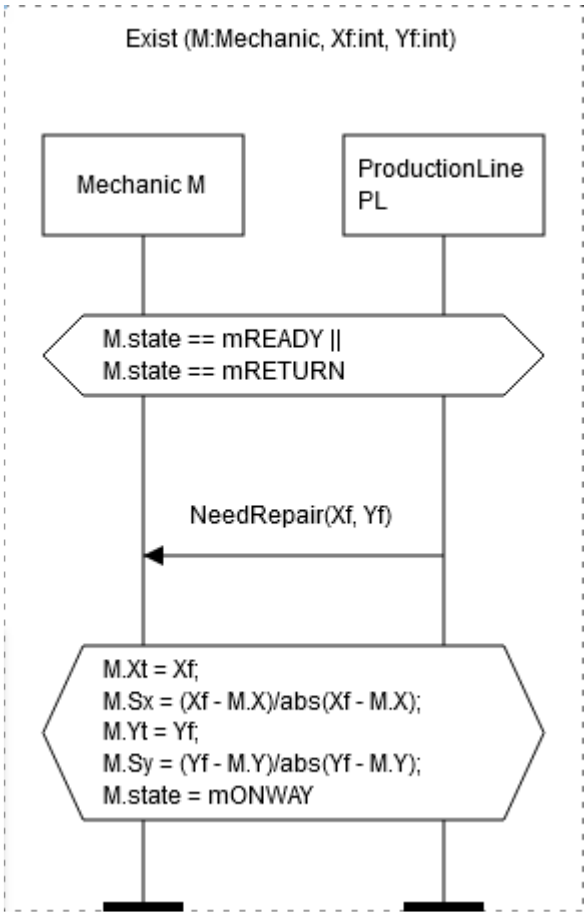
протокол дії MgetCallC



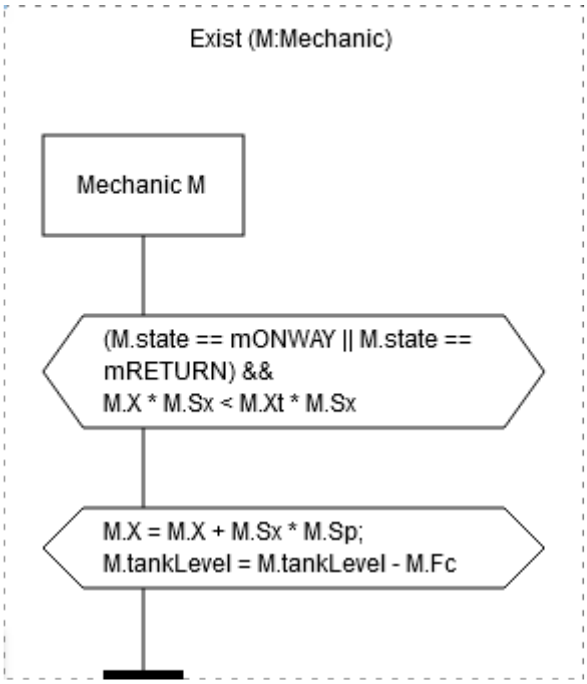
протокол дії MgetCallT



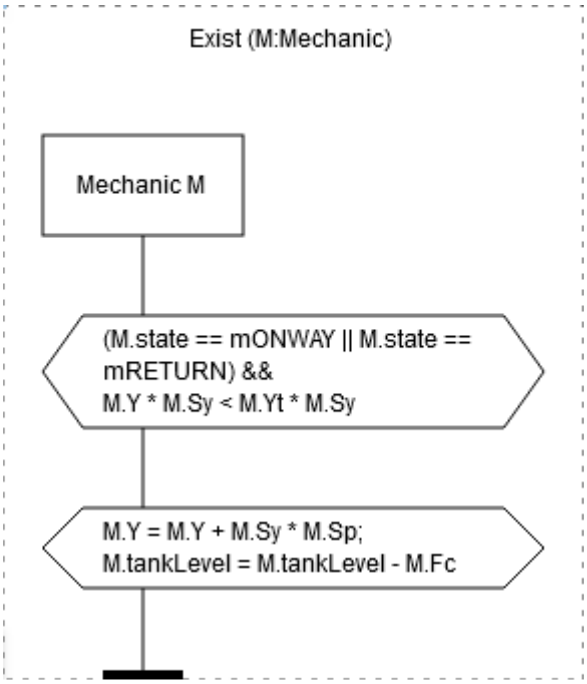
протокол дії MgetCallFT



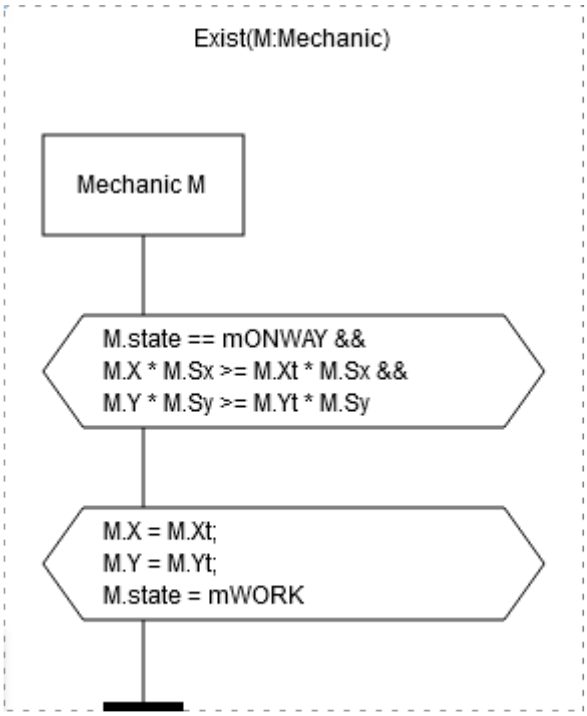
протокол дії MgetCallPL



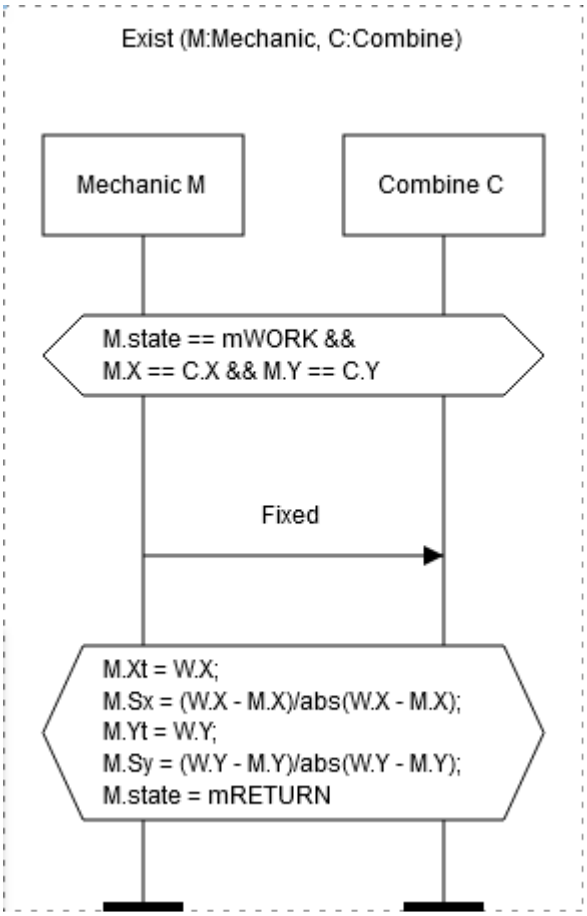
протокол дії MonwayX



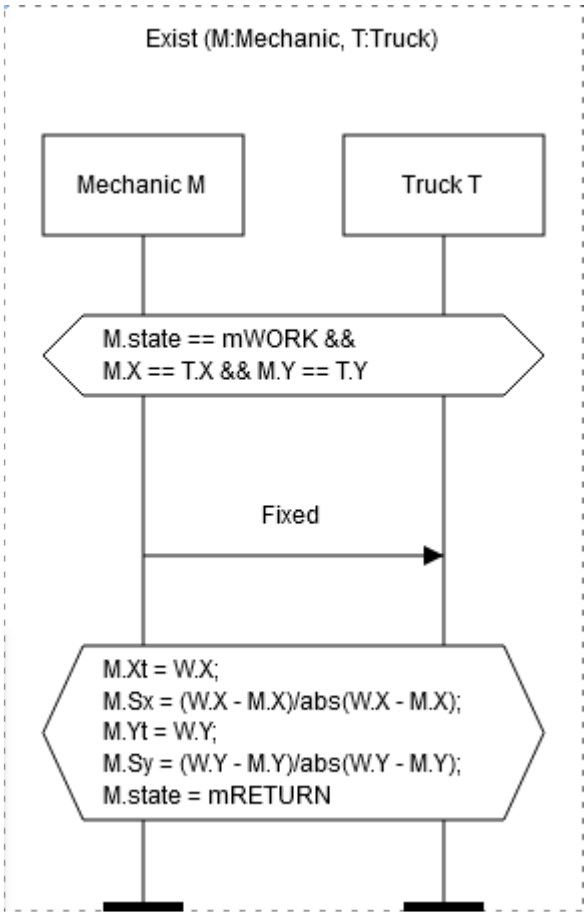
протокол дії MonwayY



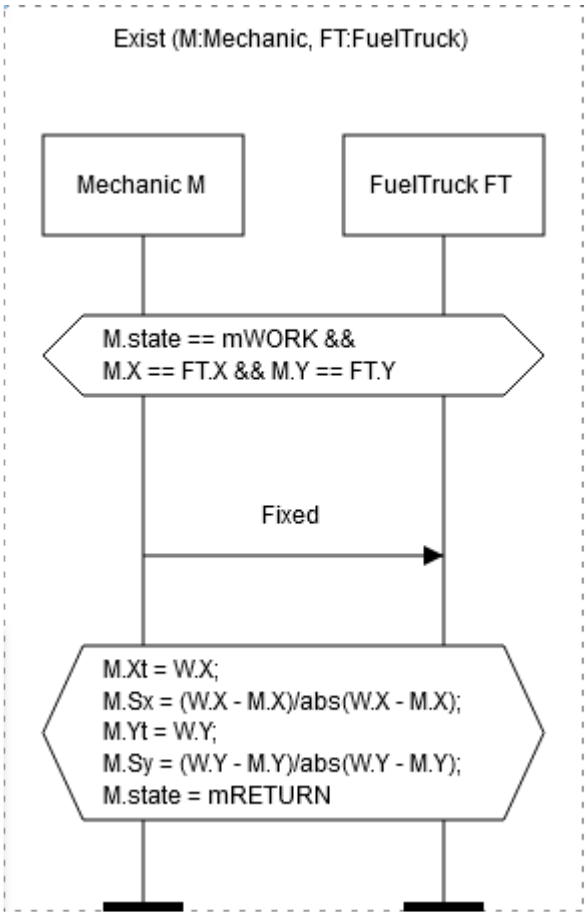
протокол дії Monplace



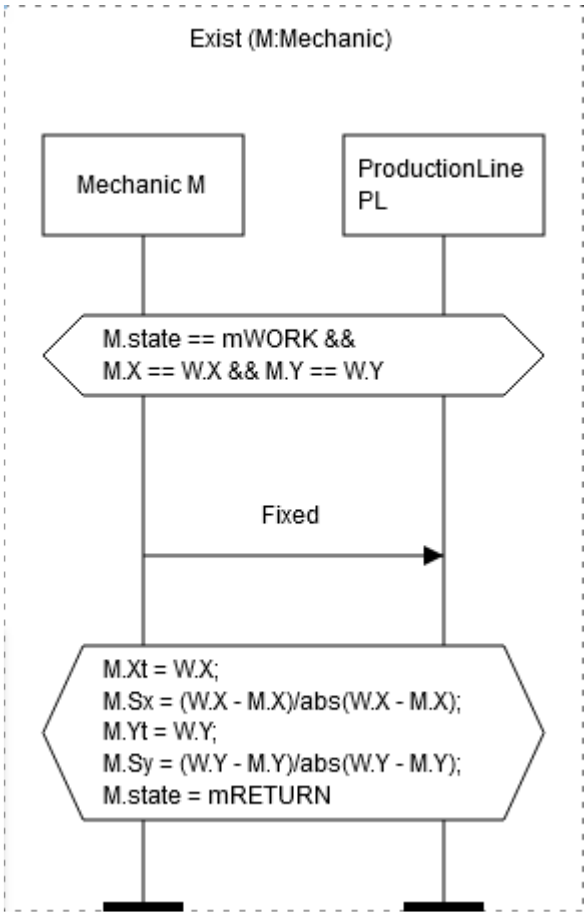
протокол дії MfixedC



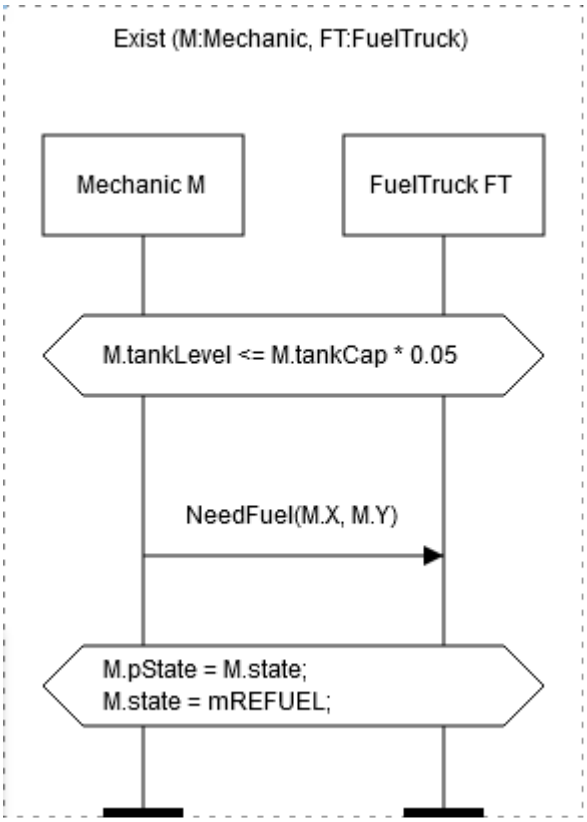
протокол дії MfixedT



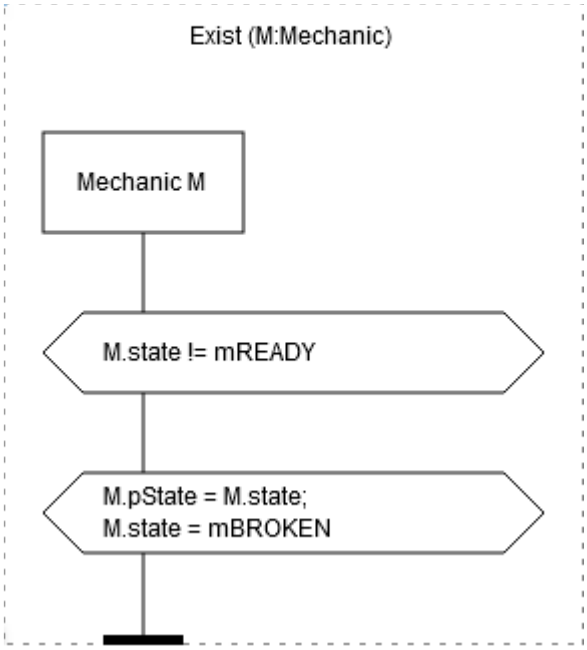
протокол дії MfixedFT



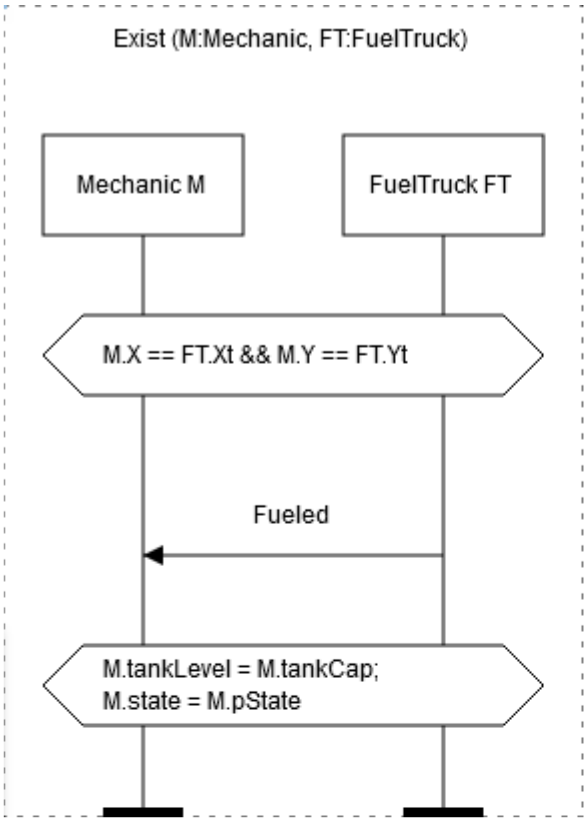
протокол дії MfixedPL



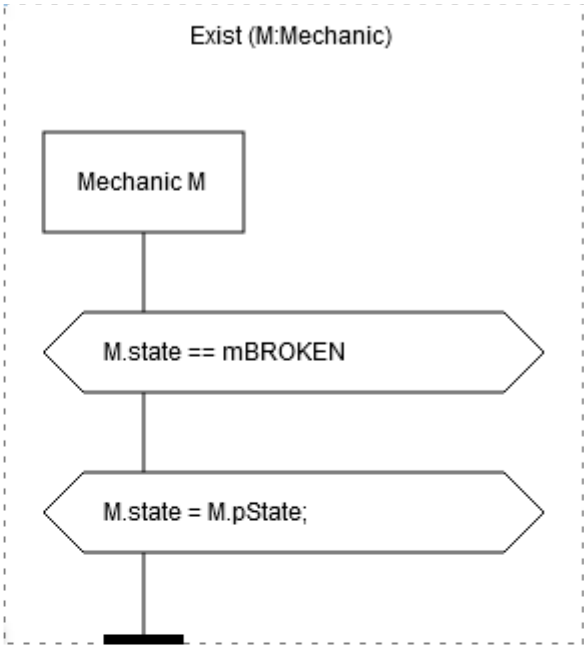
протокол дії MneedFuel



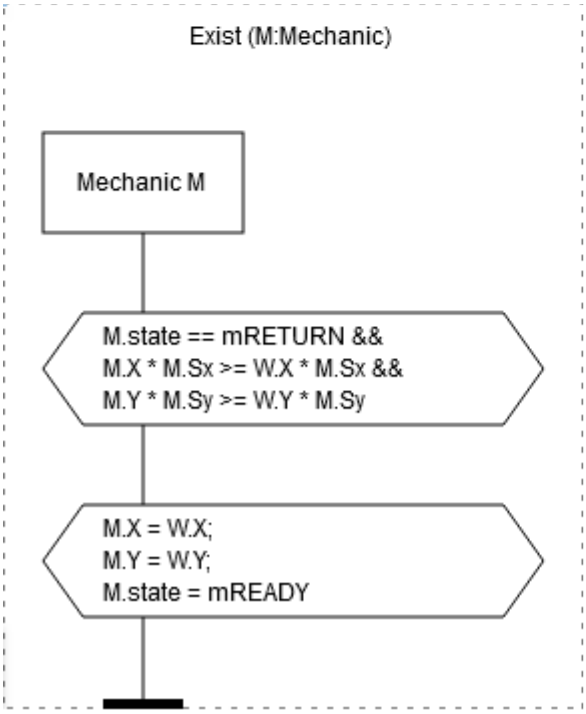
протокол дії MneedRepair



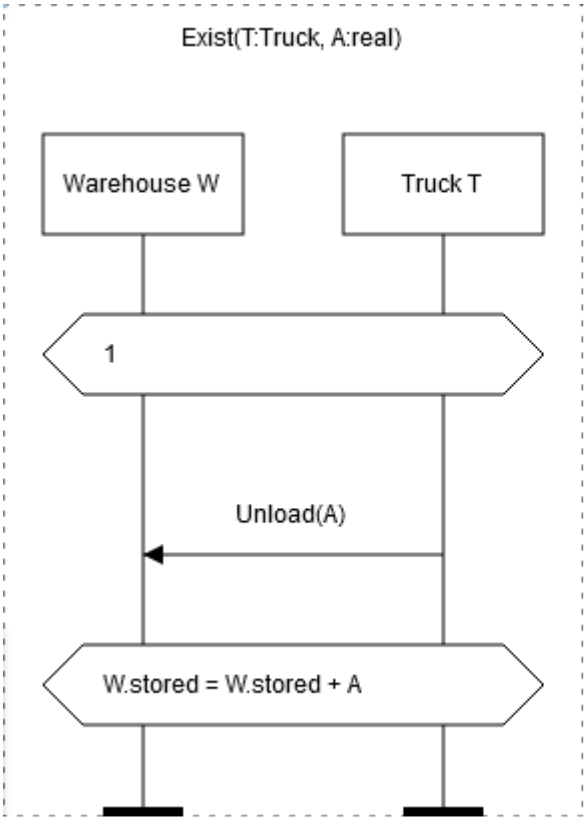
протокол дії MgetFueled



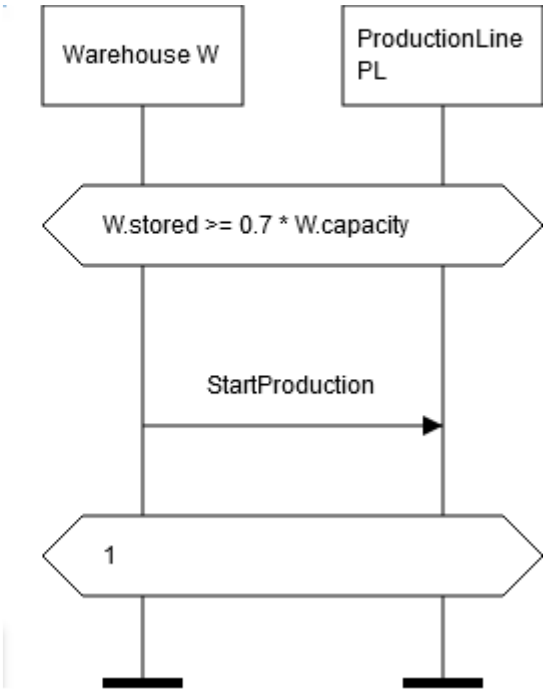
протокол дії MgetFixed



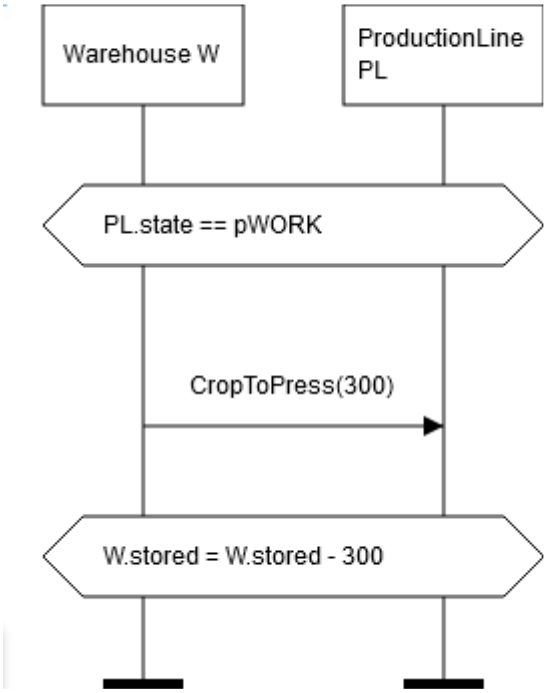
протокол дії Mreturned



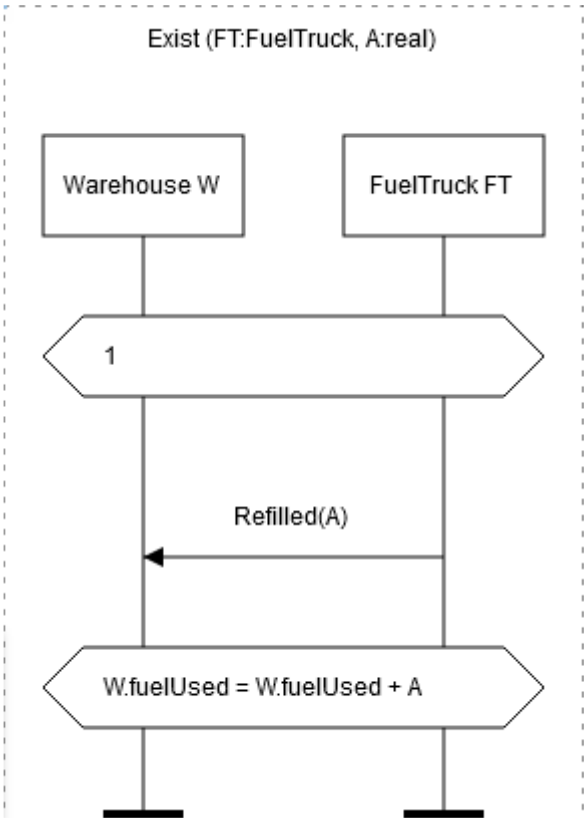
протокол дії WgetCrop



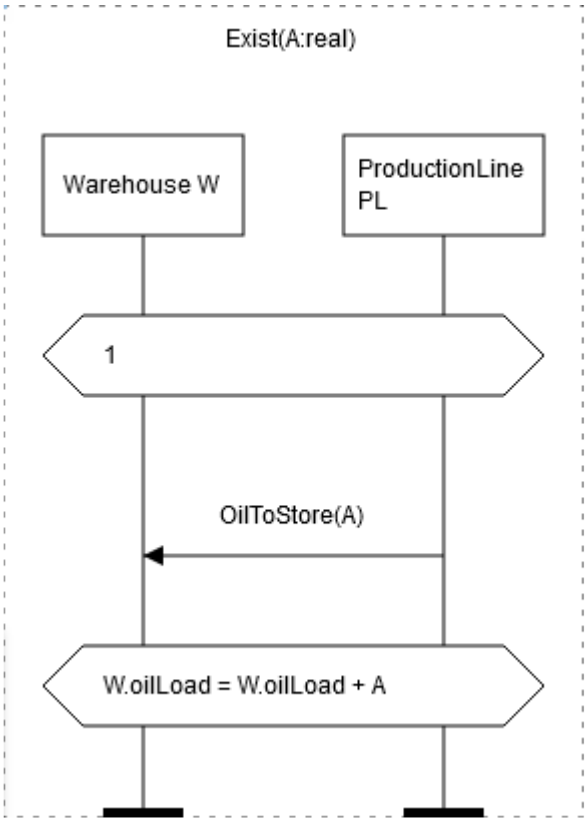
протокол дії WstartProd



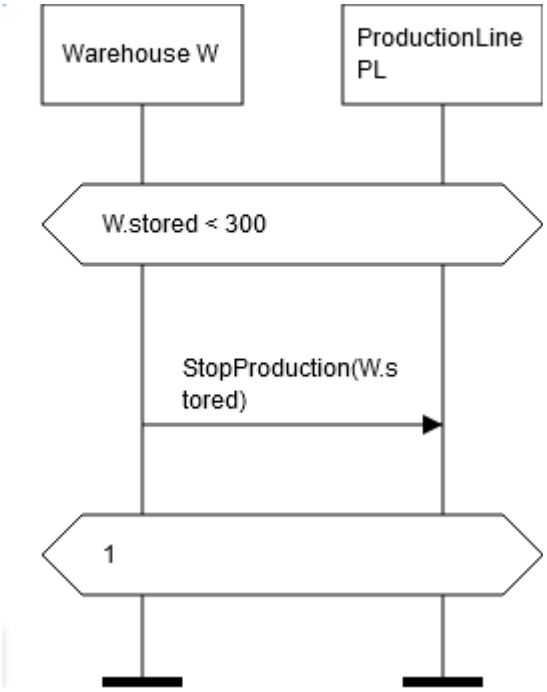
протокол дії WcropToProd



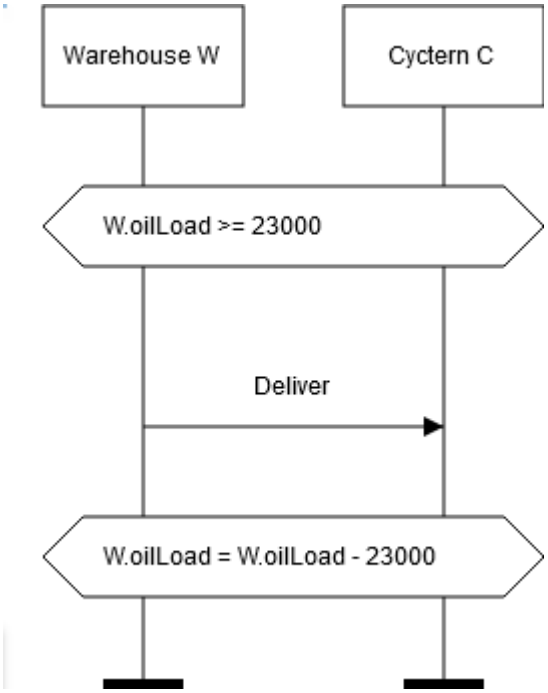
протокол дії WgiveFuel



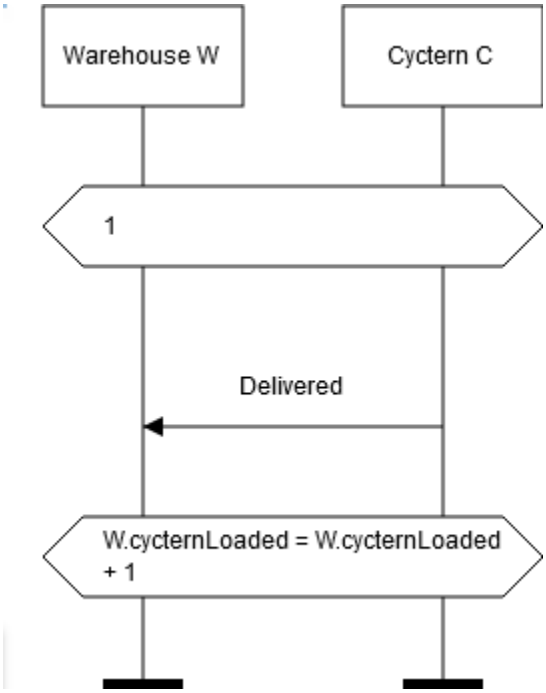
протокол дії WgetOil



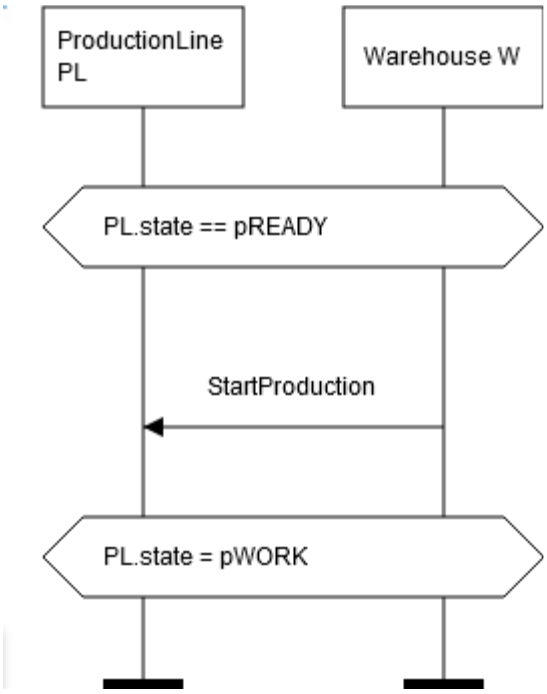
протокол дії WstopProd



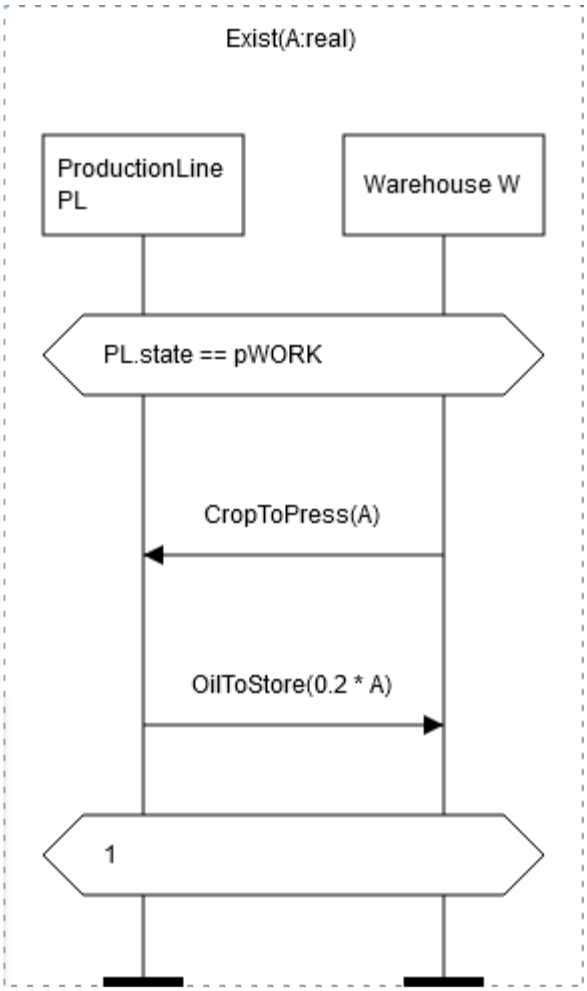
протокол дії WoilToDeliver



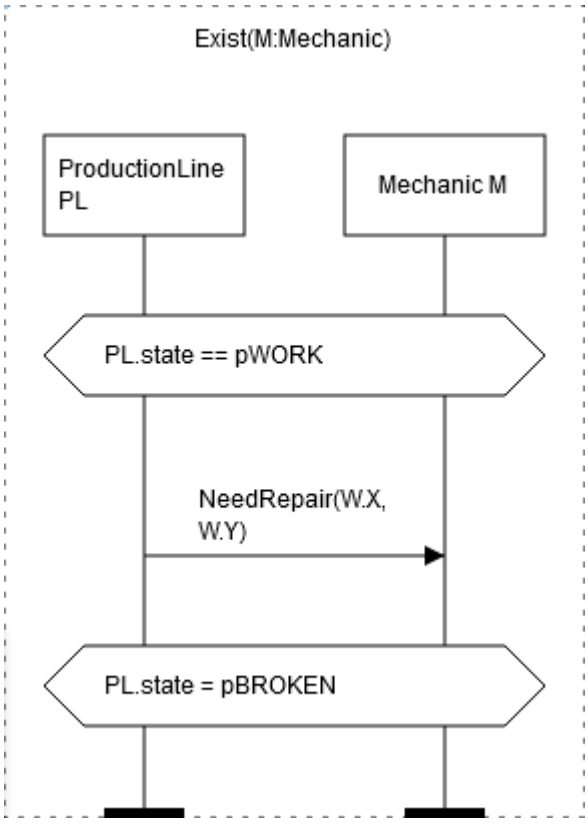
протокол дії WoilDelivered



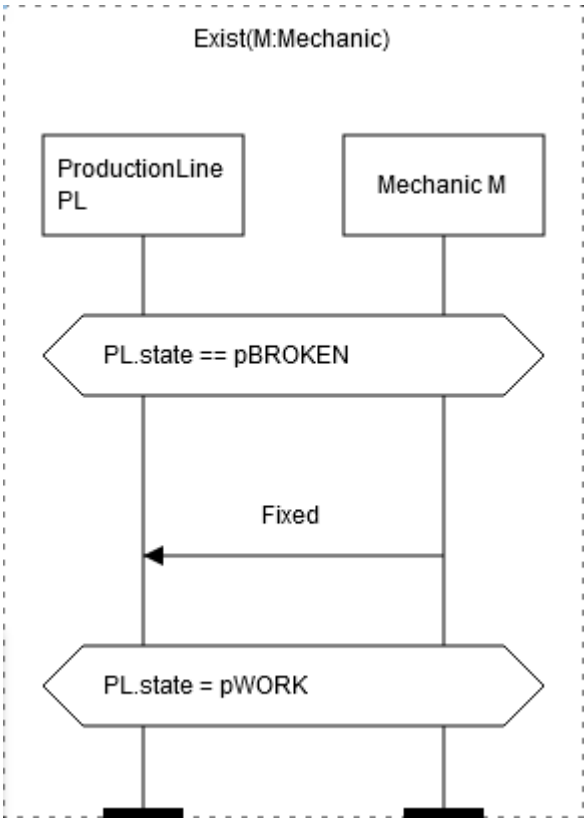
протокол дії PLstartWork



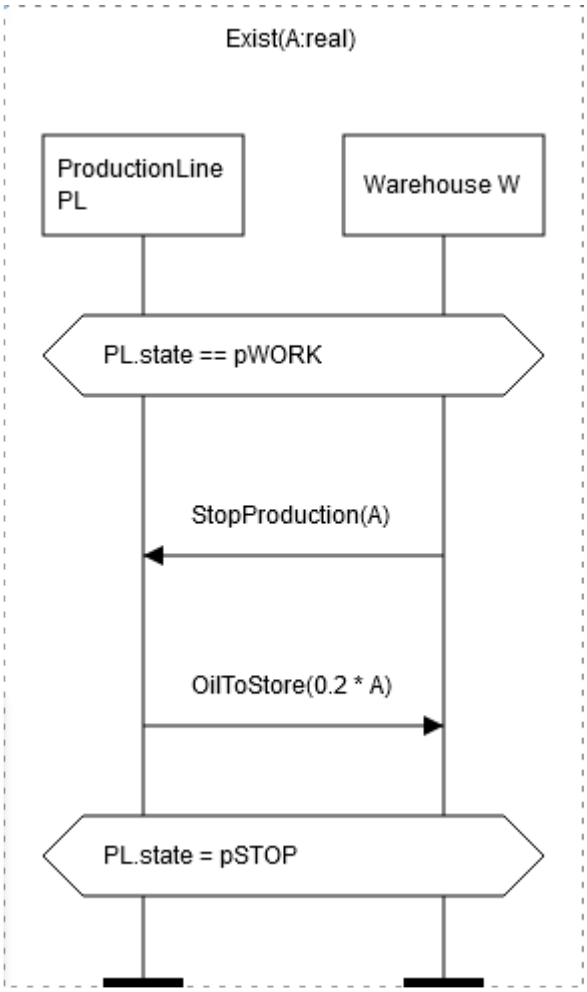
протокол дії PLgetCrop



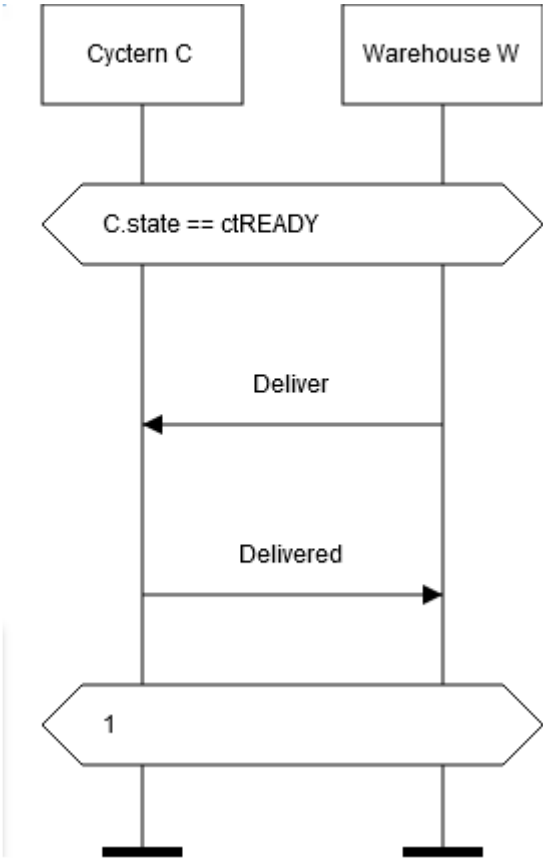
протокол дії PLneedRepair



протокол дії PLgetFixed



протокол дії PLstopWork



протокол дії CycToDeliver

Додаток Д

АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ

У ВИРОБНИЧОМУ ПРОЦЕСІ

ФГ «Надія»
ЄДРПОУ/ДРФО 31000629
19232, Черкаська обл., Жашківський р-н,
С. Марійка, вул.. Лесі Українки 22

Акт

про впровадження результатів дисертаційного дослідження

Наукові висновки, практичні напрацювання та результати дисертації Горбатюка Сергія Олександровича на тему: «ПРОБЛЕМИ КІБЕРБЕЗПЕКИ ТА МОДЕЛЬНИЙ МЕТОД РОЗРОБКИ РОЗПОДІЛЕНИХ СИСТЕМ В МІЖНАРОДНІЙ ЛОГІСТИЦІ» використані, реалізовані та впроваджені на технічній діючій базі фермерського господарства «Надія», що знаходиться в селі Марійка, Жашківський р-н, Черкаська обл.

Зокрема початкові дослідження по проектуванню та створенню інтелектуальної системи збору та аналізу інформації та централізованого керування технічними одиницями почалися в 2017 році, а в 2018 році перший діючий проект системи почав роботу. Всього під час роботи було залучено 14 одиниць техніки (трактори, комбайни, вантажівки та інше).

Директор: Шелестовський Г.Г.

LLC “SMART TRADING COMPANY” / ООО « СМАРТ ТРЕЙДИНГ КОМПАНИ »

Украина, 19200, Черкасская обл, г. Жашков, ул. Заводская 35

Ukraine, 19200, Cherkassy reg, Zhashkov, str. Zavodska 35

ЄДРПОУ: 42043882

Акт

про впровадження результатів дисертаційного дослідження

Наукові висновки, практичні напрацювання та результати дисертації Горбатюка Сергія Олександровича на тему: «ПРОБЛЕМИ КІБЕРБЕЗПЕКИ ТА МОДЕЛЬНИЙ МЕТОД РОЗРОБКИ РОЗПОДІЛЕНИХ СИСТЕМ В МІЖНАРОДНІЙ ЛОГІСТИЦІ» використані, реалізовані та впроваджені в діяльності ТОВ «Смарт Трейдинг Компані», що знаходиться в місті Жашків, Жашківський р-н, Черкаська обл.

ТОВ «Смарт Трейдинг Компані» займається експортом сільськогосподарської продукції, в тому числі продукцію, вироблену ФГ «Надія». Початкові дослідження по проектуванню та створенню інтелектуальної системи збору та аналізу інформації та електронного документообороту митних документів почалися в 2017 році, а в 2018 році перший діючий проект системи почав роботу.

Директор: Скиба І.І.

ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО «КСІБЕКС»

Код ЄДРПОУ – 35648356; ІПН – 356483526595; МФО – 354347

Адреса – 19202, Черкаська обл., м.Жашків, вул.Спортивна, 3

Тел. – (096) 702-19-86

Акт

про впровадження результатів дисертаційного дослідження

Наукові висновки, практичні напрацювання та результати дисертації Горбатюка Сергія Олександровича на тему: «ПРОБЛЕМИ КІБЕРБЕЗПЕКИ ТА МОДЕЛЬНИЙ МЕТОД РОЗРОБКИ РОЗПОДІЛЕНИХ СИСТЕМ В МІЖНАРОДНІЙ ЛОГІСТИЦІ» використані, реалізовані та впроваджені в діяльності ПрАТ «КСІБЕКС», що знаходиться в місті Жашків, Жашківський р-н, Черкаська обл.

ПрАТ «КСІБЕКС» займається міжнародними автомобільними перевезеннями, в тому числі продукцію експорту ТОВ «Смарт Трейдинг Компані» виробництва ФГ «Надія». Початкові дослідження по проектуванню та створенню інтелектуальної системи збору та аналізу інформації та електронного документообороту митних документів почалися в 2017 році, а в 2018 році перший діючий проект системи почав роботу.

Компанія володії власним автопарком з 6 машин, які здійснюють міжнародні перевезення та залучає найманих супідрядників, ці машини виступали агентами в створеній системі керування міжнародними автоперевезеннями.

Генеральний директор – Шелестовський Віталій Григорович

Шелестовський В. Г.