Міністерство освіти і науки україни
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра авіоніки

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____ Павлова С.В.
“_____”_____2021 р.

ДИПЛОМНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
“МАГІСТР”

Тема: Технічні засоби діагностування та контролю бортових систем інформаційного обміну на літаку

Виконавець: Горбаченко Святослав Русланович

Керівник: Слободян Олександр Петрович
Консультанти з окремих розділів пояснювальної записки:

Охорона праці – Козлітін Олексій Олександрович

Охорона навколишнього середовища – доц. Дмитруха Тетяна Іллівна

Нормоконтролер: Левківський Василь Васильович

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет аеронавігації електроніки та телекомунікацій
Кафедра авіоніки
Напрям (спеціальність) 173 «Авіоніка»
(шифр, найменування)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Павлова С.В.
«_____»_____2021р.

ЗАВДАННЯ
на виконання дипломної роботи
_____Горбаченка Святослава Руслановича_____
(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи Технічні засоби діагностування та контролю бортових систем інформаційного обміну на літаку
затверджена наказом ректора від «_____»_____20__ р. №___

2. Термін виконання роботи : з _____ по _____

3. Вихідні дані до роботи :
За основу розробки формального аналізу властивостей безпеки системи була обрана мова AADL, стандартна мова моделювання Суспільства автомобільної техніки (SAE), для проектування систем на основі моделей (MBSE), яка забезпечує більш суворий опис системи та семантику часу виконання і добре підходить для моделювання вбудованих систем у реальному часі, яка описана в розділі 3.

4. Зміст пояснювальної записки:
Розділ 1. | Огляд ключових технологій Розподіленої інтегрованої модульної системи авіоніки
Розділ 2 «Новий Метод аналізу на основі моделі з безліччю обмежень для Процесу Динамічної реконфігурації Інтегрованої модульної авіоніки»
Розділ 3 «Аналіз безпеки на основі AADL з використанням формальних методів, застосовуваних до цифрових систем повітряних суден»
Розділ 4. Охорона праці. Тема «Комбінований вплив шуму і змішаних розчинників на функцію слуху у працівників авіаційної промисловості»

Розділ 5. Тема «Сталий розвиток зеленої авіаційної промисловості на шляху до комплексної системи підтримки»

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Аналіз літератури	01.10.2020-17.09.2021	
2	Написання розділу I	18.09.2021-30.09.2021	
3	Написання розділу II	30.09.2021-15.10.2021	
4	Написання розділу III	15.10.2021-30.10.2021	
	Написання розділу IV	30.10.2021-15.11.2021	
	Написання розділу V	15.11.2021-31.11.2021	
5	Оформлення пояснювальної записки та графічного матеріалу.	1.12.2021-10.12.2021	

8. Дата видачі завдання: “_____"_____ 202__ р.

Керівник дипломної роботи (проекту) _____ Слободян Олександр Петрович
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Горбаченко Святослав Русланович
(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи Магістра «Технічні засоби діагностування та контролю бортових систем інформаційного обміну»

131 сторінок ,39 рисунків, 5 таблиць, 211 використаних джерел.

Об’єкт дослідження – Номінальна модель визначеннями несправностей в архітектурі мови AADL.

Мета дипломної роботи – підтримка загальної моделі, що відображає поточний стан проектування системи протягом всього життєвого циклу розробки, що дозволяє всім учасникам процесу 4754 A "Рекомендовані аерокосмічні практики" (ARP) мати можливість обмінюватися інформацією та аналізувати проект системи.

Метод дослідження – розширення мови AADL з підтримкою інструментів для формального аналізу властивостей безпеки системи при наявності несправностей.

Установлено, що для розробки бортових засобів контролю оцінки технічного стану авіаційного обладнання необхідно використовувати інтелектуальні засоби контролю циклічного опитування датчиків, що дозволять вирішити задачі оцінки технічного стану та діагностування працездатності та функціонування обладнання в реальному часі.

Матеріали дипломної роботи рекомендуються використовувати для подальшого проведення наукових досліджень, навчальному процесі та в практичній діяльності фахівців авіаційної галузі.

Прогнозовані припущення щодо розвитку об’єкта дослідження – розробка та удосконалення апаратно-програмних засобів бортових систем інформаційного обміну і контролю з урахуванням вимог нормативних документів та стандартів, що вказують на вимоги до сучасного обладнання літаків.

Зміст

Вступ

Технічний прогрес в авіаційній та будь-якій іншій галузі тісно пов'язаний з автоматизацією технологічних процесів. Сьогодні Автоматизація технологічних процесів використовується для підвищення характеристик надійності, довговічності, екологічності, ресурсозбереження і, найголовніше, економічності і простоти експлуатації. Завдяки швидкому розвитку комп'ютерних технологій і мікропроцесорів у нас є можливість використовувати більш досконалі і складні методи моніторингу та управління системами авіаційної промисловості і будь-якими іншими. Мікропроцесорні та електронні обчислювальні пристрої, з'єднані обчислювальними і керуючими мережами з використанням загальних баз даних, мають стандарти, що дозволяють модифікувати і інтегрувати нові пристрої, що, в свою чергу, дозволяє інтегрувати і вдосконалювати виробничі процеси і управляти ними.

Проектування системи розподіленої інтегрованої модульної авіоніки (DIMA) з використанням розподіленої інтегрованої технології, змішаного планування критичних завдань, резервний планування в режимі реального часу і механізму зв'язку, який запускається за часом, значно підвищує надійність, безпеку і продуктивність інтегрованої електронної системи в режимі реального часу. DIMA являє собою тенденцію розвитку майбутніх систем авіоніки. У цій статті вивчаються і обговорюються архітектурні характеристики DIMA. Потім він детально вивчає та аналізує розвиток ключових технологій в системі DIMA. Нарешті, в ньому розглядається тенденція розвитку технології DIMA.

Головною вимогою, що висовується до мережі, це виконання мережею її головної функції — забезпечення доступу до ресурсів всіх комп’ютерів та систем на літаку. В залежності від якості виконання даної функції залежать і інші вимоги, так як: надійність, захищеність, розширюваність, продуктивність та керованість системою ЛА. Основна задача авіоніки на сучасних літальних пристроях полягає в формуванні для пілота та

автоматичних керуючих систем літака найбільш об’єктивної реальної інформації про стан навколишнього середовища та стан ЛА і всіх його систем та агрегатів. Проектування таких систем виконується за допомогою використання відкритої архітектури і інтерфейсів які являються стандартними, що дозволяю в деякій степені прискорити розвиток авіоніки а також зменшити економічне навантаження при розробці нових систем. Ефективне використання модульної архітектури дозволяє нарощувати обчислювальні потужності без заміни інтерфейсів, що на сьогодні актуально як ніколи раніше.

Проектування системи розподіленої інтегрованої модульної авіоніки (DIMA) з використанням розподіленої інтегрованої технології, змішаного планування критичних завдань, резервний планування в режимі реального часу і механізму зв'язку, який запускається за часом, значно підвищує надійність, безпеку і продуктивність інтегрованої електронної системи в режимі реального часу. DIMA являє собою тенденцію розвитку майбутніх систем авіоніки. У цій статті вивчаються і обговорюються архітектурні характеристики DIMA. Потім він детально вивчає та аналізує розвиток ключових технологій в системі DIMA. Нарешті, в ньому розглядається тенденція розвитку технології DIMA.

З розвитком інтегрованої модульної авіоніки (IMA) динамічна реконфігурація IMA не тільки забезпечує великі переваги у використанні ресурсів і конфігурації літаків, але і виступає в якості ефективного засобу управління відмовами ресурсів. Життєво важливо забезпечити корекцію процесу динамічної реконфігурації IMA. Аналіз процесу динамічної реконфігурації є важливим завданням. Мова аналізу і проектування архітектури (AADL) широко використовується в складних вбудованих системах реального часу. Мова може описувати конфігурацію системи і поведінку виконання, наприклад зміни конфігурації. Мережа Петрі є широко використовуваним інструментом для проведення імітаційного аналізу в багатьох аспектах. У цьому дослідженні був запропонований метод аналізу

на основі моделі з безліччю обмежень для процесу динамічної реконфігурації IMA. По-перше, було досліджено кілька конструктивних обмежень процесу. По - друге, процес динамічної реконфігурації був змодельований на основі AADL. Потім був згенерований набір правил для переходу моделі з AADL в мережу Петрі, і запропоновані численні обмеження були включені в мережу Петрі для аналізу. Нарешті, був проведений імітаційний аналіз з кількома обмеженнями за допомогою мережі Петрі для процесу динамічної реконфігурації IMA. Нарешті, для демонстрації цього методу було використано тематичне дослідження. Цей метод вигідний для обґрунтованості динамічної реконфігурації IMA на початку проектування системи.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

AADL - Architecture Analysis and Design Language.

AGREE - Assume Guarantee REasoning Environment.

AIR - Aerospace Information Report.

ARP - Aerospace Recommended Practices.

ASAAC - Allied Standards Avionics Architecture Council

BIP - Behavior Interaction Priority

BSCU - Braking System Control Unit.

CFMs - common functional modules

CPN - colored Petri net

CPU - Central Processing Unit.

DIMA - Distributed integrated modular avionics

GSPN - generalized stochastic Petri net

IMA – Integrated Modular Sistem.

ESM - Existing System Model.

FEM - Failure Effect Modeling.

FLM - Failure Logic Modeling.

HW - Hardware.

MDE - Model-Driven Engineering

MBSA - Model-based Safety Analysis/Assessment.

MBSE - Model-based Systems Engineering.

OSL - operating system layer

OSATE - Open Source AADL Tool Environment.

SysML - System Modeling Language

SAE - Society of Automotive Engineering.

SASM - Safety Analysis System Model.

SW - Software.

WBS - Wheel Brake System.

Розділ 1. | Огляд ключових технологій Розподіленої інтегрованої модульної системи авіоніки

1.1 | Введення

З 40-х років минулого століття розвиток системи авіоніки минуло чотири типових етапи, тобто децентралізовану, федеративниу, інтегровану модульну та вдосконалениу інтегровану модульну [1]. Процес розробки системи авіоніки також відображає процес еволюції інтегрованої технології авіоніки.

У самій ранній розділеній системи авіоніки кожна функціональна область була окремою, і датчик, процесор, дисплей і з'єднання між ними були з'єднані по протоколу точка-точка. Федеральна система авіоніки використовує мультиплексну шину даних з розподілом команд/часу відгуку MIL-STD-1553B, спрощує з'єднання між існуючим обладнанням авіоніки, знижує вагу системи та вплив електромагнітних перешкод, забезпечує обмін інформацією, вирішує комплексні проблеми деяких систем і функцій обробки і змінює традиційну децентралізовану структуру. Таким чином, на основі надшвидкісний інтегральної схеми і загального модуля побудована інтегрована модульна система авіоніки, представлена структурою авіоніки F-22, яка використовує нову технологію після 80-х років.

Були реалізовані три інтегровані функціональні області обробки для обробки сигналів, обробки завдань і управління повітряними судами, і функція обробки сигналів була особливо посилена, вона реалізує три комплексні функціональні області обробки сигналів, обробки завдань і керування повітряними суднами, особливо зміцнюючи можливості інтегрованої основний обробки. Тим самим ще більше поліпшується структура системи авіоніки і досягається більш високий рівень інтеграції. Удосконалена інтегрована система авіоніки розширює область синтезу функцій до області попередньої обробки апертури антени і сигналу датчика, об'єднуючи область обробки сигналів та обробки завдань в комплексну область обробки сигналів і даних і використовуючи відкриту структуру системи, об'єднане мережеве з'єднання авіоніки і готовий модуль, досяг високого ступеня фізичної і функціональної інтеграції.

В останні роки, з розвитком електронної системи космічного апарату НАСА "Оріон" у Сполучених Штатах і дослідженням розподілених електронних модулів у проекті "СКАРЛЕТТ" Сьомої рамкової програми Європейського союзу (FP7), це вказує на те, що майбутня система авіоніки буде розвиватися в напрямку просунутої розподіленої інтегрованої модуляризації [2-7].

Цей документ вперше вводить систему архітектури розподіленої інтегрованої модульної системи авіоніки, а потім докладно розглядаються тенденції розвитку трьох ключових технологій в розподіленої інтегрованої модульної авіоніки, системи, які є змішані дані задач планування і schedulability технології аналізу, в режимі реального часу faulttolerant планування змішаної критичної системи, і затримка аналіз технології комунікації в реальному часі Мережі. Остання частина-це резюме і перспектива.

1.2 | Архітектурні характеристики DIMA

Концепція розподіленої інтегрованої модульної авіоніки (DIMA) заснована на групи відомих в Європі дослідних проектів SCARLETT project, які були зосереджені на архітектурі авіоніки. Вони розробили архітектуру DIMA і визнали, що одним з основних критеріїв проектування DIMA є досягнення суворої ізоляції на фізичному рівні модуля обробки вводу-виводу і модуля обробки додатків [2].

Порівняно з традиційним IMA, це буде серйозною зміною в способі розподілу ресурсів обробки. З одного боку, модуль обробки вводу-виводу може бути розміщений поруч з віддаленими датчиками і виконавчими механізмами, щоб звести до мінімуму ймовірність помилок введення-виведення даних під час передачі. У той же час, він також вирішив проблему розсіювання тепла об'єднавчої плати, тим самим задовольняючи вимогам безпеки конфігурації. З іншого боку, усувається фізичний зв'язок між модулем обробки вводу-виводу і модулем обробки додатків в системі IMA, що забезпечить більш гнучкий вибір для проектування системи.

Як показано на рис. 1, архітектура розподіленої системи модульної електроніки (DME) у проекті SCARLETT показує, що DME суворо ізолює основні компоненти обробки від компонентів обробки вводу-виводу і забезпечує зв'язок для розподілених компонентів через мережу передачі даних avionic (ADCN).

У дослідницької та прикладної практиці система DIMA втілює дві функції інтегрованої модульності" і "розподілу". З одного боку, інтеграції та модульності конструкції системи спільне використання апаратних ресурсів дуже великий, а можливість повторного використання модулів висока. З іншого боку, розподілене розподіл апаратних ресурсів дозволяє ізолювати підсистеми від фізичного рівня.

Система DIMA використовує високоточні тимчасові "віртуальні об'єднавчі плати" для доступу до даних для виконання точного розподіленої обробки. На рисунку 2 наведена принципова схема підключення розподілених апаратних ресурсів через Ethernet з тимчасовим запуском (TTE) в цій архітектурі. TTE забезпечує точний протокол передачі даних по часу для передачі даних для забезпечення загальної тактової синхронізації розподілених ресурсів, механізмів зв'язку з високою цілісністю часу з доступом TDMA [8].

При проектуванні системи DIMA використовуються три ключові технології: змішане планування критичних завдань , надійне планування у реальному часі та аналіз затримок в мережі зв'язку в реальному часі.

Технологія змішаного планування критичних завдань в основному використовується для підвищення надійності всієї системи DIMA і використання системних ресурсів. Технологія резервний планування у реальному часі є основним методом забезпечення надійності розподіленої системи реального часу. Аналіз тимчасової затримки мережі зв'язку в реальному часі є важливим засобом забезпечення продуктивності мережі зв'язку розподіленої системи в реальному часі.

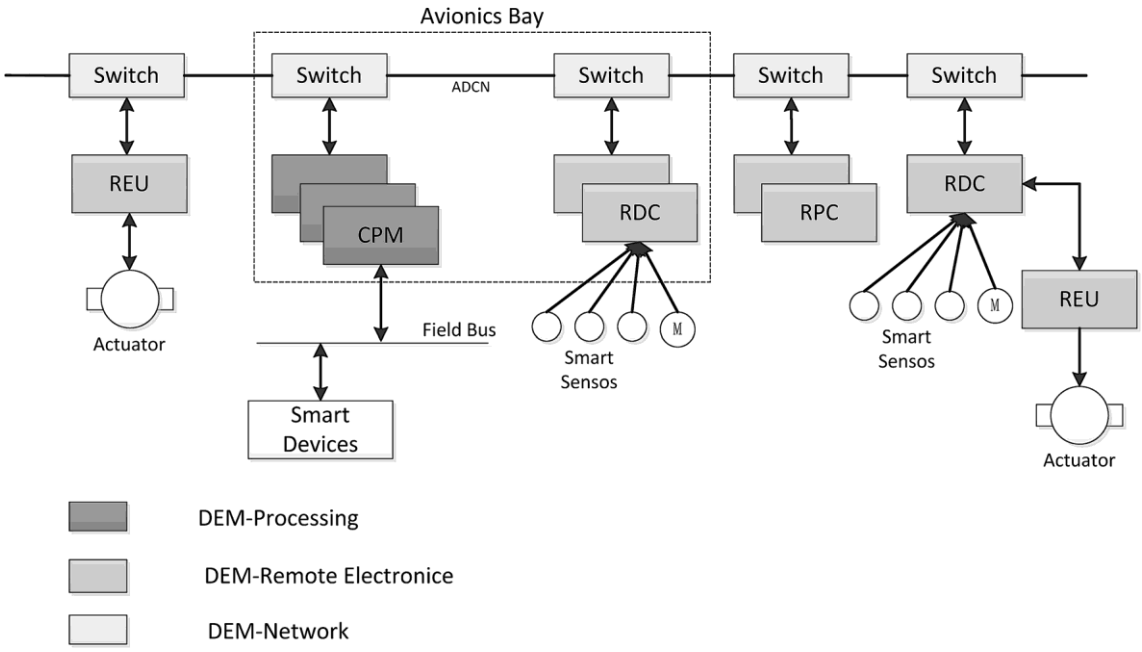


Рис. 1.1 Архітектура розподіленої модульної електронної системи

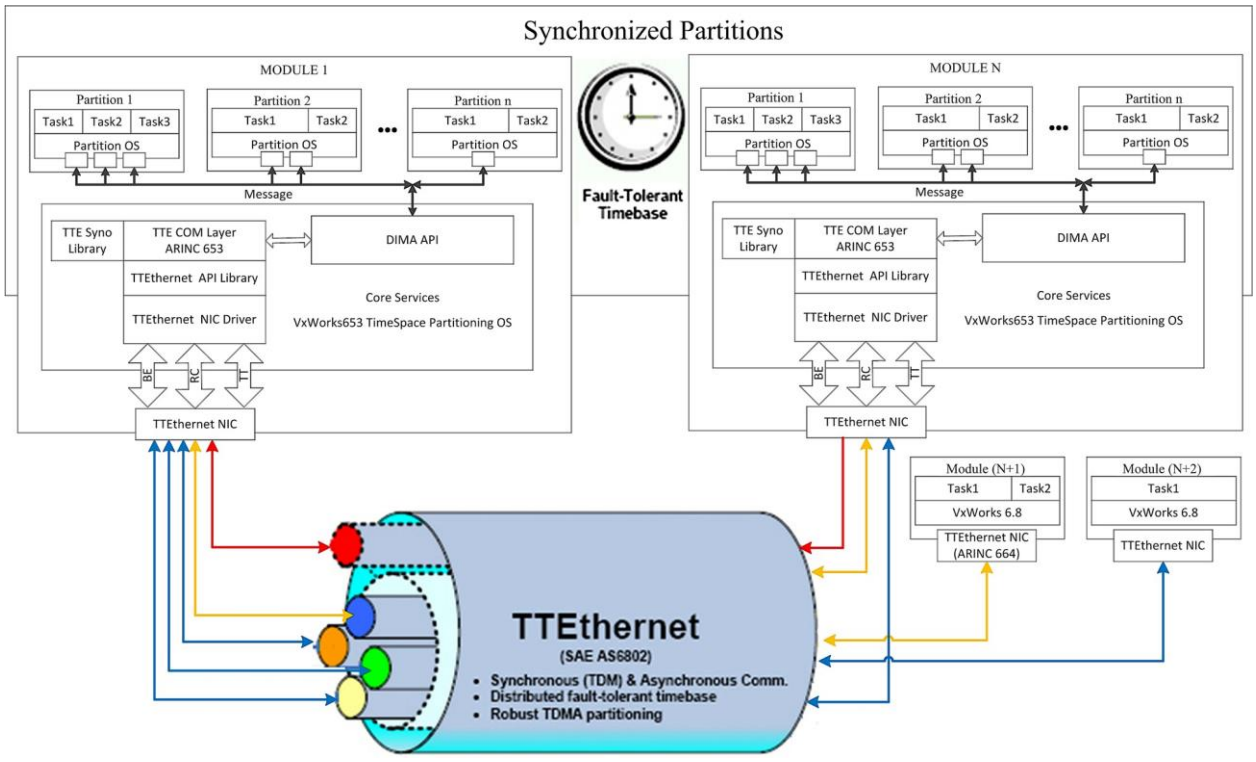


Рис. 1.2 Принципова схема мережевого з'єднання

1.3 | Змішане Планування Важливих Завдань

З безперервними інноваціями в області мікроелектроніки та комп'ютерних програмно-апаратних технологій функції системи авіоніки стають все більш і більш складними. У зв'язку з необхідністю інтеграції інформації і розподілу ресурсів, а також з урахуванням розмірів, ваги і потужності бортової

платформи необхідність зниження експлуатаційних витрат і необхідних фізичних ресурсів змушує системи авіоніки продовжувати розвиватися в напрямку інтеграції і модульності. Різні критичні системні функції мають тенденцію переходити від децентралізованих виділених процесорних блоків до централізованих загальним платформ обробки [1]. За умови, що система авіоніки обробляє і з'єднує ресурси через кордони фізичної обладнання і всі платформи, кілька платформ спільного використання функцій підсистем утворюють інтегровану середу. Критичність використовується для опису ступеня важливості функцій підсистеми або серйозності збоїв. Різні функції підсистеми пред'являють різні вимоги до забезпечення достовірності, відповідні їх критичних рівнів.

Для запобігання шкідливих перешкод між функціональними програмами різних критичних систем і захисту критично важливих для безпеки характеристик систем авіоніки необхідно прийняти певні заходи ізоляції.

Стандарт ARINC653 [9] в області авіоніки визначає технологію розділення для ізоляції різних функціональних додатків, щоб зменшити зв'язок між компонентами і обмежити застосування різних функцій у системі авіоніки її областю діяльності, щоб не впливати на інші функціональні програми, щоб полегшити розробку системи і аналіз перевірки.

У стандарті ARINC653 планування розділів в основному завершує розподіл ресурсів ЦП за допомогою дворівневої моделі планування завдань і реалізує виконання прикладного програмного забезпечення в кожному розділі. Верхній планувальник на рівні операційної системи планує кілька розділів у відповідності з атрибутами розділів і політикою верхнього планування та призначає для кожного з них вікно часу активації.

Нижній планувальник, розташований в кожному розділі, планує завдання, вибравши у вікні активації в відповідності з атрибутами завдань і політикою нижнього планування.

В однокомпонентній системі статичний розподіл ресурсів часто використовується для планування розділів і управління ними. Наприклад, Саевонг та ін [10], Алмейда і Педрейрас [11], Девіс і Бернс [12] обговорили метод розрахунку статичного розподілу ресурсів і алгоритм аналізу максимального часу відгуку завдання для планувальників верхнього і нижнього рівнів з використанням стратегії планування з фіксованим пріоритетом (FP–FP). Однак, враховуючи численні вимоги до перевірки змішаної критичної системи, зміни рівня критичності місії під час роботи

системи викличуть зміни в статусі завдання і потреби в ресурсах межах розділу. Простий підхід до статичного розподілу ресурсів не сприяє повному використанню ресурсів.

Щоб повною мірою використовувати системні ресурси і підвищити загальну надійність системи, необхідно прийняти відповідну модель і алгоритм планування, відповідні критичних рівнів кожної підсистеми, і інтегрувати стратегії планування. В даний час відповідна дослідницька робота з планування завдань для змішаних критичних систем в основному зосереджена на таких двох аспектах: (1) надійність під час виконання; (2) планування системи для перевірки проектування.

1.3.1 | Надійність під час виконання

Надійність робочого часу є ключовою вимогою при проектуванні змішаних критичних систем. Приймаючи деякі заходи тимчасової і просторової ізоляції для запобігання шкідливих перешкод між функціональними додатками систем, зокрема, уникнення функцій системи з низькою критичністю негативно впливає на правильне виконання функцій системи з високою критичністю. При плануванні завдань в режимі реального часу ця вимога надійності також означає, що у випадку, коли не можуть бути гарантовані всі крайні терміни виконання завдань, такі як миттєва перевантаження завдань, необхідно забезпечити виконання критично важливих завдань, в першу чергу.

Хоча вищевказані вимоги не можуть бути досягнуті за рахунок надання виділених системних ресурсів для різних критичних системних функцій, такі методи фізичної ізоляції небажані для систем авіоніки з обмеженими ресурсами при заданих обмеженнях, таких як розмір платформи, вага і потужність. Крім того, як згадувалося вище, використовувана в даний час стандартна технологія поділу ARINC653 не може в повній мірі використовувати ресурси інтегрованої системи і може призвести до проблеми інверсії пріоритетів. Тому для цієї ситуації розроблено новий метод розподілу та планування ресурсів, використання ресурсів поліпшується за рахунок подальшого спільного використання ресурсів, гарантуючи при цьому надійність роботи системи . Традиційне рішення проблеми критичного реверсування полягає в тому, щоб розподілити пріоритети завдань у відповідності з критично важливими завданнями, званими критичністю як призначенням пріоритетів (CAPA), для поліпшення використання процесора. де Низ [8] пропонує алгоритм планування

нульового холостого ходу з асиметричною захистом від перевантаження, керований змішаною критикою. На цій основі Лакшманан та ін [13, 14] провели подальші розширені дослідження. В літературі [13] вивчається метод керування синхронним взаємним виключенням завдань доступу до спільних ресурсів у змішаних критичних системах, а в літературі [14] пропонується метод розподілу ресурсів для змішаних критичних завдань в розподіленому середовищі. Через проблеми критичного зміни, яка може виникнути при плануванні розділів з двома завданнями, Цзінь і Хань [15] вводять критичність на рівні розділів і пропонують механізм динамічного розподілу ресурсів розділів, який може підвищити пропускну здатність системи, уникаючи критичного зміни. Tamas-Selicean і Pop [16, 17] вивчають проблему планування оптимізації змішаних критичних завдань в розподіленій архітектурі моделі планування розділів завдань дворівневого статичного опитування і пропонують метаэвристический метод інтелектуальної оптимізації, заснований на пошуку Табу. Гу та ін [18] вивчили змішане планування критичних завдань для платформи багатоядерних процесорів і запропонували стратегію планування з одним розділом критичності (OCOP) для багатоядерних процесорів. OCOP дозволяє системі перерозподіляти завдання реального часу, поставлені при перемиканні в критичний режим, що, в свою чергу, дозволяє краще збалансувати використання ресурсів кожного процесора в різних критичних режимах. Trüb та ін [19] реалізували гібридний алгоритм планування критично важливих завдань з використанням адаптивного поділу часу в реальних багатоядерних системах і перевірили правильність і ефективність гібридних алгоритмів планування критично важливих завдань в реальних системах авіоніки.

1.3.2 | Планування системи для перевірки проектування

Системне планування для верифікації проектування є ще одним важливим аспектом досліджень змішаних критичних систем, а також є актуальною темою в критичній для безпеки системи авіоніки в реальному часі [20]. Критично важлива для безпеки система–це система, яка після виходу з ладу призведе до значних людських жертв і матеріальних втрат, а також до серйозних збитків. Його аналіз, проектування і перевірка повинні враховувати надійність системних функцій. У зв'язку із зростаючим попитом

на надійність систем в нинішній області авіоніки безпека як важливий аспект надійності також стала важливим обмеженням проектування систем авіоніки. Для різних критично важливих функцій безпеки, що використовують ресурси платформи змішаної критичної системи з загальним доступом, традиційний метод перевірки конструкції перевірить всі функціональні додатки на основі вимог до надійності самого високого критичного рівня. Це консервативне припущення передбачає строгий прогностичний аналіз. Як і надмірне резервування ресурсів, результат дуже песимістичний.

Чим вище критичний рівень функціонування системи, тим більш суворими є вимоги до її передбачуваності і визначеності, тим більш консервативним є відповідний час виконання завдання і тим більш песимістичними є результати Найгіршого часу виконання (WCET). Ґрунтуючись на цьому, Вестал [21] розширила традиційну модель спорадичних завдань [22], вперше застосувала модель змішаних критичних завдань на випадок непередбачених обставин і використовувала метод призначення пріоритетів Одсли [23] в традиційній теорії планування у реальному часі, щоб запропонувати однопроцесорний алгоритм планування з фіксованим пріоритетом для змішування критично важливих завдань. Дорін та ін [24] довели, що алгоритм є оптимальним у всіх однопроцесорних алгоритмах планування з фіксованим пріоритетом. Баруа і Вестал [25] запропонували нові алгоритми планування зі змішаним пріоритетом у поєднанні з FP і EDF, Баруа [26] Розглянув функцію моніторингу часу виконання, питання планування з фіксованим пріоритетом був додатково вивчений.

Література [27, 28] вивчила можливість планування однопроцесорних гібридних критично важливих операцій і довела, що це NP-складна проблема, і запропонувала власний алгоритм планування пріоритетів на основі критичності (OCBP). В літературі [29] наводиться достатня умова запланированности, засноване на завантаженні завдання. В літературі [30, 31] алгоритм OCBP застосовується для планування завдань зі змішаним критичним контингентом. Грунтуючись на дослідній роботі [27, 28], Баруа і Фолер [32] запропонували алгоритм планування з запуском по часу (TT) для однопроцесорних змішаних критично важливих завдань. Баруа та ін [33, 34] запропонували алгоритм динамічного планування пріоритетів EDF-VD, який динамічно коригує віртуальний крайній термін виконання завдання згідно з критичним рівнем системи. На відміну від задачі налаштування EDF-VD з тим же масштабом, стаття [35] дозволяє налаштовувати віртуальний крайній строк окремо для кожної задачі і дає достатню умову для прийняття рішення про планируемости на основі функції попиту на час виконання завдання DBF

[36]. Крім того, в літературі [37, 38] вивчалася проблема планування змішаних критичних завдань в багатопроцесорної середовищі. Santy та ін [39] послабили строгість змішаного критичного планування і дозволили завдань з низькою критичністю тривати протягом певного періоду часу після підвищення критичного рівня системи. Яо та ін [40] прийняли алгоритм планування на системному рівні для передачі критично важливих завдань, використовували розподіл пропускної здатності на основі планування на основі прогалин для критично важливих завдань, опитали критичні завдання і використовували мережеве обчислення для гібридної критичності. Метод планування виконує аналіз в реальному часі.

1.3.3 | Аналіз Системи Планируемости

поведінка передбачувано, це вимагає спільного аналізу планування системного процесора і мережі зв'язку. Аналіз планируемости, заснований на часу відгуку системи, є важливим методом перевірки правильності системного часу. Коли час відповіді транзакції від кінця до кінця менше або дорівнює крайнього терміну транзакції, транзакцію можна запланувати; в іншому випадку транзакція не може бути запланована. Система може бути запланована, коли всі транзакції в системі можуть бути заплановані. Для розрахунку часу відгуку системи від кінця до кінця [41, 42] відповідно запропоновано методи цілісного планування і планованого аналізу для розподілених жорстких систем реального часу. Аналіз планування процесорів і мережі зв'язку на основі аналізу планування протоколу TDMA інтегрований в одну і ту ж структуру. Метод цілісного аналізу планування підтримує лінійну модель транзакцій. Грунтуючись на цій лінійної моделі транзакцій, Паленея довела обґрунтованість методу цілісного аналізу планування [43]. В літературі [44, 45] додатково вивчений метод розрахунку мінімального часу відгуку, заснований на методі цілісного аналізу планування. Щоб вирішити проблему аналізу часу відгуку нелінійних моделей транзакцій, Паленсія запропонував метод підтримки синхронізації декількох подій [46].

Метод цілісного аналізу планування враховує залежності між завданнями на різних процесорах при розрахунку часу відгуку транзакцій від кінця до кінця, але не враховує можливі залежності між різними завданнями на одному і тому ж процесорі, в результаті чого результат аналізу занадто песимістичний, а розрахунковий час відгуку більше, ніж фактичний час відгуку [47]. Для отримання більш точних результатів аналізу в літературі [48] розширено метод за рахунок введення статичних наборів. В літературі

[49] вводяться динамічні набори для аналізу внутрішніх залежностей і досягається більш висока ефективність планування. Pop [50, 51] запропонував завдання аналізу запланированности для систем запуску часу і запуску подій. В літературі [52] пропонується покращений метод для алгоритму глобального планованого аналізу для мережі протоколу TDMA. Ределл і Томгрен запропонували метод аналізу максимального часу відгуку, заснований на початковій фазі даної транзакції [53]. Яо та ін [54] вивчили проблему планування глобального алгоритму планування в багатоядерної процесорної платформи і використовували аналіз зображень функцій для вивчення вимог до системної планируемости різних критичних рівнів, і на основі цього дано точний діапазон ефективних параметрів налаштування віртуального крайнього терміну. На основі моделі планування завдань у поєднанні з теорією планування системи реального часу Мохонг [55] завершив розробку алгоритму аналізу планируемости на основі моделювання для систем реального часу і розробив набір інструментального програмного забезпечення для візуального моделювання та аналізу планування автоматизації. Хан та ін [56] запропонували структуру аналізу планируемости завдань для розподіленої інтегрованої модульної системи авіоніки, яка включала класичну перевірку моделей (MC), перевірку статистичних моделей (SMC), комбіновану перевірку моделей трьома методами для аналізу планируемости системи DIMA в рамках цієї структури. Універсальність і точність методу аналізу забезпечують потужний інструмент аналізу для розробки планування завдань системи DIMA.

1.4 | Надійне Планування У Реальному Часі

1.4.1 | Традиційна Технологія Відмовостійкий

Системи авіоніки працюють в суворих фізичних умовах або навіть в умовах війни. Висока надійність має велике значення і тісно пов'язана з безпекою самого повітряного судна. В критично важливою для безпеки системи авіоніки будь-яка незначна помилка може призвести до непоправного збитку. Щоб гарантувати, що завдання в системі в режимі реального часу можуть бути виконані до закінчення крайнього терміну, навіть якщо система вийде з ладу, необхідно використовувати певні методи для підвищення надійності системи. Запобігання помилок, усунення/тестування помилок, прогнозування помилок і допуск помилок-це розповсюджена міра для забезпечення надійності систем реального часу. Перші три заходи можуть зменшити помилку в системі, наскільки це можливо, завдяки досконалому дизайну, але їх неможливо вирішити з-за

помилок, які не були виявлені в процесі роботи системи, а відмовостійкість-це ініціатива щодо усунення помилок, які можуть виникнути в системі. Таким чином, розробка відмовостійкості є реалістичним і ефективним способом підвищення надійності системи.

Технологія відмовостійкості полягає в підвищенні надійності ресурсів, щоб захистити ефект збою, викликаного надлишком, щоб у разі локального збою система все ще могла виконати алгоритм заданого алгоритму. В залежності від різних ресурсів надмірність можна розділити на чотири види: апаратна надмірність, програмна надмірність, тимчасова надмірність і інформаційна надмірність. Традиційні методи відмовостійкості включають повторне виконання, програмування N-версій [57-59], блок відновлення [60, 61] та інші відмовостійкі методи. Хоча ці традиційні відмовостійкі методи мають важливе прикладне значення для підвищення надійності системи і продовження терміну служби системи, вони не враховують суворі накладні витрати системи в режимі реального часу і системи, тому їх не можна використовувати безпосередньо в області авіоніки з обмеженням ресурсів. Крім того, більшість відмовостійких методів розглядають лише загальні вимоги до надійності з точки зору системи, але ігнорують критичні відмінності між різними функціями системи. Так що в разі збою системи всі завдання будуть без розбору відмовостійкими. Отриманий в результаті рівень відмовостійкості не цілком відповідає вимогам надійності критично важливих функцій системи.

1.4.2 | Змішане Критичне Надійне Планування У Реальному Часі

Відмовостійка технологія планування у реальному часі призначена для управління і планування надлишкових ресурсів, щоб гарантувати, що завдання можуть укластися в терміни навіть у разі збою системи. Це основний метод досягнення відмовостійкості в розподілених системах реального часу. Традиційні алгоритми планування відмовостійкості програмного забезпечення та апаратної відмовостійкості в реальному часі фокусуються на надійності на системному рівні, ігноруючи відмінності в критичності безпеки між різними функціями системи, тобто різні критичні завдання мають відповідно різні вимоги до надійності. Згідно різним моделям помилок, це вимога надійності має різні методи опису в відмовостійких системах реального часу. Модель обмеженої помилки-це зазвичай використовується допущення моделі помилок, яке описує найгірший сценарій помилок, з якими може зіткнутися система, обмежуючи мінімальний інтервал, протягом якого помилки відбуваються послідовно, або

максимальна кількість допустимих помилок, які можуть відбутися протягом певного періоду часу. Однак на практиці системні помилки виникають випадковим чином. Ця випадковість означає, що важко точно отримати кордону параметрів моделі з обмеженою помилкою, що може призвести до більш песимістичною оцінкою в процесі проектування системи. Модель стохастичної помилки може описувати характеристики помилок з допомогою випадкових параметрів, дозволяючи помилок виникати випадковим чином. Однорідний процес Пуассона (HPP) є поширеним методом моделювання випадкових помилок [62].

Для моделі з обмеженою помилкою Добрин та ін [63] забезпечили відмовостійкість для змішаних критичних завдань за рахунок тимчасової надмірності, де ключовою продуктивністю є кількість помилок, які може винести кожен екземпляр задачі. Цей метод використовує цілочисельне лінійне програмування для визначення пріоритету завдання, щоб гарантувати, що кожен критичний примірник завдання може поновити виконання в короткі терміни і вкластися в термін, у той час як некритичні завдання можуть виконуватися з високим пріоритетом для поліпшення використання системних ресурсів. В літературі [64] вивчалася відмовостійкість змішаних критичних систем розподіленої архітектурі. Запропоновано евристичний жадібний алгоритм для визначення міграції критично важливих для безпеки завдань на процесорі з постійним відмовою і налаштування параметрів CBS сервера з постійною пропускною здатністю на звичайному робочому процесорі для максимального підвищення якості обслуговування QoS завдань м'якого реального часу в умовах жорстких вимог до обмеження часу виконання задач реального часу.

В моделі з обмеженою помилкою детерміновані гарантії безвідмовності в реальному часі зазвичай виходять при аналізі статичної планируемости в найгіршому разі. Тобто в даній середовищі операційної системи і можливі дефекти система може бути або запланована і успішно виконана, або не може бути запланована і виконана. Однак, беручи до уваги випадкові характеристики помилок, суворі допущення найгіршого випадку можуть призвести до неточних або занадто песимістичним результатами. Цей абсолютний спрощений аналіз непридатний до моделей випадкових помилок. Для вирішення вищевказаних проблем в літературі [65] вводиться концепція ймовірнісної відмовостійкості в реальному часі, яка ефективно поєднує теорію ймовірностей з аналізом в реальному часі і використовує статистичні методи для аналізу можливості гарантованого планування в рамках моделі стохастичної помилки. На основі цього дослідження запропоновано

імовірнісний метод резервний аналізу в реальному часі в змішаних критичних умовах. Цей метод дозволяє розроблювачам визначати вимоги до надійності на рівні місії у відповідності з їх критичністю і перетворює вимоги до надійності на рівні місії в параметри задачі, які можуть використовуватися алгоритмом планування, і надає метод перевірки запланированности на відповідність системи вимогам надійності кожної задачі.

вимоги до кожної задачі. Донг і Чен [66] запропонували неперіодичну і не включає гетерогенну розподілену динамічну відмовостійку модель в реальному часі, і були представлені два відмовостійких алгоритму планування, DRFSA і DSFSA, для задоволення вимог до надійності і планируемости традиційного розподіленого алгоритму планування у реальному часі. Чжоу і ін [67] запропонували резервний метод планування, засноване на змішаній критичності, для забезпечення можливості відновлення завдань з різними рівнями безпеки при виникненні помилок передачі, що підвищує безпеку і надійність систем реального часу.

1.5 | Аналіз тимчасової затримки мережі зв'язку в реальному часі

У розподіленій системі реального часу завдання, що виконуються на різних процесорних вузлах, взаємодіють за допомогою передачі повідомлень. Щоб гарантувати, що всі завдання можуть відповідати обмеженням по часу, затримка зв'язку між відправкою і отриманням повідомлень повинна бути строго обмежена. Ця затримка зв'язку, яку ми називаємо часом відповіді на повідомлення, відноситься до часу, який проходить з моменту, коли відправляє завдання починає відправляти повідомлення, до моменту, коли отримує завдання отримує повідомлення. Повне час відповіді на повідомлення зазвичай складається з затримки формування повідомлення, затримки очікування, затримки передачі і затримки доставки. Загальний аналіз системної планируемости вищезгаданої системи полягає в простій структурі мережі з топологією шини. Розподілені апаратні ресурси у майбутній системі авіоніки пов'язані між собою через Ethernet з тимчасовим запуском (TTE), використовуючи топологію комутованої мережі, з механізмом зв'язку в режимі реального часу з високою цілісністю в часі з доступом TDMA і більш складними мережевими та комунікаційними протоколами. Аналіз комунікаційних повідомлень в режимі реального часу повинен проводитися у відповідності з обмеженнями інформаційного потоку та правилами обслуговування інтегрованого взаємодії і зв'язку в режимі реального часу.

1.5.1 | Мережеве обчислення

Аналітичні методи і важливі теореми і висновки в деяких мережевих середовищах. Sariowan [68] розширив ці результати досліджень та надав відповідні правила застосування, щоб зробити їх придатними до більш загальної мережевому середовищі. Вона була розроблена та систематизована у систему теорії мережевих обчислень, яка фактично використовується для аналізу продуктивності мережного середовища. Теорія систем. Традиційна теорія аналізу продуктивності мережі використовує теорію випадкових черг для виведення статистичних властивостей мережі [69], таких як середня затримка, пропускна здатність і т. Д., Але верхня межа передбачуваності або затримки повідомлень зв'язку від кінця до кінця більш важлива, ніж статистичні властивості для мереж реального часу. Для аналізу обслуговування мережі в режимі реального часу дослідники створили спеціальний метод аналізу-мережеве числення, вперше запропоноване Рене і Крузом [70, 71], а потім поступово вдосконалене спільними зусиллями таких вчених, як Чанг [72] і Ле Будек і Тирэн [73], Запропоновано аналітичні методи, засновані на кривій прибуття і кривий обслуговування, а також деякі важливі теореми і висновки в мережевому середовищі. Sariowan [68] розширює ці результати досліджень та дає відповідні правила застосування для застосування в більш універсальною мережевому середовищі. Вона буде розроблена та систематизована у систему теорії мережевого обчислення, тобто системну теорію для аналізу продуктивності мережного середовища.

Хоча детерміноване мережеве обчислення дає верхню межу затримки трафіку з обмеженням швидкості, сценарій планування найгіршого випадку є результатом потоків агрегування. Але ймовірність найгіршого випадку визначається іншою схемою планування. Детерміноване мережеве обчислення призведе до песимізму відкладеної оцінки верхньої межі, що призведе до марної трати ресурсів. У порівнянні з детермінованим мережевим обчисленням, стохастичне мережеве обчислення може бути використане для розрахунку верхніх меж параметрів продуктивності при гарантованій ймовірності та скорочення втрат ресурсів. Імовірнісний мережеве обчислення вводить імовірнісні операції в детерміноване мережеве обчислення для опису та аналізу статистичних характеристик мультиплексування мережевих потоків даних. Аналіз продуктивності, заснований на стохастичному мережевому обчисленні, може забезпечити певну ступінь гарантії якості для потоку даних і ефективно поліпшити

використання мережевих ресурсів, а також ефективно заповнити недолік теорії детермінованого мережевого обчислення. В останні роки стохастичне мережеве обчислення широко вивчається і знаходиться в постійному розвитку. Цзян [74] запропонував базову теоретичну основу для побудови стохастичного мережевого обчислення на міжнародній щорічній зустрічі SIGCOMM. Цзян і Лю [75] розширили основні теоретичні основи стохастичного мережевого обчислення і створили набір щодо повної системи теорії стохастичного мережевого обчислення. Робота Цзяна ефективно сприяла теоретичним і прикладним дослідженням стохастичного мережевого обчислення. Філдер [76] узагальнив і проаналізував останні теоретичні результати мережевого обчислення і вказав, що основна теоретична складність поточного стохастичного мережевого обчислення полягає в тому, щоб обчислити верхню межу вирази $P\{\sup 0\leq s\leq t\ [F(s, t)]>x\}$, в якому P{.}- оператор ймовірності (ймовірність події), a sup [.] - операція верхній границі безлічі, a F(s, t) - вираз продуктивності мережі. При використанні стохастичного мережевого обчислення для аналізу продуктивності ми очікуємо отримати верхню межу ймовірності наведеного вище виразу, припускаючи, що ймовірність того, що затримка очікуваного потоку даних перевищує певний поріг в мережі, не перевищує заданого значення ε ($0<\varepsilon<1$). Чжао [77] запропонував дві моделі аналізу затримок, засновані на детермінованому мережевому обчисленні і стохастичному мережевому обчисленні для невизначеності обмеження швидкості (RC) руху в TTE. У детермінованому мережевому обчисленні верхня межа детермінованої затримки RC виходить шляхом побудови кривої агрегованого прибуття трафіку TT і кривий обслуговування трафіку RC. У стохастичному мережевому обчисленні модель розподілу Бернуллі двох станів RC fow будується по граничній теоремі Черрі, і виходить запізніла верхня межа ймовірностей. Чжоу [78] також пропонує режим передачі з кількома пріоритетами для сумісного RC-трафіку в TTE і виводить формулу затримки трафіку. Пропонований алгоритм володіє низькою обчислювальною складністю і високою швидкістю обчислень. Це має високу практичну цінність при застосуванні аналізу часових затримок мережевого обчислення.

1.5.2 | Траекторный Підхід

Траекторный підхід є ще одним методом розрахунку детермінованої верхньої межі часу відгуку від кінця до кінця в розподіленій системі [79-81]. Метод траєкторії відрізняється від методу аналізу, такого як мережеве обчислення, яке не є найгіршою сценою на всіх переданих вузлах, але також не задає криву прибуття і криву обслуговування потоку даних, а обробляє

потік даних у відповідності з випадковим режимом і звертає увагу тільки на найгірший випадок поточного пакету даних на його траєкторії передачі. Тільки в реальній мережі ми можемо увійти у сферу обговорення методу траєкторій, і межа затримки, отримана методом траєкторій, більш точна, ніж мережеве обчислення. Основна ідея методу траєкторії полягає в тому, що максимальна затримка повідомлень може бути отримана шляхом перенесення повідомлень через "період зайнятості" вузлів. Грунтуючись на традиційному методі розрахунку траєкторії, Лабораторія IRIT Університету Тулузи запропонувала метод траєкторій для розрахунку верхньої межі затримки пакетів в мережах AFDX [82]. Згідно траекторному методом мережі AFDX, верхня межа затримки VL становить:

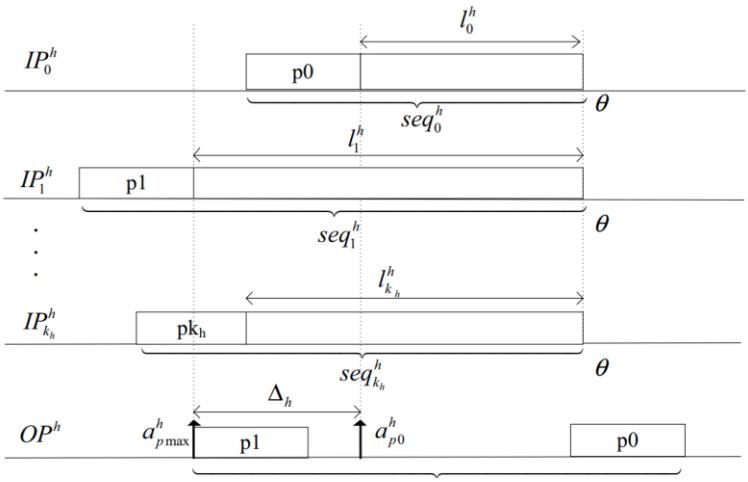


Рис. 1.3. Оптимізація траєкторного методу

$$R_i = \max_{t \geq 0} \left(W_{i,t}^{last_i} + C_i^{last_i} - t \right) \tag{1}$$

ВЛ відповідних даних трафіку записуються як ТІ (i=1,2, ..., п), т-час початку точки генерації кадру даних м ласті є останнім виходом вузла затримки на шляху, Wlasti я,т-це останній раз, що кадр даних м починається на останній вузол, і Clasti я довше часу, необхідного на останній вузол, щоб відправити дані рамки. В літературі [83, 84] оптимізовано метод траєкторій в мережі AFDX, враховані фактори серіалізації кадру даних і в традиційний метод траєкторій впроваджена пакетна технологія. Порівняно з поширеним методом розрахунку мережі AFDX доведено, що метод траєкторій, заснований на мережі AFDX, є більш компактним методом. Формула для розрахунку верхньої межі затримки в VL оновлена до:

$$R_i = \max_{t \geq 0} \left[\left(W_{i,t}^{last_i} - \sum_{\substack{h \in P_i \\ h \neq first_i}} \max\left(0, \Delta_h\right) \right) + C_i^{last_i} - t \right] \quad (2)$$

У формулі максимальної затримки методу оптимальної траєкторії доданий зменшення являє собою суму значень затримки надлишкового обчислення. Як показано на рис. 3, в порту пересилання h IPh 0 є вхідний чергою цільового кадру даних m, а IPx h ($1 \leq x \leq kh$) є інший вхідний чергою. seqh x ($0 \leq x \leq kh$) - послідовність кадрів даних вхідної черги IPx h , яка складається з кадрів даних VL, що передаються через порт пересилання h. Всі кадри з послідовності кадрів IPx h ($1 \leq x \leq kh$) будуть мультиплексованих у вихідну чергу OPh . Щоб максимізувати затримку цільового кадру даних m в порту h, всі послідовності кадрів планується завершити одночасно θ. Навіть у такому найгіршому разі цільової кадр даних m (належить VLi у вхідній черзі IPx h) не буде заблокований одночасно усіма кадрами даних в інших чергах IPx h ($1 \leq x \leq kh$). Δh розраховується як формула (3), тому оптимізація методу траєкторії в кожному порту пересилання складає максимум від 0 до Δh.

$$\Delta_h = \max_{1 \leq x \leq k_h} \left(\min\left(l_x^h\right) \right) - \max\left(l_0^h\right) \quad (3)$$

Існують також деякі методи [85-87] для розрахунку мережевої затримки, такі як метод моделювання [88, 89] і метод перевірки моделі [90]. Однак метод моделювання не аналізує верхню межу затримки. Хоча метод перевірки моделі може визначити верхню межу затримки мережевого трафіку, вибух простору станів обмежений розміром мережі.

1.6 | Тенденція майбутнього розвитку технологій

В аспекті планування завдань ARINC653 визначає правила планування моноядерних розділів. Коли приймаються до уваги системи обробки кількох ядер, важливим питанням стає те, як реалізувати планування розділів з декількома ядрами. У той же час система інтегрована з плануванням розділів і мережевим плануванням для реалізації загального уніфікованого планування системи. Стратегія планування з декількома основними

розділами і метод розподіленого спільного планування при змішаних критичних характеристики стануть важливими проблемами, які необхідно вивчити і вирішити в майбутньому. У плані відмовостійкості системи, необхідно більш докладно розглянути, як використовувати час спрацьовує синхронізація часу механізм і локалізація несправностей характеристик через раз спрацьовує архітектури, щоб зробити зайвим похибка та реконструкції системи розподіленої системи управління ресурсами, у тому числі високу точність синхронізації часу алгоритм, стратегію обміну системи управління, засновані на пулі ресурсів і ресурсів спосіб реконструкції в розподіленої інтегрованої модульної системи.

В аспекті аналізу затримки в мережі зв'язку питання про те, як побудувати строгий метод оцінки, завжди було актуальним питанням при дослідженні систем реального часу. При розгляді характеристик поділу, запуску і змішаних критичних характеристик метод оцінки системи в реальному часі у відповідності з спільною стратегією планування розподілу і запуску по часу та метод оцінки системи в реальному часі у відповідності зі змішаними критичними характеристиками можуть використовуватися в якості напрямків досліджень для подальшої роботи.

1.7 | Висновок

Інтегрована розподілена модульна система авіоніки - це напрям розвитку авіоніки наступного покоління, який може ефективно підвищити інтелектуальний рівень і надійність системи авіоніки і одночасно знизити вартість. У цьому розділі вивчаються та аналізуються особливості архітектури DIMA, а також аналізуються і докладно обговорюються дослідження і розробки трьох ключових технологій в системі DIMA за останні роки. Нарешті, висувається тенденція розвитку технології DIMA future. Результати досліджень, представлені в даному розділі, можуть забезпечити теоретичну підтримку для дослідження і проектування розподіленої інтегрованої модульної системи авіоніки для нових літаків у майбутньому.

Розділ 2 «Новий Метод аналізу на основі моделі з безліччю обмежень для Процесу Динамічної реконфігурації Інтегрованої модульної авіоніки»

2.1. | Введення

Авіаційна система розробляється від дискретної до федеральної і інтегрованою модульної авіоніки (IMA). Система має більш відкриту і більш складну архітектуру. Система IMA виконує функції на основі загальних функціональних модулів (CFM). CFM допомагають зменшити вагу і розміри літака. В системі IMA різні програмні функції виконуються на CFM. Програмна система високо інтегрована з-за своєї складної структури.

На основі опису IMA в ARINC 653 та Ради з архітектури авіоніки союзних стандартів (ASAAC) [91,92] архітектура програмного забезпечення має трирівневу структуру. Прикладне програмне забезпечення спочатку зберігається на пристрої зберігання даних, а не в CFMS. Програмне та апаратне забезпечення не є обов'язковим. Програмне забезпечення може використовуватися на різних апаратних засобах з різними конфігураціями. В цілях безпеки літаки зазвичай перезапускають додатки резервними резервними копіями, як тільки відбувається якийсь збій. У цій статті динамічна реконфігурація відноситься до змін конфігурації, що виконуються при виникненні збою під час польоту. Динамічна реконфігурація може допомогти у створенні нових областей резервного копіювання для перезапуску програми, що робить площину більш гнучкою і більш ефективно використовує апаратні ресурси.

Існує безліч досліджень динамічної реконфігурації з точки зору статики [93-96]. Дін, М. [97] запропонував підхід до побудови та перевірки моделі автомата для діаграми активності мови системного моделювання (SysML), щоб забезпечити відповідність логічної архітектури реконфигурированной системи функціональним вимогам. Шукла, Дж. [98] запропонував методологію реконфігурації розподільної системи з обмеженою стабільністю сигналу (DSR) в умовах невизначеності, пов'язаної з потребою в навантаженні і вихідною потужністю розподіленої генерації на основі відновлюваних джерел енергії. Елліс, С. М. [99] встановив можливість забезпечення відмовостійкості

апаратного забезпечення за допомогою динамічної реконфігурації програмного забезпечення і продемонстрував його життєздатність в контексті типового авионического додатки в реальному часі. Однак існує кілька досліджень, присвячених процесу динамічної реконфігурації. Слід враховувати правильність процесу динамічної реконфігурації. Розуміння обмежень, пов'язаних з динамічною реконфигурацией, може допомогти в правильному і плавному завершення процесу. Однак, як змоделювати і додати обмеження для аналізу динамічної реконфігурації, є складним завданням. Для аналізу процесу динамічної реконфігурації у цьому дослідженні ми запропонували метод, заснований на моделі, з декількома обмеженнями. Моделювання динамічної реконфігурації IMA може принести користь аналізу.

Система IMA - це вбудована система реального часу [100]. Мова аналізу і проектування архітектури (AADL) [101] є міжнародним стандартом SAE (раніше відомим як Товариство автомобільних інженерів) (SAE AS5506) [102], заснованим на інженерії, керованої моделями (MDE). У AADL використовуються концепції моделювання для опису програмно-апаратної архітектури та середовище з точки зору окремих компонентів та їх взаємодії [103], і це особливо ефективно для моделювання складних вбудованих систем реального часу [104]. Тому AADL широко застосовується у вбудованих системах, особливо в аерокосмічних системах [105-108]. Чжан Ф. [109] застосував AADL до моделі F-16 "Контролер автопілота" і проаналізував властивості поведінки живучості та уточнення трасування з різними припущеннями про справедливості, враховуючи часові можливості і терміни. Чжао, З. [110] побудував модель AADL для складної апаратної структури і надійного програмного забезпечення системи відображення avionic. Лю, З. [111] представив метод моделювання розподілу IMA на основі AADL з двох аспектів: моделювання архітектури та моделювання політики планування, а також проаналізував можливість планування поділу IMA.

AADL не тільки описує компоненти системи, але також описує поведінку системи та інші елементи, такі як режим і всі типи додатків. Режими можуть представляти різні стану конфігурації системи або компонента, коли подія викликає зміну режиму. Всі ці функції роблять AADL хорошим методом для опису процесу переходу системи, наприклад, динамічної реконфігурації IMA.

Однак AADL-це тільки напівформальна модель, і вона ще не дозріла для аналізу надійності [112]. Неточно використовувати AADL для аналізу надійності вбудованих систем [113]. Хоча AADL забезпечує ефективну підтримку моделювання вбудованих систем, вона повинна бути формалізована,

щоб зробити модель зручною для формальної перевірки [114]. Перетворення моделей займають центральне місце в інженерії, керованої моделями (MDE), де вони використовуються для перетворення моделей між різними мовами; для рефакторінгу та моделювання моделей або для створення коду з моделей [115,116]. Тому вчені і галузі промисловості, як правило, використовують методологію перетворення моделей для перевірки та аналізу моделі AADL з використанням існуючих інструментів перевірки та аналізу [117]. Було запропоновано багато досліджень про перетворення AADL: перетворення AADL в пріоритет поведінкового взаємодії (BIP) [118], в Fiacre [119], у мережі Петрі [120], EDA (Автомати даних про події) [121] і т. Д. Метою такого перекладу є повторне використання існуючих інструментів перевірки та аналізу і їх формальної моделі обчислень і зв'язку з метою перевірки моделей AADL [103].

чи мету перевірки моделей AADL [13]. Мережі Петрі-це формальний графічний і математичний інструмент, здатний моделювати і аналізувати динамічну поведінку систем. Вони також все частіше використовуються для оцінки безпеки, надійності і ризиків систем [122,123]. Мережі Петрі зарекомендували себе як потужний інструмент моделювання та аналізу використовуються при моделюванні та аналізі кооперативних систем і систем дискретних подій завдяки їх обґрунтованої формалізації [124]. Потім мережа Петрі розширюється до кольоровий мережі Петрі (CPN) [125-127], узагальненої стохастичної мережі Петрі (GSPN) [128-130], нечіткої мережі Петрі [131] для підвищення її здатності опису, щоб вона могла більш ефективно моделювати динамічний процес. Тому мережі Петрі широко застосовуються для оцінки безпеки, надійності і ризиків систем у багатьох областях. Li та ін [132] запропонували метод моделювання надійності на основі PN, коли оцінювалась надійність системи з урахуванням залежності механізму відмови. Віланд та ін [133] запропонували модель на основі PN для розрахунку даних про надійність стеків паливних елементів з полімерним електролітом і мембраною. Дані про надійність включають середній термін служби одного пакета або надійність пакетів всього парку транспортних засобів на паливних елементах протягом заданого часу. Сунанда та ін [134] запропонували підхід до моделювання несправностей на основі мережі Петрі, і цей підхід був підтверджений шляхом його застосування до прототипу системи перетину залізничних шляхів. Чи, У. [135] запропонував новий багаторівневий метод моделювання і обґрунтування нечітких мереж Петрі для оцінки ризику відмови технологічного обладнання, щоб чітко описати взаємозв'язок та зробити обчислювальний процес гнучким. Гонсалвес, П. [136] представив моделювання процесу оцінки безпеки

безпілотного літального апарату за допомогою мереж Петрі для аналізу аварійних умов, які призводять до найбільш небезпечних подій. Лю, Р. [137] перевів модель AADL в GSPN для аналізу та оцінки надійності платформи системи IMA. Li, Z. [138] запропонував аналіз небезпеки за допомогою покращеної тимчасової CPN з обмеженням безпеки у часі і просторі для IMA.

Однак в аспекті моделювання вбудованих систем мережі Петрі страждають від комбінаторних проблем і проблем складності, і, отже, їх важко використати при моделюванні складних систем зі значним числом станів [139]. Тому в цьому дослідженні ми застосували AADL для моделювання процесу динамічної реконфігурації IMA, а також для моделювання і аналізу моделі шляхом перетворення в кольорові мережі Петрі.

Інша частина цієї статті організована наступним чином: Розділ 2 містить короткий вступ в динамічну реконфігурацію IMA, AADL, класичну мережу Петрі, CPN і т. д. У розділі 3 пропонується набір обмежень для динамічної реконфігурації. Потім у розділі 4 представлено метод аналізу, що містить три етапи. По-перше, процес динамічної реконфігурації моделюється на основі AADL, і в цю модель додаються властивості, пов'язані з обмеженнями. По-друге, модель AADL перетворюється в CPN в деяких конкретних правилах. По-третє, процес динамічної реконфігурації моделюється на основі мережі Петрі. Для докладного опису цього методу в розділі 5 пропонується тематичне дослідження. Висновок за нашим дослідженням і майбутньої дослідницької роботи наведено в розділі 6.

2.2. | IMA

2.2.1. | Архітектура програмного забезпечення IMA

Система IMA-це складна система, яка має відкриту архітектуру, більш широку інтеграцію, більш інтегровані функції і високу зв'язок між модулями. Багато проблеми також виникають при перенастроюванні. Динамічна реконфігурація в цьому дослідженні відноситься до програмного забезпечення. Потім у цій статті представлена архітектура програмного забезпечення IMA. Система IMA включає в себе основну систему IMA і неосновне обладнання у відповідності зі стандартом ASAAC. Базова система IMA містить кілька авионических стійок. Ці стійки містять CFM та мережі зв'язку між ними. Крім того, стійки мають функціональні додатки, засновані на апаратне забезпечення, операційної системи і програмне забезпечення для управління системою.

CFM забезпечує обчислювальну потужність, мережеву підтримку і перетворення потужності для основної системи IMA. Програмна система розділена на три рівні—рівень підтримки модулів (MSL), рівень операційної системи (OSL) і рівень додатків (AL). MSL надає інтерфейс для вищенаведеного рівня для доступу до ресурсів та відокремлює операційну систему і апаратну платформу. OSL включає в себе операційну систему в реальному часі і управління системою. Прикладне програмне забезпечення та управління додатками виконуються на AL. Загальне управління системою (GSM) OSL, яке виконує управління системою, налаштовує систему і управляє нею за допомогою доступу до схеми системи. GSM включає в себе управління здоров'ям, управління несправностями, управління конфігурацією та управління безпекою. Управління додатками є частиною управління системою і управляється на основі програми. Архітектура програмного забезпечення наведена на рисунку 1.

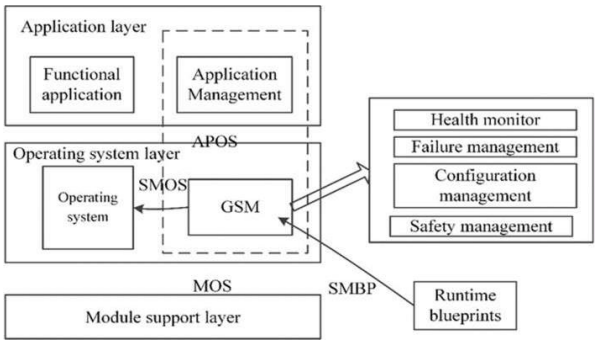


Рис. 2.1. Архітектура програмного забезпечення інтегрованої модульної авіоніки (IMA).

2.2.2. | Механізм реконфігурації IMA

Реконфігурація IMA включає як статичну, так і динамічну реконфігурацію [140,141]. Деякі механізми є загальними як для статичного, так і для динамічної реконфігурації. В основній системі IMA всі програмні додатки зберігаються у масовій пам'яті. Коли система ініціалізований, додатки завантажуються на цільовий модуль [142]. Ця операція зменшує необхідність у технічному обслуговуванні і гарантує, що модуль може бути замінений. При виникненні деяких збоїв диспетчер працездатності виявляє збій і повідомляє диспетчеру несправностей, щоб він його обробив. Диспетчер несправностей може обробляти серію збоїв по порядку при всіх типах механізмів. З одного боку, диспетчер несправностей може виявляти, знаходити і пов'язувати збої, а потім повідомляти про результати аналізу на верхній рівень. З іншого боку, система запитує у менеджера конфігурації почати перенастроювання, щоб уникнути збою. Таким чином, диспетчер несправностей синтезує безліч технологій

управління несправностями. Потім диспетчер конфігурацій починає свою роботу після отримання повідомлення від диспетчера збоїв [143,144].

У ASAAC є багато принципів для перенастроювання. Система IMA повинна бути перенастроєна між стабільними станами. Реконфігурація повинна якомога швидше зупинити поширення несправності. При проектуванні системи необхідно враховувати всю ситуацію, коли існує деяка реконфігурація. Перенастроювання передбачає переміщення програми з-за деяких вимог або в разі збою. Дії з налаштування виконуються у відповідності з порядком, зазначеним в схемі. Рейс та ін [145] вказали, що перенастроювання корисна, коли існують динамічні нефункціональні вимоги, дефекти обладнання або вимоги до додатків для системи.

Однак існують відмінності між статичної та динамічної реконфігурації. Зазвичай система статичної реконфігурації включає в себе резервні модулі. Наприклад, CRACK2 є надлишковою тріщиною для CRACK1. Потім CFM3/REP2 є резервною копією для CFM3/REP1. При збої CFM3/REP1 додатки запускаються на CFM3 і можуть бути перенесені в REP2 з REP1, щоб система не вийшла з ладу. Статична реконфігурація системи представлена на рисунку 2.

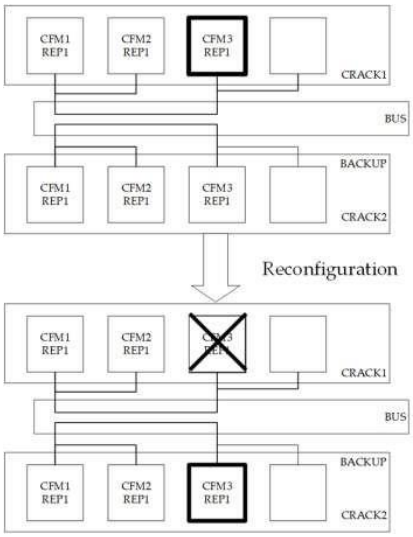


Рис. 2.2. Конфігурація системи з резервними модулями.

2.2.3. | Пов'язані з цим роботи з динамічної перенастроюванні

На основі ARINC 653 [146] помилки (несправності) в IMA і механізмах реагування класифікуються, як показано в таблиці 2.1.

Таблиця 2.1 Класифікація помилок і механізми реагування.

Рівень помилок	Приклади помилок	Механізм реагування на помилки
Рівень модуля	• Помилка програми викликана процесом додатки • Незаконний запит O/S • Помилки виконання процесу (переповнення, порушення пам'яті і т. д.)	• Ігнорувати • Вимкніть модуль • Перезавантажте модуль • Дії по відновленню, визначені реалізацією
Рівень Розділу	• Помилка в таблиці конфігурації розділів при ініціалізації розділів • Помилка ініціалізації розділу • Помилки, що виникають приуправлении процесами • Помилки, що виникають під час процесу обробки помилок	• Ігнорувати • Зупиніть розділ (в режимі ОЧІКУВАННЯ) • Перезавантажте розділ
Рівень Процесу	• Помилка в таблиці конфігурації модуля при ініціалізації модуля • Інші помилки при ініціалізації основного модуля • Помилки при виконанні специфічних для системи функцій • Помилки при перемиканні розділів • Збій живлення	• Проігноруйте помилку n раз до відновлення дії. • Ігноруйте, реєструйте збій , але не робіть ніяких дій. • Зупиніть несправний процес і повторно ініціалізувати його адреси введення. • Зупиніть несправний процес (припустимо, що розділ виявляє і відновлює). • Перезавантажте розділ • Зупиніть розділ (в режимі ОЧІКУВАННЯ).

Тому, коли в системах IMA виникають помилки, механізми реагування, наприклад, методи відмовостійкості і т. д. почне відповідати першим. Коли механізми реагування не зможуть усунути помилку (несправність), помилка викличе динамічну реконфігурацію. У цьому дослідженні ми прагнемо обговорити і проаналізувати ситуації після початку процесу реконфігурації, тому ми припускаємо, що механізми реагування не можуть усунути несправність і що вона викликала реконфігурацію.

Динамічна реконфігурація IMA відбувається під час роботи системи. У цьому дослідженні реконфігурація відноситься тільки до програмного забезпечення, оскільки апаратний збій незворотній [147]. Динамічна реконфігурація IMA може змінювати свої завдання в залежності від вимог і швидко відновлюватися після збою [148]. Це робить систему більш гнучкою, знижує надмірність обладнання і витрати на позапланове технічне обслуговування. Більше того, коли в цьому бере участь людина, складність динамічної реконфігурації зростає. Потім визначення того, як обмежити процес, щоб забезпечити його безпеку, стає проблемою. Багато дослідники дослідили поліпшення динамічної реконфігурації у всіх типах аспектів.

безліч аспектів. Topping, C. [149] представив динамічно реконфигурируемый модуль обробки (DRPM) для динамічної реконфігурації. DRPM складається з перепрограммируемых польових програмованих вентильних решіток (ПЛІС), що є базовим обладнанням, необхідним для зручної перенастроювання. Cyo [150] припустив, що традиційні методи аналізу в основному зосереджені на відмову компонентів. STPA використовується для виконання аналізу ризиків, який фокусується на людських чинниках, що лежать в основі динамічного процесу. Тимчасові планувальники запропоновані в [151] для планування процесу динамічної реконфігурації в умовах жорстких часових і ресурсних обмежень. Підготовка плану реконфігурації в минулому була важкою для більшості дослідників з-за відсутності автоматизованих та інтелектуальних інструментів. Вони автоматизували динамічну реконфігурацію, використовуючи тимчасові планувальники штучного інтелекту (ШІ), щоб зменшити її складність. Планувальники ІЇ проводять оптимізоване планування завдань, що ускладнює для людей визначення того, коли система реконфигурируется. Монтано [152] визначив елементи динамічної реконфігурації своєї дисертації. Динамічна реконфігурація критичних пілотованих систем (СКМС) безпеки обумовлена подією зміни функцій або ресурсів відповідно до вимог оператора. У дисертації обговорюються питання автоматизації та участі людини при динамічної реконфігурації критично важливою для безпеки системи.

удосконалення в ході динамічної реконфігурації критично важливою для безпеки системи. Для моделювання динамічної реконфігурації IMA Чжан [153] запропонував метод надійності, заснований на AADL для реконфігурації IMA. Потім система була переведена в мережу Петрі для аналізу надійності. Розрахунок надійності виконується для компонентів архітектури системи. Суо представив метод для рішення проблем в реальному часі при реконфігурації в іншому дослідженні [154]. Система IMA також моделюється з використанням AADL. Потім система переводиться в TPN для перевірки. Перш за все, аналіз процесу динамічної реконфігурації не привернув особливої уваги. У цьому дослідженні був запропонований метод аналізу процесу, заснований на моделях для підвищення коректності, безпеки і надійності динамічної реконфігурації.

2.3. | AADL

AADL-це ефективний інструмент моделювання для аналізу вбудованих систем у реальному часі і складних систем. У цьому дослідженні AADL був використаний для моделювання процесу динамічної реконфігурації IMA.

2. 3.1. | Компоненти

Компоненти є основними елементами системи. Компоненти розділені на три набору— програмне забезпечення, апаратне забезпечення і композитний. Стан конфігурації системи може бути описане за допомогою AADL. Крім того, можна описати структуру системи та пристрої. Програмна архітектура системи IMA розділена. Структура логічної конфігурації потребує додатку ARINC 653 в AADL. Елементи ARINC 653 сформували архітектуру системи і відповідають компонентам AADL [155].

2.3.2. | Режими

Стабільний стан конфігурації системи при динамічної реконфігурації представлено режимом. Режими можуть представляти різні стани системи або компонента, з'єднання і асоціації значень властивостей [156]. Переходи визначають режим, коли система динамічно перенастроюється в нову конфігурацію. Текстове і графічне представлення специфікацій переходу в режим для простого прикладу показано на рисунку 3.

2.3.3. | Додаток про поведінку

Кожен стан системи і докладні переходи виражаються в додатку "Поведінка" в AADL. Додаток "Поведінка" визначає специфікації поведінки компонентів AADL більш точно, ніж ядро мови. Поведінка, описане в цьому додатку,

засноване на змінних стану, еволюція яких визначається переходами, які можуть бути охарактеризовані умовами і діями [157].

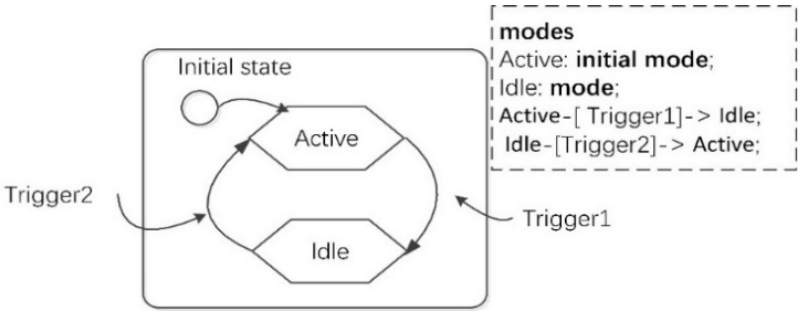


Рис. 2.3. Перехід в режим.

2.4. Мережа Петрі

AADL широко використовується для моделювання вбудованої системи. AADL не може виконати імітаційний аналіз динамічного процесу візуально. Мережа Петрі - це сумісний інструмент для проведення імітаційного аналізу.

Мережа Петрі-це інструмент графічного та математичного моделювання для опису паралельних і асинхронних систем. Мережі Петрі можуть не тільки моделювати динамічну діяльність або передачу інформації в мережі, але і використовувати математичні моделі для управління поведінкою систем [158]. Мережа Петрі описується як трьохкорпусний $PN = (P,T,F)$ [159], де P-кінцевий набір місць; T-кінцевий набір переходів; F -набір спрямованих дуг. Класична мережа Петрі включає в себе місця, переходи і дуги між місцями. Стан системи описується місцем. Переходи являють собою процес зміни систем. Дуги від переходу до місця або від місця до переходу мають свою вагу. Жетони присутні в кожному місці, щоб показати стан цього місця. Токени також використовуються для подання даних або ресурсів. Однак у класичній мережі Петрі є деякі недоліки. Класична мережа Петрі не має уявлення про час. Більш того, метод опису класичних мереж Петрі занадто одиничний. Таким чином, існує безліч розширень і доповнень для мереж Петрі. Були запропоновані деякі високорівневі мережі Петрі, такі як CPNs і тимчасові мережі Петрі.

CPN-це високорівневі мережі Петрі, що використовуються для проектування, аналізу специфікацій, валідації та верифікації [160,161]. CPN є кортежем = (Σ, P,T, A, N, C, G, E, I) [162], де: Σ - це кінцеве безліч пустих типів, також званих колір комплекту; P являє собою кінцеве безліч місць; T - кінцева множина переходів; A це кінцева множина дуг; N є функціональним вузлом; C - колірна функція; G охоронна функція; E - дуга вираз функції; I - функція ініціалізації. CPNs може описувати стану складних систем і зміни стану, викликані запускаючих подіями. Особливістю CPN є те, що він надає визначення наборів

кольорів. Набір кольорів, прикріплений до місця, містить жетони. Кожен жетон повинен мати свій колір. Охорона переходу повинна бути задоволена до того, як перехід буде здійснено. CPN об'єднує мережі Петрі і стандарт мови програмування ML [163].

Останнім часом було проведено декілька досліджень підходів до аналізу динамічної реконфігурації на основі моделей. Метод надійності, заснований на AADL для реконфігурації IMA, був запропонований Чжаном [153]. Розрахунок надійності компонентів системи проводиться після переходу моделі з AADL на мережу Петрі. Suo [154] представив метод, змодельований AADL, і переведений в TPN для рішення проблем в реальному часі при реконфігурації. Ван дер Аалст [164] зазначив, що мережі Петрі не тільки використовуються як мови проектування для специфікації складних робочих процесів, але також надають потужні методи аналізу для перевірки правильності процедур робочого процесу. Наскільки нам відомо, ні один підхід до аналізу не був зосереджений на процесі динамічної реконфігурації. У цьому дослідженні був запропонований метод заснований на моделі аналізу процесу динамічної реконфігурації для виконання аналізу небезпеки.

2.5. | Численні обмеження для процесу динамічної реконфігурації

Безліч обмежень для процесу динамічної реконфігурації Тут був запропонований набір обмежень для процесу динамічної реконфігурації IMA. Обмеження включають в себе багато аспектів, такі як стану системи, можливості в реальному часі і можливості використання ресурсів. Всі зазначені обмеження були інтегровані в метод аналізу для перевірки правильності проектування динамічної реконфігурації IMA.

2.5.1. | Обмеження стану системи для динамічної реконфігурації

Перед запуском динамічної реконфігурації слід провести деякі попередні перевірки модулів в системі, за винятком модуля збою. Якщо відбулося поширення помилок, слід відмовитися від первісної стратегії параметри динамічної реконфігурації. Слід розглянути можливість нової динамічної реконфігурації. Задається логічне значення S, що представляє початковий стан системи. Якщо інші модулі системи також виходять з ладу після поширення, то S = 0. Таким чином, реконфігурація завершується. Якщо немає ознак збою, система може почати реконфігурацію, S = 1.

2.5.2. | Обмеження в реальному часі для переходу Системи в стан

Короткий вступ в процес динамічної реконфігурації наведено в частині 2. В ході цього процесу система переходить з одного стану в інший за ініціюючих подій і дій. Це важлива проблема, в якій повинні бути дотримані часові рамки. Якщо час між двома станами дуже велике, це впливає на наступне стан і викликає деякі небезпеки реконфігурації. Властивості аналізу повинні бути додані в модель.

Властивість часу пов'язане з переходом з одного стану в інший. Щоб гарантувати тимчасові обмеження, було запропоновано алгебраїчне рівняння, яке порівнює суму часу, витраченого всіма підстанціями в процесі динамічної реконфігурації, зі значенням обмеження.

На рисунку 4 показано, що в процесі є п'ять станів. Існує тригер для переходу між станом 0 і станом 1. Час переходу одно T_1. Існує дія, затрачивающее час T_2 між станом 1 і станом 2. Те ж саме відноситься і до інших штатів. Кожна дія або перехід між двома станами у процесі позначається часом, витраченим *на* T_i. Граничне час становить t. Тоді загальне витрачений час T_s дорівнює сумі T_i . Весь час задовольняє рівнянню (1).

$$T_S = T_1 + T_2 + \ldots + T_n = \sum_{i=1}^{n} T_i \tag{1}$$

Поведінка динамічної реконфігурації повинно підкорятися нерівності (2):

$$T_S \leq t \tag{2}$$

Використовуючи це рівняння, порівняння різних значень часу може дати нам результат, чи може система досягти цілей обмеження в реальному часі.

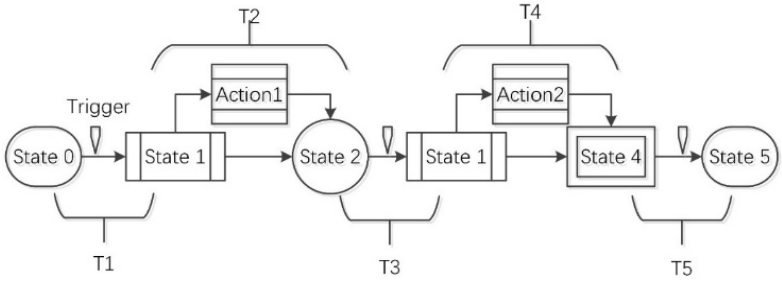


Рис. 2.4. Час переходу системи в інший стан.

2.5.3. | Обмеження пам'яті для стану системи

Аналогічно обмеження по часу, всіма операціями потрібно місце в пам'яті, як показано на рис. 5.Очевидно, що обсяг пам'яті, що виділяється кожним станом під час динамічної реконфігурації, може бути змінений. Обмеження на розмір пам'яті повинні бути гарантовані, незалежно від того, як змінюється обсяг пам'яті підстанцій.

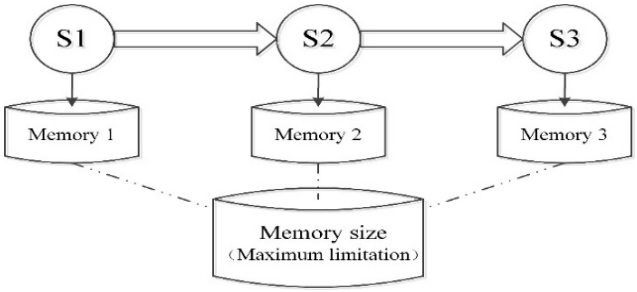


Рис. 2.5. Пам'ять для стану системи.

Кожне стан, включаючи дії, займає пам'ять . Максимальне обмеження об'єму пам'яті становить . Тоді слід дотримуватися нерівність (3).

$$M_i \leq m(i = 1,2\ldots,n) \tag{3}$$

2.5.4. | Обмеження можливостей для спільного використання даних ресурсів

В різних режимах роботи системи компоненти системи взаємодіють з компонентами даних шляхом читання і запису. Спільне використання ресурсів, таких як дані, повинно бути позначено серійним номером, що належать до часу після експлуатації в різних станах. Якщо після зміни даних в змозі 1 на компонентах даних немає позначки, система не може вирішити, чи є дані результатом, який система хоче отримати в випадку 2, після перевірки на початку стану 2. Відсутність операції в стані 1 може перешкодити переходу системи в наступний стан. Це можна представити у вигляді малюнка 6.

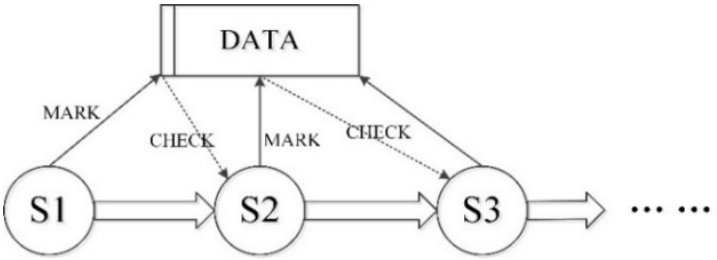


Рис. 2.6. Обмін даними між різними компонентами.

Якщо оцінка вірна, то можливість спільного використання ресурсів (даних) перевіряється відповідним чином. Припустимо, що компонент спільного використання даних позначений *символом* D_i в стані і після кожного стану. Потім $M(D_i)$ представляє серійний номер D_i. На початку наступного стану *перевіряється значення* D_i компонента даних, і значення *дорівнює* $C(D_{i+1})$.

Система повинна задовольняти рівнянню (4).

$$M(D_i) = C(D_{i+1}) \tag{4}$$

Якщо виконується рівняння (4), то виходить, що ресурс даних працює правильно. В іншому випадку в колишньому стані щось не так або відсутня.

2.6. | Метод Аналізу На Основі Моделей

2.6.1. | Підхід до моделювання на основі AADL

Складний процес динамічної реконфігурації важко проаналізувати без моделювання. AADL-це ефективний інструмент моделювання для вбудованої системи реального часу. Динамічна реконфігурація-це процес, пов'язаний з людським оператором і автоматизацією. Діапазон аналізу досить широкий. Однак у цьому дослідженні події та умови, які змінюють процес, спрощуються деякі тригери. Об'єкт-це просто сам процес. Таким чином, тут обговорюється детальна декомпозиція і формалізоване вираження спрощеного процесу.

2.6.1.1. | Процес Динамічної Реконфігурації

Коли один або кілька збоїв відбуваються в модулі IMA, диспетчер працездатності виявляє збій і повідомляє диспетчеру збоїв, щоб обробити його. Диспетчер несправностей може обробляти серію збоїв при всіх типах механізмів. Потім диспетчер несправностей визначає тип збою, щоб зробити дії для його усунення, наприклад, закрити динамічну реконфігурацію або повідомити про це менеджерові верхнього рівня.

Якщо диспетчер несправностей не зможе усунути збій, він запустить динамічну реконфігурацію. Коли обмеження не будуть задоволені в процесі динамічної реконфігурації IMA, процес зупиниться, і система вийде з ладу. Коли починається процес динамічної реконфігурації, система зупиняє додаток з помилкою і створює резервні копії даних. Зв'язку зруйновані. Потім вибирається цільовий модуль реконфігурації, виходячи із функціональних і нефункціональних вимог, таких як мінімізація витрат на зв'язок. Наступним

кроком є створення нового розділу в іншому модулі програми. Згодом виконується перезавантаження програми та відновлення з'єднання. У цьому дослідженні процес динамічної реконфігурації IMA описується у вигляді послідовних блок-схем, заснованих на припущенні, що ймовірність виникнення кожного етапу процесу реконфігурації становить 100%. Ми прагнемо змоделювати весь процес динамічної реконфігурації та проаналізувати, які кроки і незадоволені обмеження призводять до зупинки процесу реконфігурації.

Типовий процес динамічної реконфігурації представлений на рисунку 7. Стрілки вказують, що повідомлення надсилаються під час процесу. Прямокутники являють собою важливі дії, які відбулися. Порівняно з процесом реконфігурації, згаданим в іншому дослідженні, в якому завжди присутні резервні модулі, динамічна реконфігурація, обговорювана в цьому дослідженні, відноситься до системи без запасних модулів, особливо коли реконфігурація у разі надмірності не розроблена або використовується в системі, коли починається динамічна реконфігурація.

У наступній частині описується метод моделювання цього процесу, за яким слід інтерпретація методу аналізу на основі моделі з цими логічними обмеженнями.

2.6.1.2. Моделювання процесу динамічної реконфігурації

AADL представлений вище для опису системи IMA. Режим системи може бути пов'язаний з логічними конфігураціями. Переходи між режимами увазі, що стан конфігурації змінюється від одного до іншого [165]. Система або компонент мають різні статичні структури і властивості в різних режимах. Властивість може описувати планування завдань, характеристики у реальному часі, зв'язок, пам'ять і т. Д. Потім режими на системному рівні становлять зміст конфігурації системи. Система має свої власні модулі, розділи, процесори і шину зв'язку в кожному режимі. Таким чином, статична структура системи в одному режимі побудована за додатком ARINC 653 в AADL.

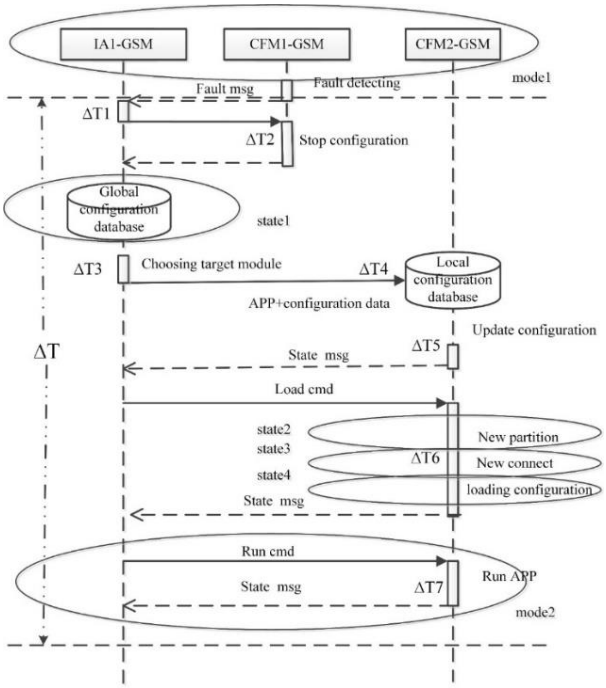


Рис. 2.7. Типовий процес динамічної реконфігурації.

На рис. 7 показаний ряд проміжних станів між двома режимами. Підстанції і переходи між режимами можуть бути описані в додатку "Поведінка". Початковий стан в додатку відповідає попереднього режиму, в той час як останнє повне стан є останнім режимом. Інші підстанції можуть описувати певний стан системи, коли завершується перехід під час динамічної реконфігурації. Перехід дії в програмі можуть описувати перехід режимів. У поведінковому додатку можна описати припинення і перезапуск програм, створення і знищення процесів та їх потоків, створення і видалення інтерфейсів зв'язку, побудова і розрив з'єднань передачі і віртуальних каналів, а також відправку і прийом повідомлень з іншими компонентами GSM.

Додаток моделі помилок [166,167] являє умови запуску при перенастроюванні, викликані збоями. Тип моделі помилок може оголошувати стану помилок, події помилок та поширення помилок. Реалізації моделі помилок оголошують переходи між станами помилок. Переходи оголошуються для подання помилок, які поширюються з компонента, на основі поточного стану помилки цього компонента. Властивість помилки захисного події може вказувати, що певні шаблони станів та поширення помилок виявляються і викликають основна подія AADL, наприклад, запуск переходу в режим.

Метод моделювання, запропонований у даному дослідженні, представлений на рисунку 8. Зміна режиму означає, що сталася динамічна реконфігурація. Більш детальна інформація та проміжні стани між режимами описані в додатку "Поведінка". Умова запуску переходу в режим оголошено в додатку моделі помилок.

Властивості додаються в модель, особливо в додаток "Поведінка" для подальшого аналізу. В якості основи аналізу з кількома обмеженнями такі елементи, як властивості часу, обсяг пам'яті і стану даних, є важливими елементами системи.

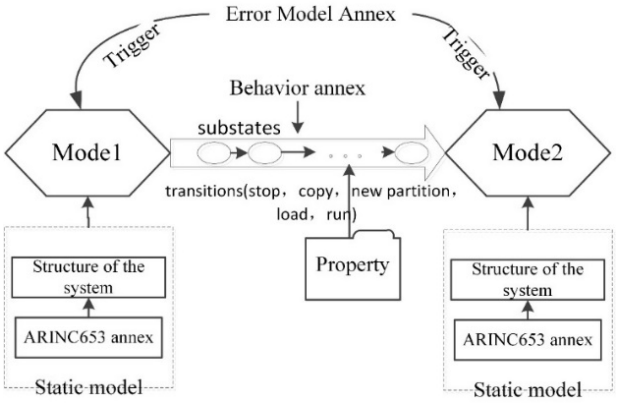


Рис. 2.8. Підхід до моделювання динамічної реконфігурації IMA.

2.6.2. | Правила перетворення моделі

Модель AADL динамічної реконфігурації IMA ефективна для опису структури системи і процесу складної реконфігурації. Деякий аналіз може бути проведений в інструментах для AADL, таких як середовище інструментів AADL з відкритим вихідним кодом (OSATE) [168]. Проте автоматичне моделювання і аналіз не є сильними сторонами AADL, але підходять для мережі Петрі. Між тим, існують також недоліки в моделюванні вбудованих систем для мереж Петрі. У цьому дослідженні режими в моделі AADL можуть бути представлені місцями в CPN. Активні режими в AADL можуть бути представлені місцем з певним кольоровим маркером в CPN. Переходи режимів в моделі AADL можуть бути перетворені в перехід токенів в CPN. Часові властивості переходів станів в моделі AADL відповідають часовим мітках токенів в дугах в CPN. Ресурси, такі як пам'ять і дані в моделі AADL, які спільно використовуються у системі, можуть бути представлені токенами в місці CPN. Нарешті, обмеження на пам'ять і час в моделі AADL можуть бути перетворені в захисні функції мережі Петрі. Таким чином, модель AADL може бути переведена в CPN цілком, як показано на рисунку 9.

AADL Component	Petri Net
Active Mode	Place With Token
Non-active Mode	Place Without Token
Transition	Transition
Time Property	Time Stamp
Memory Data Components	Tokens In Places
Constraints	Guard Function

Рис. 2.9. Взаємозв'язок перетворення моделі між мовою аналізу архітектури та дизайну (AADL) і кольоровий мережею Петрі colored Petri net (CPN).

2.6.3. | Аналіз моделювання з допомогою CPN

Для наочної демонстрації нашої моделі CPN на основі методу аналізу був використаний приклад CPN. Інтуїтивно зрозуміло, що на рисунку 10 показано приклад простого CPN. Інструменти CPN [169] використовуються для створення мереж Петрі. Приклад мережі Петрі має шість місць. Три з цих місць представляють стану системи — запуск, A і B. Місце, відоме як збій, виявляє запускає подія. Місце, позначене D, являє компонент даних. Місце з ім'ям M являє ресурс пам'яті. Дуга з'єднує місце і перехід.

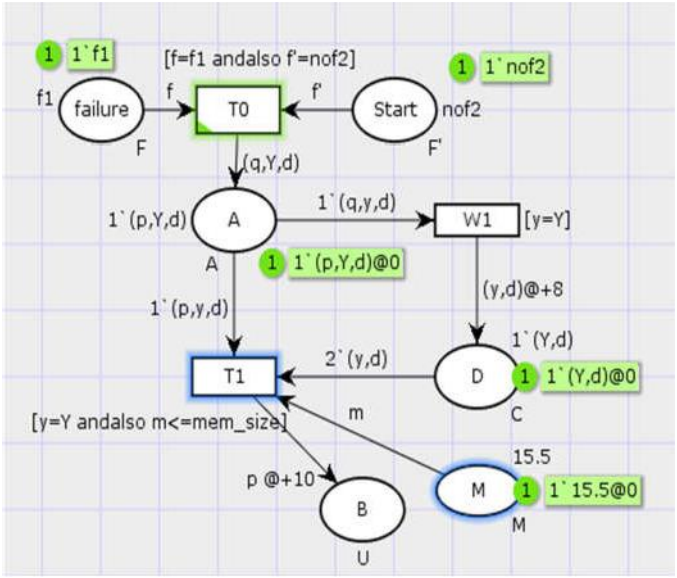


Рис. 2.10. Приклад CPN.

Різні набори кольорів використовуються для представлення запису даних, обсягу пам'яті і активності стану. Позначка часу використовується для представлення часу, витраченого на перехід.

Декларація ця сітка виглядає наступним чином: : closet U = with p|q timed; closet W = with Y|N; closet D = with d; closet C = product W × D timed; closet F = with f1|nof1; closet S = with f2|nof2; var a:A; closet M = real with 1.0..30.0 timed; var x: U; var y:W; var m:M; var f':F'; var f:F; val mem_size = 10.0. У наборі кольорів U, p використовується для позначення того, чи активовано місце, q означає, що місце A починає запис в компонент даних D, Y або N означає, чи може місце A записувати дані. M являє пам'ять, і T1 використовується тільки в тому випадку, якщо M знаходиться в діапазоні обмежень. Набір кольорів F і F' показує, відбулася подія збою і чи вплинуло це на початковий стан системи.

Після моделювання за допомогою мережі Петрі можна було отримати кілька типів результатів, заснованих на умовах обмеження.

1. Якщо всі обмеження виконані, мережа моделюється до останнього місця і зупиняється.
2. Стан системи для динамічної реконфігурації потребує перевірки. Перехід T0 може бути запущений тільки в тому випадку, якщо функція захисту [f = f1 і f' = nof2] задовільна. Це означає, що подія збою відбулося для запуску динамічної реконфігурації і не поширилося на інші модулі системи.
3. Слід визначити, чи виконуються обмеження переходу стану системи в реальному часі, чи ні. Кожен крок у процесі моделювання фіксується відміткою часу в переході. Коли один крок завершено, витрачений час порівнюється з обмеженнями в реальному часі. Результат може підказати нам, дотримано чи обмеження в реальному часі. В цьому обмеженні є слабке місце, оскільки моделювання повинно виконуватися вручну крок за кроком.
4. Обмеження пам'яті стану системи не відповідають вимогам. Функція захисту T1 [y = Y і m ≤ mem_size] встановлюється для визначення того, чи менше обсяг пам'яті, займаний в стані (набір кольорів M), ніж обмеження на розмір пам'яті. У цій мережі обмеження на розмір пам'яті становить 10 M, в той час як для цього потрібно 15,5 M. Потім мережеве моделювання припиняється в точці T1, оскільки воно не може бути запущено без виконання функції захисту.
5. Виконано обмеження на можливість спільного використання даних ресурсів. Якщо токен з місця A на перехід W1 не відповідає функції захисту, моделювання зупиняється. У цій мережі набір кольорів Y в кольорі W відправляється в W1. Функція [y = Y] виконана, і моделювання триває. Це

означає, що в компоненті даних додається позначка на вимогу (місце D). Наступне стан може бути викликане цією міткою.

тематичне дослідження

Тут, у випадку системи IMA, інтегрований ряд функціональних модулів, включаючи навігацію, дисплей, зв'язок та інтегровані радіочастотні датчики (IRFS). Навігаційний модуль визначає місцеположення літака і направляє літак в маршрутизаторі визначення. Модуль для відображення кабіни літака забезпечує людино–машинний інтерфейс для пілота. Модуль зв'язку відповідає за зв'язок між повітряним судном та наземним підрозділом. IRFS об'єднує всі радіочастотні датчики в літаку для відправки і прийому сигналів у всіх частотних діапазонах.

2.7.1. | Моделювання, Перетворення й Моделювання

Для спрощення система IMA з чотирма модулями моделюється з використанням AADL у цьому розділі. Ми позначаємо кожен модуль першою літерою його назви—навігаційний модуль (N), модуль відображення (D), зв'язок (C) і IRFS (I). В кожному модулі є кілька розділів у відповідності з їх функціями. Додаток запускається в розділі. Процес відноситься до додатка тут. Більш того, передбачається, що в модулі N і модуля D. є один розділ. У модулі I. налаштовані три розділу. Два інших розділу знаходяться в модулі C. Додаток на кожному розділі взаємодіє з GSM для визначення роботи з'єднань і додатків.

По-перше, стан конфігурації системи може бути описане за допомогою AADL. Структура логічної конфігурації потребує додатку ARINC 653 в AADL. Об'єкти ARINC 653, виготовлені з використанням архітектури системи, що відповідають компонентам AADL, представленим у розділі II. Модель системи IMA представлена в графічному вигляді на основі AADL, як показано на рисунку 11.

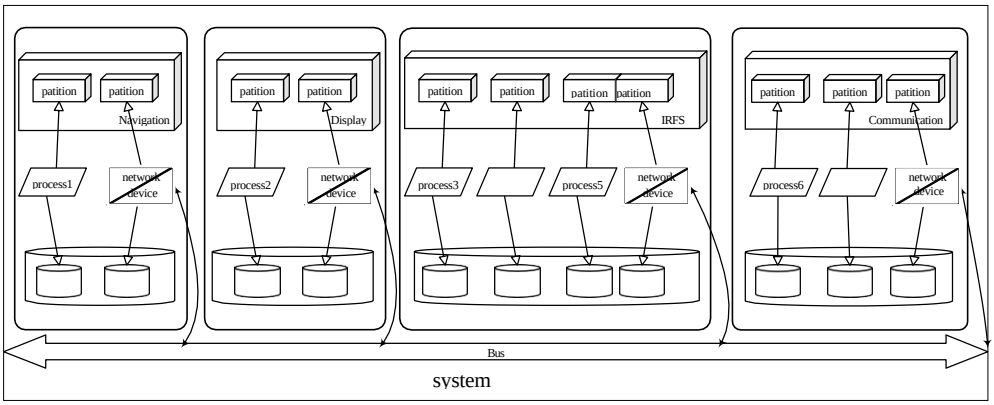


Рис. 2.11. Модель структури системи в одному режимі на основі AADL.

Коли модуль N виходить з ладу, GSM виявляє збій і запускає управління збоями. У цьому випадку збій призводить до виходу з ладу процесу 1 і реконфігурації системи. Таким чином, запускається динамічна реконфігурація.

1) Після резервного копіювання даних для процесу 1 процес завершується 1, і з'єднання процесу 1 в модулі N знищуються.

2) Система вибирає відповідний модуль для створення нового розділу для запуску процесу 1. Стратегія вибору цільового модуля тут не представлена. Цільовим модулем в даному випадку є модуль D.

3) У цільовому модулі створюється новий розділ D. Крім того, налаштовуються нові канали і з'єднання. Процес 1 перезавантажується і перезапускається у новому розділі модулі D.

Процес представлений на рисунку 12.

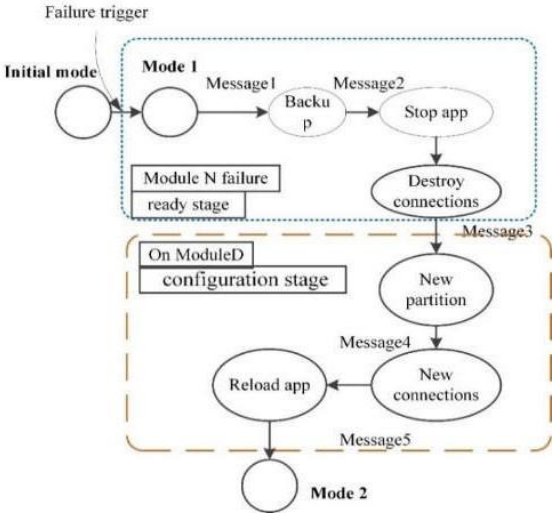


Рис. 2.12. Випадок декомпозиції динамічної реконфігурації.

У цьому випадку режим використовується для представлення стану конфігурації системи під час динамічної реконфігурації. Початковий режим-це робочий стан без збоїв системи. При виникненні збою система переходить в

режим 1, і модуль N виходить з ладу. Після перенастроювання система переходить в режим 2. Таким чином, система працює в новій конфігурації без збоїв. Режим переходу системи показаний на рисунку 13.

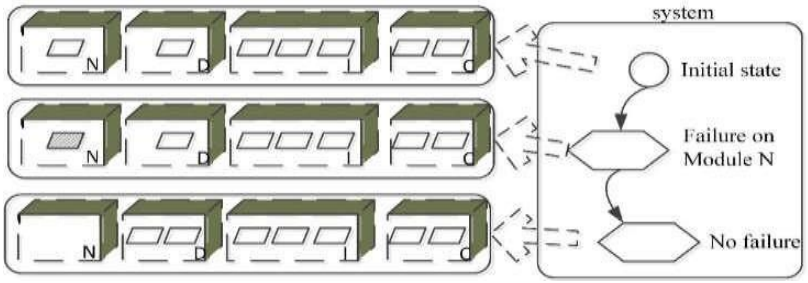


Рис. 2.13. Перехід в режим динамічної реконфігурації IMA.

Додаток "Поведінка", яке застосовується між двома режимами, являє переходи режимів з низкою дій, тригерів і умов, таких як резервне копіювання даних для процесу 1 і створення нового розділу в модулі D, як показано на рисунку 12. Певні властивості додаються в додаток "Поведінка". Внесена модифікація для визначення підстанцій між режимом 1 і 2 режимом. Оголошення набору подсостояний - "складене стан між режимом 1 і 2 режимом compstate". Потім кожен стан у стані компіляції визначається як "режим 1: початковий стан, резервне копіювання: повний стан, Stop_Process: повне стан" і так далі. Перехід між резервним копіюванням стану і Stop_Process представлений в інструкції " Резервне копіювання-[data_backup] - > Stop_Process; {RealtimeProperty : : Час процесу = >> 10.0; обсяг пам'яті ≥ 12 МБ;}". Властивості часу та пам'яті в цьому переході додаються до цього переходу.

Додаток "Модель помилок" використовується для представлення умов запуску, викликаних збоями. В цьому випадку відбувається подія збою, яке призводить до динамічної реконфігурації системи. У додатку описано, що стан системи змінюється від початкового без помилок до стану помилки. Оператор є "error_free-[error_occurred] - > error_state". Потім переходи в моделі помилок запускають систему для запуску реконфігурації. вираження може бути в вигляді 'Error1_trigger ≥ self[detected_state] застосовується до mode_transition_event.'

На основі правил, визначених у розділі 4, модель динамічної реконфігурації AADL перетворюється в CPN. Режими і стани програми "Поведінка" перетворюються на місця в CPN. Переходи режимів і переходи додатків поведінки перетворюються на переходи в CPN. Інші ресурси, такі як пам'ять і дані, представлені кольоровим набором маркерів на місцях. Умова запуску і обмеження додаються в CPN в якості захисних функцій для переходу.

В цьому випадку система створює новий розділ у модулі D, який визначається як підстанція в додатку поведінку. Перш ніж стан буде активовано, токени, що відносяться до пам'яті, і для передачі повідомлення про те, що попередній стан завершено, повинні бути відправлені в перехід. Більш того, функція охорони при переході повинна бути виконана. Наприклад, розмір пам'яті має відповідати нерівності, згідно з яким необхідний розмір повинен бути не менше розміру, як у рівнянні (3). Оголошення для моделі CPN наведено на рисунку 14. Модель CPN представлена на рисунку 15.

Буква " s "в наборі кольрів" S " - це маркер, який означає перехід системи в стан. Набір кольорів F1 вказує, чи відбулася подія збою, а F2 вказує, чи вплинуло початковий стан системи. Як тільки моделювання ініційовано, мережа автоматично запускається крок за кроком, щоб відправити кольорові жетони, коли функції захисту будуть виконані.

```
Declarations:
colset S= with s|w;
colset F1= with failure1|nofailure1; colset F2= with failure2|nofailure2;
colset Initial= product S*F2 ;
colset D= with d; colset B= product S*D timed;
colset M= real with 1.0..100.0 timed;
colset N= bool with (no,yes);
var f1:F1; var f2:F2;
var I: Initial; var p:S;
var m:M; val mem_size = 200.0;
var b:B; var n:N;
```

Рис. 2.14. Заява по справі CPN.

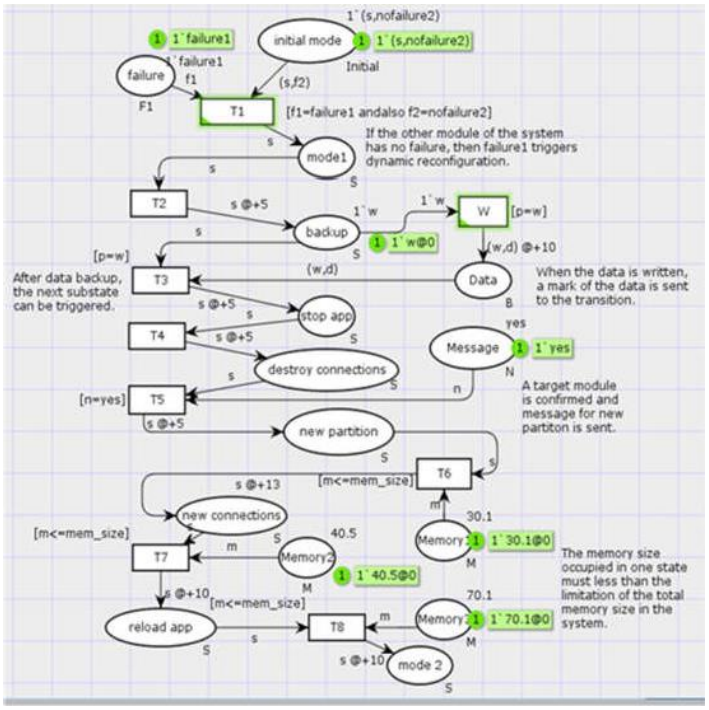


Рис. 15. Модель CPN для динамічної реконфігурації.

2.7.2. | Результати Моделювання

Результати наведені в таблиці 2 після багаторазового проведення моделювання в різних умовах. Попередньою умовою для кожного результату було те, щоб інше обмеження, встановлене в цій мережі, задовольняло вимогам для завершення моделювання, за винятком умови, вказаного в таблиці 2.

Таблиця2. 2. Результати моделювання.

No	Вихідні умови	Результати моделювання	Аналіз
1	Val mem_size = 200.0 (рис.16a)	Моделювання зоканчилось нормально	Обмеження пам'яті для стану системи виконані.
2	Val mem_size = 70.0 (рис.16b)	T8 не може бути запущений, імітація зупинена	Обмеження пам'яті для стану системи не відповідають вимогам.
3	Витрати часу під час моделювання, що виходять за рамки обмежень реального часу	Позначка часу "@ +48" показує час виконання 48 понад 30 обмеження.	Обмеження в реальному часі для переходу стану системи не виконуються.
4	1`(s, failure2) - > initial mode and 1'failure1 - > failure (Figure 16c)	Моделювання не запущено, T1 не запускається	Система знаходиться в стані поширення несправностей і не підходить для реконфігурації.
5	Немає ніякої відправки "w' в місце з ім'ям Data (Рис. 16d)	Значення функції захисту [p = w] є помилковим. W не запущений. Зупинка моделювання	Відмітка вимоги не записується в компоненти даних, тому наступного станом не вдалося надати спільний доступ до даних

1. Умова 1: Всі обмеження виконані. Модель системи, отримана при проведенні моделювання протягом 58 мс, показана на рисунку 16a. Ця модель дуже схожа на оригінальну модель системи (рис. 15).

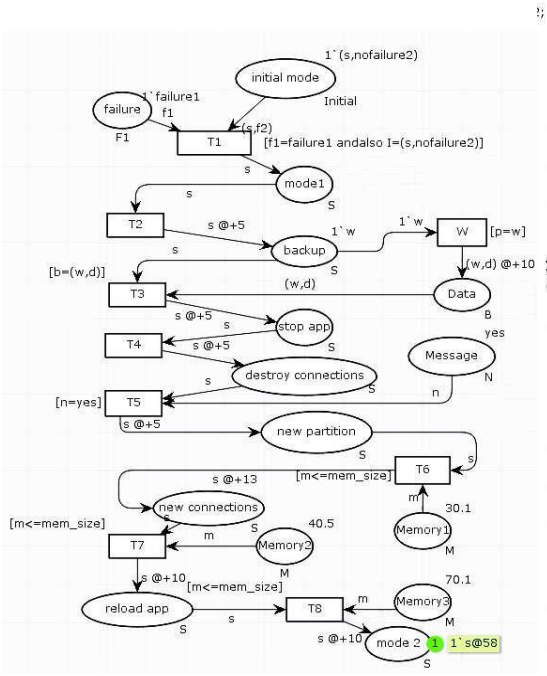
2. Умова 2: Обмеження обсягу пам'яті не виконано. Моделювання припиниться, коли воно буде виконуватися протягом 48 мс, оскільки функція захисту [m ≤ mem_size] не задоволена. Верхня межа розміру системної пам'яті mem_size складає всього 70 M, але стан повинен займати об'єм пам'яті 70,1 M, тому моделювання закінчується. Результат показаний на рисунку 16b.

3. Умова 3: Обмеження в реальному часі для переходу в стан системи не виконується. Порівнюючи спожите час і вимоги в реальному часі, можна показати, задоволено обмеження в реальному часі.

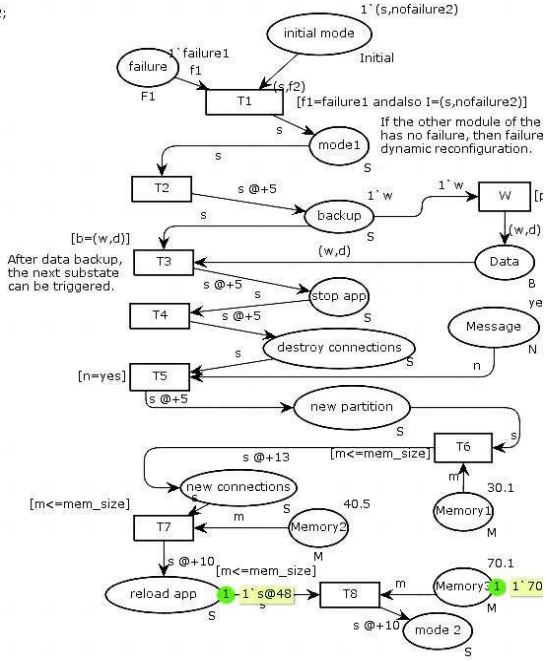
4. Умова 4: Обмеження стану системи для динамічної реконфігурації не виконано. Коли система переходить до переходу T1, про це можна судити по функції захисту [f1 = 1 збій, а також I = (s, nofailure2)]. Якщо поширення несправностей відбувається до реконфігурації системи і зачіпаються інші

модулі, схема реконфігурації не може бути прийнята. Процес реконфігурації зупиняється і не може бути виконаний, як показано на рисунку 16c.

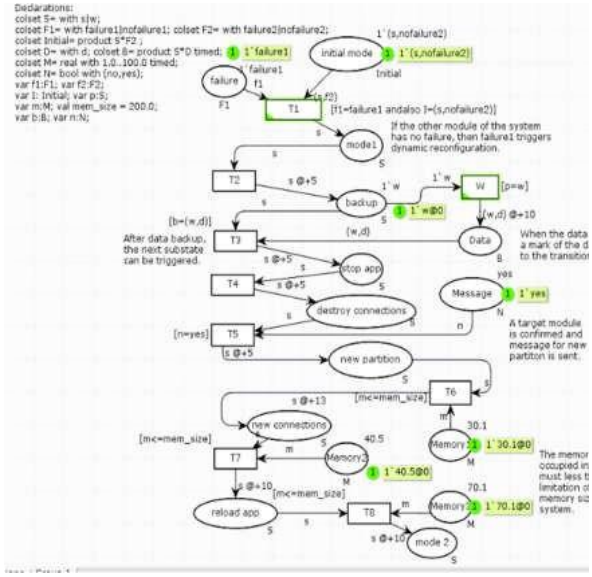
5. Умова 5: Обмеження можливостей для спільного використання ресурсів даних не виконано. Коли система виконує операцію із загальним ресурсом даних, якщо резервна копія прямого стану не може виконати запис в компонент даних, то перевірка компонента даних і останнього стану не запускається. Потім процес зупиняється з кроком 15 мс, як показано на рисунку 16d.



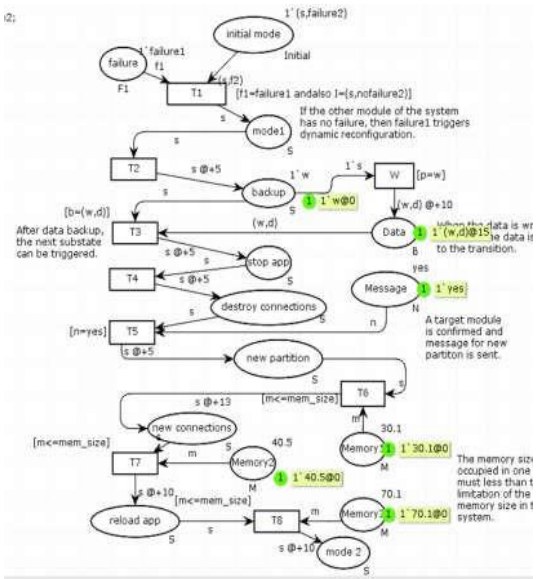
(a)



(b)



(c)



(d)

Рис. 16. (a) Модель системи за умови, що всі обмеження виконані. (b) Модель системи за умови, що обмеження об'єму пам'яті не виконується. (c) модель системи за умови, що обмеження стану системи для динамічної реконфігурації не виконано. (d) модель системи за умови, що не виконується обмеження можливостей для спільного використання даних ресурсів.

2.8. | Висновки

Існуючі дослідження динамічної реконфігурації рідко фокусувалися на аналізі технологічних ризиків. У цьому дослідженні був запропонований новий метод аналізу на основі моделей з безліччю обмежень для процесу динамічної реконфігурації IMA, що є основним внеском в це дослідження. Метод аналізу проводиться в три етапи—моделювання, перехід до моделі і моделювання. Для моделювання процесу динамічної реконфігурації був застосований новий метод моделювання, заснований на AADL, і перехід перетворює модель в CPN для моделювання. Кілька обмежень концентруються на декількох різних аспектах для роботи з моделюванням. Цей підхід був продемонстрований з використанням чотиримодульною системи IMA. Результати тематичного дослідження показали ефективність цього методу. У цьому дослідженні був запропонований метод аналізу на основі моделі з безліччю обмежень для процесу динамічної реконфігурації IMA, який допомагає вирішити проблеми безпеки при динамічної реконфігурації IMA, викликані високою інтеграцією і складністю.

У майбутній роботі необхідно доповнити додаткові обмеження для більш всебічного аналізу. Якщо система стане більш складною і число станів динамічної реконфігурації зросте, то аналітичній роботі буде складніше. Висока робоча навантаження виникає аналізу. Таким чином, необхідно розробити інструмент для автоматичного додавання обмежуючих умов. Потрібна перевірка цього підходу на більш складних прикладах та інженерних практиках.

Розділ 3 «Аналіз безпеки на основі AADL з використанням формальних методів, застосовуваних до цифрових систем повітряних суден»

3.1. | Введення

Аналіз системної безпеки має вирішальне значення в життєвому циклі розробки критично важливих систем для забезпечення належної безпеки, а також для демонстрації відповідності застосовним стандартам. Необхідною умовою для будь-якого аналізу безпеки є глибоке розуміння архітектури системи і поведінки її компонентів; інженери з безпеки використовують це розуміння для вивчення поведінки системи для забезпечення безпечної експлуатації, оцінки впливу збоїв на загальні цілі безпеки і побудови системи аналізу з виявлення артефактів. Розвиток адекватного розуміння, особливо для програмних компонентів, є складною і трудомісткою завданням. Відсутність точних моделей архітектури системи і режимів її відмов часто змушує аналітиків безпеки докладати значні зусилля для збору архітектурних відомостей про поведінку системи з декількох джерел. Всебічна ідентифікація всіх небезпечних взаємодій у все більш складних системах з інтенсивним програмним забезпеченням також є складним завданням. Використання розробки систем на основі моделей в критичних системах і використання загальної формальної моделі, що розділяється розробкою систем і аналізом безпеки, має великі перспективи. Підхід, заснований на моделі, може допомогти усунути двозначність, підвищити узгодженість артефактів, точність аналізу і звести до мінімуму ітерації аналізу проектування/безпеки [170-174–174].

Підходи до аналізу/оцінки безпеки на основі моделей (MBSA) були розроблені для різних мов моделювання, включаючи SysML [175-177], Мова аналізу і проектування архітектури (AADL) [178,179], SLIM [180] і Simulink [181,182]. Кожна мова має цільову область застосування і містить різні рівні формального підходу. AADL, стандартний мова моделювання Суспільства автомобільної техніки (SAE) для проектування систем на основі моделей (MBSE) [178], забезпечує більш суворе опис системи та семантику часу виконання і добре підходить для моделювання вбудованих систем у реальному часі. AADL має досить чітко певну семантику, що дозволяє застосовувати

підходи до перевірки формальних моделей, тому для нашого підходу вибрати мову.

Ось чому ця мова була обрана для нашого підходу. Підходи, використовувані в інструментах MBSA, істотно розрізняються. Питання полягає в тому, поширюються помилки явно або за допомогою поведінкового моделювання. Такі інструменти, як Додаток до моделі помилок AADL версії 2 (EMV2) [179], HiP-HOPs для EAST-ADL [183] та Ansys Medini [184], є явними підходами до поширення. Враховуючи безліч можливих помилок, ці взаємозв'язки поширення потребують значних зусиль користувача для розуміння і визначення. Крім того, відсутність поширення призводить до неправильного аналізу.

Ще одним важливим міркуванням є те, що позначення для аналізу безпеки існуюча модель системи доповнено інформацією про аналіз безпеки. Прикладами таких інструментів аналізу безпеки є SmartIFlow [172], Мова аналізу і моделювання безпеки (SAML) [174] і AltaRica [185,186]. При розробці моделі системи окремо від моделі безпеки це вимагає тісного зв'язку між групами розробників на кожній ітерації розробки моделі. Важливі зміни в моделі системи автоматично не враховуються в моделі безпеки.

В даній роботі буде описано програму з безпеки для мови системної інженерії AADL. Додаток з безпеки дозволяє аналітику моделювати режими відмов компонентів, а потім 'сплітати' ці режими відмов разом з оригінальними моделями, розробленими в рамках MBSE. Потім аналітик безпеки може використовувати об'єднані поведінкові моделі для поширення помилок по системі, щоб дослідити їх вплив на вимоги безпеки. Цей документ є продовженням більш короткого документа конференції, в якому представлено програму з безпеки [187].

Нашу роботу можна розглядати як продовження роботи, проведеної Джоші та ін. де вони досліджували поведінковий підхід аналізу безпеки на основі моделей, визначений за моделями Simulink/потоку станів [170,181,181,188]. Наша поточна робота розширює і узагальнює цю роботу і надає нові можливості моделювання і аналізу, раніше недоступні. Він також переміщує аналіз з мови реалізації компонентів (Simulink) на мову архітектури вбудованих систем реального часу (AADL). Додаток безпеки дозволяє моделювати як зорові, так і явне поширення помилок, підтримує верифікацію складу і забезпечує вивчення номінального поведінки системи, а також поведінки системи в умовах збою. Наша робота також пов'язана з існуючими підходами до аналізу безпеки, зокрема, Додатком до помилок AADL (EMV2) [179],

COMPASS [180] і AltaRica [185,186]. Наш підхід істотно відрізняється від попередньої роботи в тому, що на відміну від EVM2 ми використовуємо поведінкову модель для прихованого поширення помилки, ми надаємо композиційний аналіз можливостей не доступний на компас, і крім того, безпека додаток повністю інтегровано в модель розвитку процесу і середовища, на відміну від ізольованого мови, такі як AltaRica.

Цілі і завдання даного дослідження полягають у наступному.

• підтримка загальної моделі, що відображає поточний стан проектування системи протягом всього життєвого циклу розробки, що дозволяє всім учасникам процесу 4754 A "Рекомендовані аерокосмічні практики" (ARP) мати можливість обмінюватися інформацією та аналізувати проект системи;

• інтегрувати аналіз поведінкових помилок в мову системного моделювання з чітко визначеною семантикою;

• підтримка поведінкової специфікації несправностей та їх неявного поширення (як симетричним, так і асиметричним через поведінкові відносини в моделі;

• використання формальних методів для автоматичної перевірки властивостей безпеки при наявності несправностей і одержання доказів аналізу, виконаного для досягнення цілей процесу оцінки безпеки.

3.2. | Попередні заходи

Ми використовуємо Мову архітектурного аналізу і проектування (AADL) [189] для побудови моделей системної архітектури. AADL-це міжнародний стандарт SAE, який визначає мову і забезпечує унифицирующую основу для опису архітектури системи для 'систем з критичної продуктивністю, вбудованих систем реального часу" [178]. З моменту своєї концепції AADL застосовувався при проектуванні і створенні систем авіоніки. Замість того, щоб бути просто описовими, моделі AADL можуть бути досить конкретними, щоб підтримувати генерацію коду на рівні системи. Таким чином, результати проведених аналізів, включаючи запропонований тут новий аналіз безпеки, відповідають системі, яка буде побудована на основі моделі.

Модель AADL описує систему в термінах ієрархії компонентів і їх взаємозв'язків, де кожен компонент може представляти або логічну сутність (наприклад, функції прикладного програмного забезпечення, дані), або фізичну сутність (наприклад, шини, процесори, пам'ять). Модель AADL може бути

розширена мовними додатками, щоб забезпечити більш багатий набір елементів моделювання для різних потреб проектування та аналізу системи (наприклад, характеристик, пов'язаних з продуктивністю, параметрів конфігурації, динамічної поведінки). Визначення мови досить суворе, щоб підтримувати інструменти формального аналізу, що дозволяють виявляти помилки/несправності на ранній стадії.

Середа міркувань з гарантією припущення (AGREE) [190] є інструментом для формального аналізу поведінки в моделях AADL. УГОДА реалізовано у вигляді програми AADL і аннотирует компоненти AADL формальними поведінковими контрактами. Контракти кожного компонента можуть включати припущення і гарантії щодо вхідних та вихідних даних компонента відповідно, а також предикати, що описують, як стан компонента змінюється з плином часу. ЗГОДА переводить модель AADL і поведінкові контракти в Lustre [191], а потім запитує засіб перевірки моделей JKind [192] для проведення внутрішнього аналізу. Аналіз може бути виконаний композиційно згідно з ієрархією архітектури таким чином, щоб аналіз на більш високому рівні ґрунтувався на компонентах на наступному нижчому рівні. Порівняно з монолітним аналізом (тобто аналізом сплощеної моделі, що складається з усіх компонентів) композиційний підхід дозволяє масштабувати аналіз до більш великих систем [190].

3.3. | Методологія

В якості поточного приклад у цій методології представлено Колісну гальмівну систему (WBS), описану в Звіті за аерокосмічної інформації (AIR) 6110 [193]. Ця система є добре відомим прикладом, який використовувався в якості прикладу для аналізу безпеки, формальної перевірки і проектування на основі контрактів [171,182,194,195].

3.3.1. | Огляд колісної гальмівної системи

Попередня робота над додатком з безпеки була заснована на простій моделі WBS [196]. Щоб продемонструвати більш складний процес моделювання несправностей, було побудува функціонально і структурно еквівалентну версію AADL більш складною WBS, яка була захоплена в моделях NuSMV/xSAP [194]. Рис. 1 для наочності показана лише одна пара коліс і їх взаємодія з іншою частиною системи. Повна версія, змодельована в AADL, містить в загальній складності 8 коліс.

WBS складається з двох основних частин: системи управління та електромеханічної фізичної системи. Фізична система складається з резервних гідравлічних контурів (позначених зеленим і синім кольором), що йдуть від гідравлічних насосів до колісних гальм, а також клапанів, які управляють потоком гідравлічної рідини. Фізична система забезпечує гальмівне зусилля для кожного з восьми коліс літака. Всі колеса механічно загальмовані попарно. Система управління керує електронним управлінням фізичної системою. Блок керування гальмовою системою (BSCU) складається із двох каналів для резервування на випадок виявлення несправності в активному каналі. BSCU також подає команди на протиковзке гальмування і управляє режимом роботи системи за допомогою команд на селекторний клапан. Ці команди передаються компоненту клапана-селектора, який вибирає, який гідравлічний насос подає тиск, в залежності від поточного режиму роботи системи.

Входи верхнього рівня в систему включають механічні датчики педалей і потужність. Вони вважаються компонентами чорного ящика. Єдине взаємодія пілота, моделируемое в цій системі, здійснюється з допомогою команди механічного гальмування.

У моделі WBS існує три режими роботи:

• У звичайному режимі система використовує зелений гідравлічний насос і один вимірювальний клапан на кожне з восьми коліс (на рис. 1 це відповідає, наприклад, 'Вимірювальний клапан (колесо 1))'. Кожен з вимірювальних клапанів управляється за допомогою електронних команд, що надходять з активного каналу BSCU. Ці сигнали забезпечують команди гальмування і протиковзання для кожного колеса. Команда на гальмування визначається датчиком на педалі, а команда на протиковзання визначається датчиками колеса і виявленням заносу.

• В альтернативному режимі система використовує синій гідравлічний насос, чотири вимірювальних клапани (по одному на колісну пару, як показано на рис. 1: 'Вимірювальний клапан (пара)') і чотири запірні клапани протиковзання (по одному на колісну пару). Вимірювальні клапани механічно управляються через пілотну педаль, відповідну кожної колісної пари. Якщо селектор виявляє відсутність тиску в зеленій ланцюга, він переключається на синю ланцюг. В якості альтернативи, якщо BSCU виявляє несправність у звичайному (зеленому) режимі роботи, BSCU також може відключити зелений насос і примусово перевести перемикач в інший (синій) режим роботи.

• Аварійний режим спрацьовує при відмові синього гідравлічного насоса. Компонент акумулятора має запас гідравлічної рідини під тиском і буде подавати її в синій контур в аварійному режимі.

Модель архітектури WBS в AADL містить 30 різних типів компонентів, 169 примірників компонентів і глибину моделі 5 ієрархічних рівнів.

3.3.2. | Огляд методології

Ми пропонуємо заснований на моделі процес оцінки безпеки, підкріплений формальними методами, щоб допомогти інженерам з безпеки з раннім виявленням проблем проектування. Цей процес використовує єдину уніфіковану модель для підтримки проектування системи, так і аналізу безпеки. Він заснований на наступних кроках, як показано на рис. 2 і викладено нижче.

1. Системні інженери збирають критичну інформацію в загальній моделі AADL/AGREE: архітектура апаратного і програмного забезпечення високого рівня, номінальна поведінку на рівні компонентів і вимоги безпеки на рівні системи.
2. Системні інженери використовують засіб перевірки внутрішньої моделі, щоб переконатися, що номінальна модель відповідає вимогам.
3. Інженери по техніці безпеки використовують Додаток з техніки безпеки для доповнення номінальної моделі режимами відмови компонентів. Крім того, інженери по техніці безпеки вказують гіпотезу несправності для аналізу, яка відповідає тому, скільки одночасних несправностей система повинна бути в змозі витримати.
4. Інженери по техніці безпеки використовують засіб перевірки базової моделі для аналізу того, чи задовольняє модель вимогам безпеки і цілям відмовостійкості при наявності несправностей. Якщо конструкція моделі не допускає вказаної кількості несправностей (або порогу вірогідності виникнення несправностей), то інструмент створює контрприклади, що призводять до порушення вимог безпеки при наявності несправностей, а також всі мінімальні набори комбінацій несправностей, які можуть призвести до порушення вимог безпеки.
5. Інженери по техніці безпеки, вивчають результати, щоб оцінити правильність комбінацій несправностей і рівень відмовостійкості конструкції системи. Якщо зміна дизайну гарантовано, модель буде оновлено з урахуванням останніх змін в дизайні, і наведений вище процес повториться.

У решти цього розділу ми опишемо, як реалізується програма з безпеки, а потім опишемо, як описані вище кроки можуть бути реалізовані за допомогою програми з безпеки для AADL.

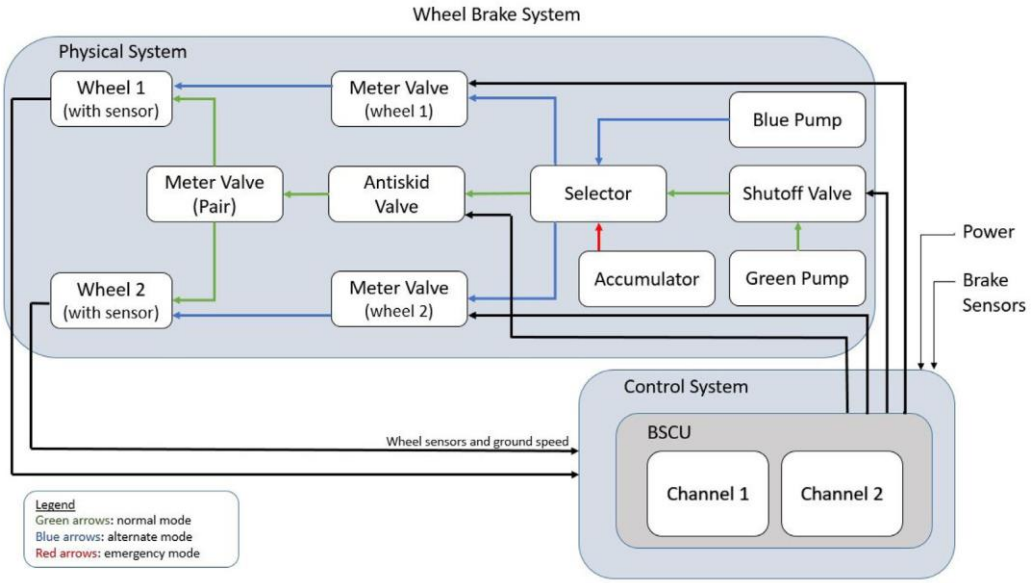


Рис. 3.1. Двоколісна схема Колісної гальмівної системи.

3.3.3. | Огляд впровадження

Додаток з безпеки написано на Java в якості плагіна для набору інструментів AADL з відкритим вихідним кодом (OSATE), який побудований на Eclipse. Він не розроблений як автономне розширення мови, але працює з поведінковими контрактами, зазначеними у додатку AGREE AADL [190]. Архітектура програми з безпеки показана на рис. 3.

Контракти на УЗГОДЖЕННЯ використовуються для визначення номінального поведінки компонентів системи в якості гарантій, які зберігаються, коли виконуються припущення про значення середовища компонента. Коли модель AADL аннотируется договорами про ЗГОДУ і модель несправностей створюється з використанням програми з безпеки, модель перетворюється через УГОДУ модель Блиску [191], що містить поведінкові розширення, визначені в договорах про ЗГОДУ для кожного компонента системи.

контракти для кожного компонента системи. При виконанні аналізу несправностей додаток з безпеки розширює контракти на УЗГОДЖЕННЯ, дозволяючи збоїв змінювати поведінку входів і виходів компонентів. Приклад частини початкового вузла УЗГОДЖЕННЯ і його розширеного контракту показаний на рис. 4. В лівій колонці малюнка показано номінальне визначення

насоса блиску з договором про погодження випуску. В правому стовпчику показано додаткові локальні змінні для помилки (поля 1 і 2), твердження, що зв'язує значення помилки з номінальним значенням поля 3 і 4), та визначення вузла помилки (поле 5). Після додавання інформації про несправності модель УЗГОДЖЕННЯ (перекладена на мову потоку даних Lustre [191]) слід стандартним шляхом перекладу на засіб перевірки моделей JKind [192], засіб перевірки властивостей безпеки з нескінченним станом.

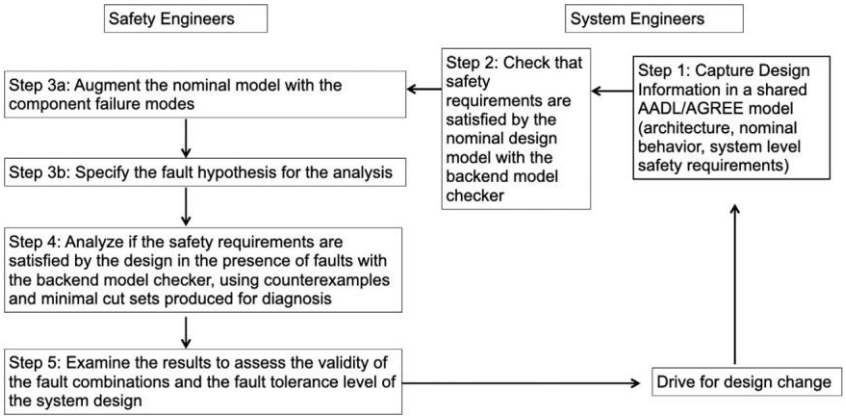


Рис. 3.2. Пропонований процес оцінки безпеки, підкріплений формальними методами.

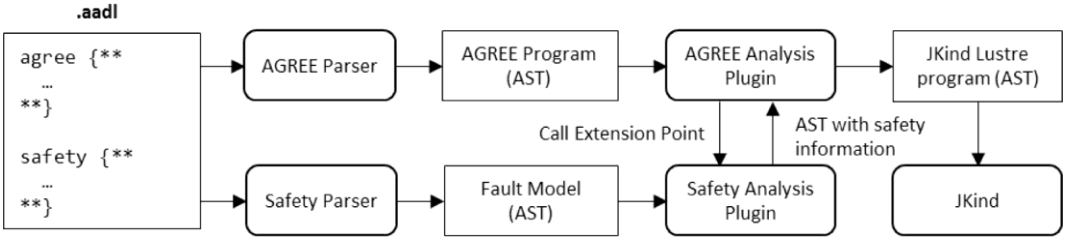


Рис. 3.3. Архітектура плагіна програми безпеки.

Формули блиску представлені в JKind у вигляді перехідної системи, а міркування виконуються з використанням k-індукції. При виконанні аналізу безпеки моделі кожна помилка визначається як літерал активації і з урахуванням обмеженого обмеження. Якщо присвоєння литералу активації істинно, це відповідає активній помилку і потенційно порушеною гарантії. Якщо це призначення порушує гарантію, то це порушення буде відображено в результатах аналізу. На системному рівні можна побачити, якщо порушена гарантія, в свою чергу, порушить властивість верхнього рівня. Отже, видно, як активні збої в компонентах кінцевого рівня порушують властивості системного рівня.

Цей підхід до аналізу дозволяє неявно поширювати порушення по всій системі. Він також допускає довільну тимчасову активацію несправностей. На збої не накладається ніяких явних обмежень, які вказують, коли може відбутися активація, що дозволяє вільної процедурі перевірки моделі активувати збої в найгірше з можливих часів. Якщо існують залежності, що стосуються активації помилок, вони обробляються за допомогою явного поширення помилок. Хоча засіб перевірки моделі може вибирати різні перестановки активації несправностей, ці перестановки несправностей з точки зору часу впливу та порядку виникнення не є частиною вихідних даних мінімального набору скорочень цього аналізу.

Основним обмеженням, що накладається на засіб перевірки моделі з точки зору активації несправностей, є твердження гіпотези несправності. Вони обмежують модель, вказуючи або кількість помилок, які можуть бути активні одночасно, або загальний допустимий поріг вірогідності. В останньому випадку кожна несправність має пов'язану ймовірність; за умови незалежності ймовірність виникнення набору несправностей не повинна бути меншою певного порогового значення.

Існує два різних типи аналізу несправностей, які можуть бути виконані на моделі несправності: перевірка на наявність несправностей або генерація мінімальних наборів скорочень. Плагін Програми безпеки перехоплює програму УЗГОДЖЕННЯ і додає інформацію про моделі несправності в залежності від того, який тип аналізу несправностей виконується.

Перевірка на наявність несправностей: Цей аналіз повертає контрприклад, якщо яке-небудь властивість гарантії або системного рівня порушено активними несправностями в системі. Контрприклад показує конкретний сценарій, по якому порушується властивість, з призначеннями кожному сигналу в моделі засобом перевірки моделі, можливо, в ступеневої послідовності. Додаток з програми "Безпека" до програми "ЗГОДА" включає інформацію про простежуваності, так що при відображенні контрприкладів користувачам візуалізуються активні несправності для кожного компонента.

Створення мінімальних наборів вирізів: Цей аналіз збирає всі мінімальні набори комбінацій несправностей, які можуть призвести до порушення властивостей. Враховуючи складну модель, часто буває корисно отримати інформацію про простежуваності, пов'язану з доказом, іншими словами, які частини моделі, необхідні для побудови докази. Алгоритм був представлений Гассабани та ін. для забезпечення ядер індуктивної валідності (IVCS) як способу визначення того, які елементи моделі необхідні для індуктивних

доказів властивостей безпеки для послідовних систем [197]. Враховуючи властивість системи безпеки, перевірка моделі може бути викликана для побудови докази цього властивості. Алгоритм генерації IVC витягує інформацію про простежуваності з процесу докази і повертає мінімальний набір елементів моделі, необхідних для доказу властивості. Більш пізні дослідження розширили цей алгоритм, щоб створити всі мінімальні ядра індуктивної валідності (All-MIVC), щоб забезпечити повне перерахування всього мінімального набору елементів моделі, необхідних для індуктивних доказів властивості безпеки [198].

У цьому підході ми використовуємо алгоритм All-MIVCs для обчислення мінімального набору елементів моделі (включаючи контракти компонентів та літерали активації помилок), необхідних для доказу властивості верхнього рівня, і перетворення їх в мінімальні набори помилок для порушення властивості верхнього рівня [199].

Щоб отримати доступ до плагіну інструменту, керівництва користувача або моделей, див. репозиторій, розташований за адресою https://github.com/loonwerks/AMASE/.

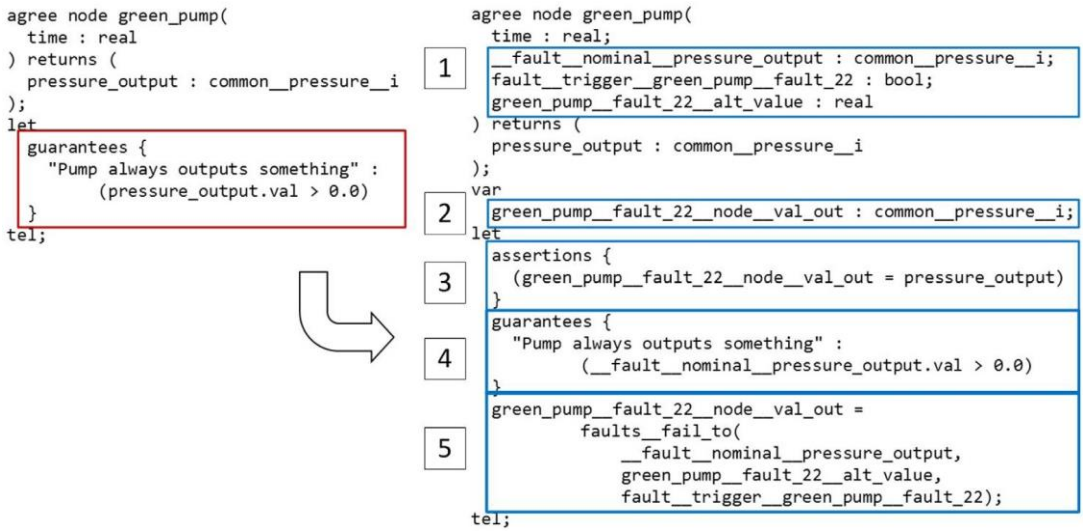


Рис. 3.4. Номінальний вузол узгодження і розширення з несправностями.

```
lemma "(S18-WBS-0325) Never inadvertent braking of wheel 1" :
        true -> (not(POWER)
            or (not HYD_PRESSURE_MAX)
            or (not W1ROLL)
            or (not SPEED)
            or (mechanical_pedal_pos_L)
            or (wheel_braking_force1 <= 0));
```

Рис. 3.5. узгодження умов характеристик вищого порядку: Ненавмисне гальмування.

Модель системи розроблена в AADL і доповнено поведінкової інформацією у тому, що ми називаємо номінальної моделлю. Номінальна, або поведінкова модель кодується з використанням програми "ЗГОДА", а поведінка засноване на описах, приведених в AIR6110. Властивості системи верхнього рівня визначаються вимогами і цілями безпеки в AIR6110. Всі контракти на подкомпоненти підтримують ці цілі безпеки системи за рахунок використання припущень про введення компонентів і гарантій на виході. Поведінкова модель WBS в додатку "ЗГОДА" включає одне припущення верхнього рівня і 11 системних властивостей верхнього рівня, при цьому 113 гарантій розподілені між підсистемами.

Прикладом властивості безпеки системи є відсутність випадкового гальмування кожного з коліс. Це засновано на умови відмови, описаному в AIR6110: Ненавмисне гальмування одним колесом під час зльоту повинен становити менше 1E-9 за зліт. Ненавмисне гальмування означає, що до колеса додається гальмівне зусилля, але пілот не натиснув на педаль гальма. Крім того, ненавмисне гальмування вимагає наявності потужності і гідравлічного тиску, літак не зупиняється, а колесо котиться (не заноситься). Властивість зазначено в ЗГОДІ таким чином, що не відбувається ненавмисного гальмування, як показано на рис. 5. (Вираз, показане на рис. 5, вірно → властивість ЗГІДНО вірно в початковому стані, а потім воно вірно тільки в тому випадку, якщо властивість виконується.)

3.3.4. | Аналіз номінальної моделі

Перед виконанням аналізу несправностей користувачі повинні спочатку переконатися, що характеристики безпеки відповідають номінальної моделі конструкції. Цей аналіз може бути виконаний монолітно або композиційно ВІДПОВІДНО. Використовуючи монолітний аналіз, контракти на всіх рівнях архітектури згладжуються і використовуються для доведення властивостей безпеки верхнього рівня системи. Композиційний аналіз, з іншого боку, буде виконувати доказ шар за шаром зверху вниз, по суті розбиваючи більш велике доказ на більш дрібні проблеми. Більш повний опис цих типів доказів і аналізів можна знайти в [190,200]

WBS має в загальній складності 13 властивостей безпеки на верхньому рівні, які підтримуються припущеннями і гарантіями підкомпонентів, наведеними в таблиці 1. Оскільки коліс 8, контракт S18-WBS-0325-wheelX повторюється 8 разів, по одному для кожного колеса. Система включає в себе як ліве (L), так і праве (R) бічне гальмування, тому S18-WBS-R/L-0322 відбувається двічі.

Поведінкова модель в цілому складається з 36 припущень і 246 допоміжних гарантій.

Результати аналізу показано на рис. 6.

Леми є специфікаціями всіх властивостей безпеки верхнього рівня моделі. Результати показують, що модель підтримує специфікації, і для кожної леми знайдено доказ. Контракти на дочірні компоненти використовуються для підтвердження дійсності властивостей безпеки.

Таблиця 3.1.

Захисні властивості WBS.
S18-WBS-R-0321 Втрата гальмування на всіх колесах під час посадки або RTO повинна становити менше 5,0×10-7 за політ.
S18-WBS-R/L-0322 Асиметрична втрата гальмування коліс (вліво/Вправо) повинна становити не менше 5,0×10-7 за політ.
S18-WBS-0323 Непреднамеренное гальмування при всіх заблокованих колесах не повинно бути менше 1,0×10-9 за зліт.
S18-WBS-0324 Непреднамеренное гальмування всіма колесами не повинно бути менше 1,0×10-9 за зліт.
S18-WBS-0325-wheelX Непреднамеренное гальмування колеса X не повинно бути менше 1,0×10-9 за зліт.

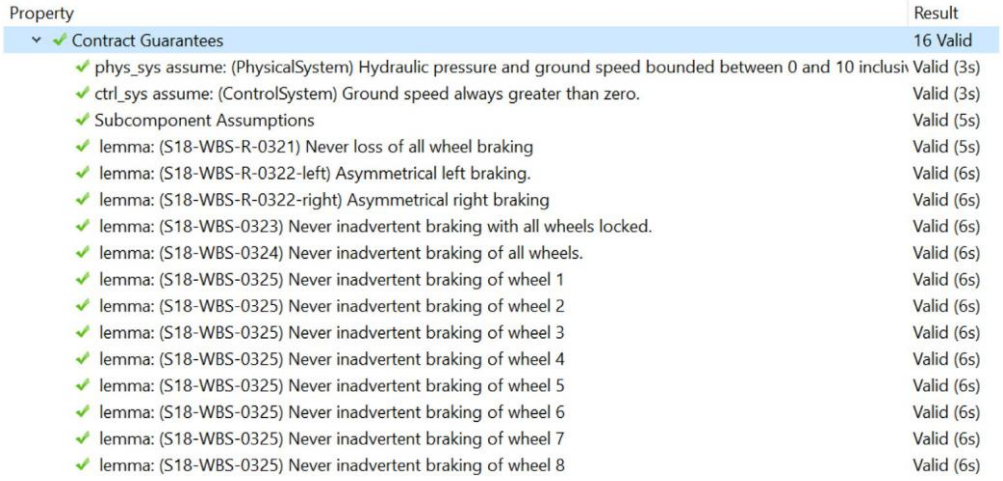


Рис. 3.6. Результати аналізу номінальної моделі для WBS.

3.3.5. | Моделювання несправностей

Використання термінів помилка, збій і несправність визначено в ARP4754 A і описано тут для зручності розуміння [201]. Помилка-це помилка, допущена в

реалізації, дизайні або вимоги. Помилка-це прояв помилки, а збій-це подія, що відбувається, коли надається служба системи відхиляється від правильної поведінки. Якщо несправність активована при правильних обставин, ця несправність може призвести до збою. Термінологія, використовувана в EMV2, трохи відрізняється для помилки: помилка-це пошкоджене стан, викликаний помилкою. Помилка поширюється по системі і може проявлятися як збій. У цьому звіті ми використовуємо термінологію ARP4754 A з доданим визначенням поширення помилок, використовуваним в EMV2. Помилка-це помилка, допущена в дизайні або коді, а поширення помилки-це поширення пошкодженого стану, викликаного активної помилкою.

Додаток "Безпека" використовується для додавання потенційних помилок в модель компонентів. В модель екземпляра компоненту AADL додається додаток, що містить визначення несправностей для даного компонента. Гнучкість визначень несправностей дозволяє користувачеві визначати безліч типів вузлів несправності, використовуючи синтаксис вузла УЗГОДЖЕННЯ. Була написана бібліотека загальних вузлів несправностей, яка доступна в репозиторії project GitHub [202]. Приклади таких несправностей включають застрявання клапанів відкритими або закритими, невизначеність вихідних даних програмного компонента або відключення живлення. Коли для аналізу несправностей потрібні більш складні визначення несправностей, ці вузли можна легко записати і використовувати в моделі.

Коли несправність активується у відповідності з заданими умовами спрацьовування, вона змінює вихідні дані компонента. Таке неправильне поведінка може призвести до порушення контрактів інших компонентів системи, включаючи припущення нижчестоящих компонентів. Вплив несправності обчислюється засобом перевірки моделі безпеки при виконанні аналізу моделі безпеки несправності.

В якості ілюстрації моделювання несправностей з використанням Програми з безпеки ми розглянемо один з компонентів, важливих для властивості ненавмисного гальмування: педаль гальма. Коли механічна натиснута педаль, датчик зчитує цю інформацію і передає електронний сигнал в BSCU, який потім командує гідравлічним тиском на колеса.

за його номінальну поведінку. Вираз true → властивість ЗГІДНО істинно в початковому стані, а потім воно істинно тільки в тому випадку, якщо властивість виконується.) Датчик має тільки один вхід-механічне положення педалі і один вихід-електричне положення педалі. Властивість, яка управляє

поведінкою компонента, полягає в тому, що механічне положення завжди повинно відповідати електронного стану.

Одним з можливих збоїв датчика педалі є інверсія його вихідного значення. Ця помилка може бути викликана з імовірністю 5,0×10-6, як описано в AIR6110 (на практиці ймовірність відмови компонента визначається специфікацій обладнання). Визначення програми з безпеки для цієї несправності показано на рис. 8. Поведінка несправності визначається за допомогою сайту несправності, званого inverted_fail. При спрацьовуванні несправності номінальний вихід компонента (elec_pedal_position) замінюється значенням його несправності (val_out).

Модель несправностей WBS, представлена в Додатку з безпеки, містить в загальній складності 33 визначення несправностей і 141 примірник несправностей. Велика кількість випадків збоїв пов'язано з надмірністю в конструкції системи і її реплікацією для управління 8 колесами.

3.3.5.1. Неявне поширення помилок

У цьому підході помилки фіксуються як неправильна поведінка, яке доповнює модель поведінки системи в договорах про ЗГОДУ. Явного поширення помилок не потрібно, так як неправильна поведінка поширюється через номінальні контракти поведінки в моделі системи так само, як і в реальній системі. Наслідки будь викликаної помилки виявляються через аналіз договорів, укладених за УГОДОЮ.

Навпаки, в Додатку до моделі помилок AADL, Версія 2 (EMV2) [179], всі помилки повинні бути явно поширені через кожен компонент (шляхом застосування типів помилок на кожному з вихідних портів), щоб компонент чинив вплив на іншу частину системи. Щоб проілюструвати ключові відмінності між неявним поширенням помилок, передбаченим у додатку з безпеки, і явним поширенням помилок, передбаченим у EMV2, ми використовуємо спрощений поведінковий потік з прикладу WBS з використанням фрагментів коду з EMV2, ПОГОДЬТЕСЯ, і програми з безпеки (рис. 9).

У цій спрощеній системі WBS датчик виявляє фізичний сигнал від компонента педалі, і значення положення педалі передається компонентів BSCU. BSCU генерує команду тиску на компонент клапана, який прикладає тиск гідравлічного гальма коліс.

У підході EMV2 (верхня половина рис. 9) помилка 'NoService" явно розповсюджується по всім компонентам. Ці типи несправностей по суті є маркерами, а не специфікацією несправного поведінки. На системному рівні інструменти аналізу, що підтримують додаток EMV2, можуть об'єднувати інформацію про поширення від різних компонентів, щоб скласти загальну схему потоку несправностей або дерево несправностей.

При виникненні несправності в додатку з безпеки (нижня половина рис. 9) поведінку вихідного компонента датчика змінюється. У цьому випадку результатом є помилка 'застряг на нулі". Поведінка BSCU отримує нульовий вхідний сигнал і реагує, як якщо б педаль не була натиснута. Це призведе до збою системи верхнього рівня: натискання педалі означає, що вихідний тиск на гальма позитивне.

```
system SensorPedalPosition
  features
       -- Input ports for subcomponent
       mech_pedal_pos : in data port Base_Types::Boolean;
       elec_pedal_pos : in data port Base_Types::Boolean;

  -- Behavioral contracts for subcomponent
  annex agree {**

     guarantee "Mechanical and electrical pedal position is equivalent" :
       true -> (mech_pedal_position = elec_pedal_position;
  };
```

Рис. 3.7. Тип системи AADL: Датчик педалі.

```
annex safety {**
  fault SensorPedalPosition_ErroneousData "Inverted boolean fault" : faults.inverted_fail {
       inputs: val_in <- elec_pedal_position;
       outputs: elec_pedal_position <- val_out;
       probability: 5.0E-6 ;
       duration: permanent;
  }
};
```

Рис. 3.8. Додаток щодо безпеки для Датчика педалі.

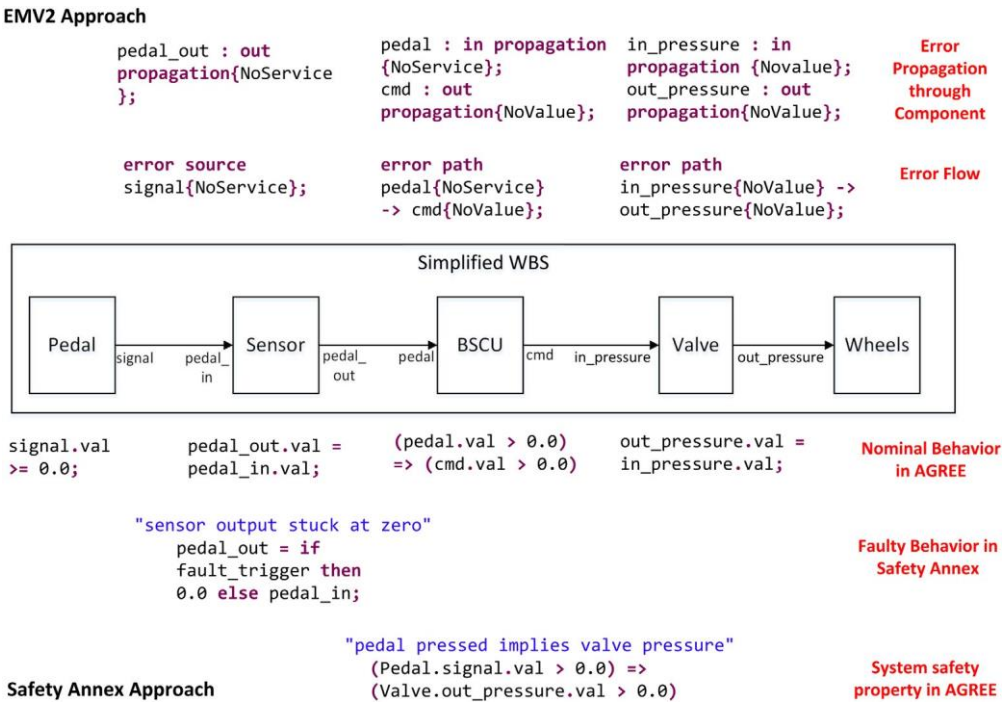


Рис. 3.9. Відмінності між Додатком з безпеки і EMV2.

3.3.6.2. Явне поширення помилок

Збої в апаратних компонентах (HW) можуть викликати поведінкові збої в компонентах системи, які залежать від них. Наприклад, збій Центрального процесора (ЦП) може викликати неправильне поводження в потоках, пов'язаних з цим ЦП. Крім того, збій в одному компоненті HW може спровокувати збій в інших компонентах HW, розташованих поблизу, таких як перегрів, пожежа або вибух в захисній оболонці. Додаток з безпеки надає можливість явно моделювати вплив апаратних збоїв на інші збої, незалежно від того, чи є вони залежними або незалежними. Явне поширення на помилки, не пов'язані з поведінкою, аналогічно тому, що передбачено в EMV2.

Для поліпшення моделей несправностей на системному рівні, що залежать від збоїв HW, вводиться елемент моделі несправностей, званий апаратної несправністю. Користувачам не потрібно вказувати поведінкові ефекти для несправностей HW, а також порти даних, до яких необхідно застосувати визначення несправності. Приклад оголошення про несправності компонента моделі показаний нижче:

Користувачі визначають залежності між несправностями компонента HW і несправностями, визначеними в інших компонентах, або в HW, або у програмному забезпеченні (SW). Апаратна несправність потім діє як тригер для

залежних несправностей. Це дозволяє простому розповсюдженню від несправного компонента SW до компонентів SW, які покладаються на нього, впливаючи на поведінку на виходах порушених компонентів SW.

У прикладі WBS припустимо, що зелений і синій гідравлічні насоси розташовані в одному і тому ж відсіку літака, і вибух в цьому відсіку вивів обидва насоса з ладу. Визначення несправності HW може бути спочатку змодельоване в зеленому компоненті гідравлічного насоса, як показано на рис. 10. Активація цієї несправності запускає активацію пов'язаних несправностей, як показано в інструкції propagate_to, показаної на рис. 11. Зверніть увагу, що ці насоси не повинні бути підключені через порт даних, щоб вказати це поширення.

Залежно від несправностей зазначаються у реалізації системи, де стає ясною конфігурація системи, що викликає залежності (наприклад, прив'язка між компонентами SW і HW, спільне розташування компонентів HW).

```
HW_fault Pump_HW_Fault "Colocated pump failure": {
    probability: 1.0E-5;
    duration: permanent;
}
```

Рис. 3.10. Визначення апаратної несправності.

```
annex safety{**

   analyze : probability 1.0E-7
   propagate_from:
     {Pump_HW_Fault@phys_sys.green_hyd_pump} to {HydPump_FailedOff@phys_sys.blue_hyd_pump};

**};
```

Рис. 3.11. Заява про поширення апаратних несправностей.

3.3.6.3. Асиметричні помилки

Асиметрична помилка-це помилка, яка представляє різні симптоми для різних спостерігачів [203]. Розглянемо вихідний компонент з виходом, підключеним до кількох входів на різних компонентах призначення. У цій конфігурації симетрична помилка призведе до того, що всі компоненти призначення будуть спостерігати одне і те ж хибне значення від вихідного компонента. При асиметричній несправності компоненти призначення можуть отримувати значення, відмінні від вихідних. Щоб зафіксувати поведінку асиметричних несправностей, необхідно було розширити наш механізм моделювання несправностей в AADL.

Щоб проілюструвати нашу реалізацію асиметричних несправностей, припустимо, що початковий компонент A має вихід 1 до багатьох, підключений до чотирьох компонентів призначення (Be), як показано на рис. 12 в розділі 'Номінальна система'. Якщо б на цьому виході була симетрична несправність, всі чотири підключених компонента вели б себе однаково. Асиметрична несправність повинна мати можливість передавати підключеним компонентів довільно різні значення.

З цією метою 'вузли зв'язку' автоматично вставляються в кожне з'єднання від компонента A до компонентів B, C, D і E (показано на рис. 12 в розділі 'Архітектура моделі несправності'). З точки зору користувачів, асиметричне визначення несправності пов'язано з виходом компонента A, і архітектура моделі не відрізняється від номінальної архітектури моделі. За лаштунками ці вузли зв'язку створюються для полегшення потенційно різних активацій збоїв у кожному з цих сполук. Визначення несправності, що використовується на виході компонента A, буде вставлено в кожен з цих вузлів зв'язку, як показано червоними кружками на виході вузла зв'язку на рис. 12.

Асиметрична несправність визначена для компонента A, як показано на рис. 13. Ця несправність визначає асиметричну несправність на компоненті A, яка, будучи активною, застряє на попередньому значенні (до(вихід, 0)). Це можна інтерпретувати наступним чином: деякі підключені компоненти можуть бачити лише попереднє значення вихідного компонента A, а інші можуть бачити правильне (поточне) значення, коли активна помилка. Це визначення несправності вводиться у вузли зв'язку, і те, який з підключених компонентів бачить невірне значення, повністю недетермінованої. Будь-кількість несправностей вузла зв'язку (0.. все) може бути викликано при активації основний асиметричної несправності на вихідному виході.

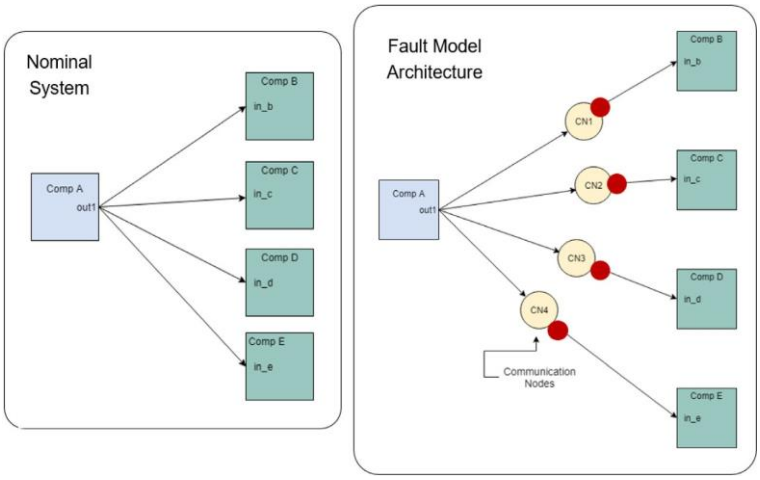


Рис. 3.12. Вузли зв'язкуїй при реалізації асиметричної несправності.

```
fault asymmetric_fault_Comp_A "Component A output asymmetric" : faults.fail_to {
        inputs: val_in <- Output, alt_val <- prev(Output, 0);
        outputs: Output <- val_out;
        probability: 5.0E-5;
        duration: permanent;
        propagate_type: asymmetric;
}
```

Рис. 3.13. Визначення асиметричної несправності в Додатку по техніці безпеки.

3.3.6.4. Заяви про аналіз несправностей

Оператор аналізу несправностей (також званий гіпотезою несправності) знаходиться в реалізації системи AADL, обраної для перевірки. Це може вказувати максимальну кількість помилок, які можуть бути активні в будь-який момент виконання (рис. 14).

В якості альтернативи в заяві про аналіз несправностей може бути зазначено, що розглядаються тільки ті несправності, ймовірність одночасного виникнення яких перевищує деякий поріг ймовірності (рис. 15).

Прив'язка до аналізу дерева несправностей у традиційному аналізі безпеки, перший аналогічний обмеження наборів скорочень заданим максимальним числом термінів, а другий аналогічний обмеження наборів скорочень тільки тими, ймовірність одночасного виконання яких перевищує деяке задане значення. У першому випадку ми стверджуємо, що сума справжніх тригерних змінних fault__знаходиться на рівні або нижче деякого цілочисельного порога. В останньому випадку ми визначаємо всі комбінації несправностей, ймовірність яких перевищує зазначений поріг вірогідності, і описуємо це як припущення про__тригерних змінних несправності.

З введенням залежних несправностей активні несправності діляться на дві категорії: незалежно активні (активуються власним запускає подією) і залежно активні (активуються, коли стають активними залежні від них несправності). Гіпотеза несправності верхнього рівня може бути застосована незалежно активним несправностей. Неправильна поведінка доповнює номінальна поведінка всякий раз, коли активні відповідні йому помилки (незалежно чи залежно).

```
annex safety {**
        analyze : max 1 fault
**};
```

Рис. 3.14. Інструкція з аналізу помилок Max N.

```
annex safety {**
        analyze : probability 1.0E-7
**};
```

Рис. 3.15. Заява про імовірнісному аналізі.

3.3.7. | Аналіз моделі несправностей

Існує два основних варіанти аналізу моделі несправностей. Перший варіант вводить помилкова поведінка, допускається гіпотезою помилковою, в модель УЗГОДЖЕННЯ і повертає цю анотовану програму Lustre з помилками в JKind для аналізу. Впровадження неправильного поведінки в модель УЗГОДЖЕННЯ дозволяє виявити наявність помилок в моделі, а інформація про простежування дозволяє користувачам переглядати контрприклад до порушення контракту при наявності помилок. Другий варіант аналізу використовується для створення мінімальних наборів вирізів для моделі. Шлях від моделі несправностей, записаної користувачем, до JKind однаковий в обох видах аналізу, але в анотаціях несправностей вказується, які результати обчислювати і відображати користувачу.

3.3.7.1. Перевірка на наявність несправностей: імовірнісний аналіз

Враховуючи ймовірну гіпотезу про несправності, це відповідає виконанню аналізу з комбінаціями несправностей, ймовірність одночасного виникнення яких менше порогового значення ймовірності. Це робиться шляхом вставки тверджень, які допускають ці комбінації в коді Lustre. Якщо перевірка моделі доведе, що властивості безпеки можуть бути порушені за допомогою будь-якої з цих комбінацій, буде показана одна з таких комбінацій у контрпримере. Ця форма аналізу виконується не за допомогою перевірки ймовірнісної моделі, а з допомогою ймовірнісних обчислень, виконуваних після завершення поведінкового аналізу. Передбачається, що несправності виникають незалежно, і можливі комбінації несправностей обчислюються і передаються в модель Lustre для перевірки перевіряє моделлю.

Алгоритм 1: Монолитный Вероятностный анализ

```
1  F = {} : fault combinations above threshold ;
2  Q : faults, q_i, arranged with probability high to low ;
3  R = Q , with r ∈ R;
4  while Q ≠ {} ∧ R ≠ {} do
5      q = removeTopElement(Q) ;
6      for i = 0 : |R| do
7          prob = q × r_i ;
8          if prob < threshold then
9              removeTail(R, j = i : |R|);
10         else
11             add({q, r_i}, Q);
12             add({q, r_i}, F);
```

Як показано в алгоритмі 1, обчислення спочатку видаляє з розгляду всі помилки, які дуже малоймовірні з урахуванням порогу вірогідності. Інші помилки розташовуються в пріоритетної черги *R* від високого до низького. Припускаючи незалежність в наборі помилок, ми беремо помилку з найбільшою вірогідністю черги (крок 5) і намагаємося об'єднати залишилися помилки в *R* (крок 7). Якщо ця комбінація нижче порогового значення (крок 8), то ми не беремо до уваги цей набір помилок і замість цього видаляємо хвіст залишилися помилок в *R*.У цьому розрахунку ми припускаємо незалежність серед несправностей.

3.3.7.2. Створення мінімальних наборів розрізів: Максимум n аналізів

Генерація мінімальних наборів вирізів була виконана в гальмівній системі колеса, і результати показано в таблиці 2. Зверніть увагу, що в Таблиці 2 мітка у верхньому рядку вказує потужність (n), а у відповідному стовпчику показано, скільки наборів вирізів створено для цієї потужності. При виконанні аналізу користувач вказує значення n. Це дає вирізані набори потужності менше або дорівнює n. У таблиці 2 показано загальну кількість вирізаних наборів потужності n. Загальна кількість наборів вирізів, обчислене на заданому порозі, являє собою суму по рядку. (Повний текст властивостей див. таблицю 1.)

Як видно з таблиці 2, кількість наборів розрізів збільшується експоненціально залежно від потужності наборів розрізів. Інтуїтивно це можна зрозуміти як просту комбінацію несправностей, які можуть порушити безпеку; чим більше в системі одночасно відбувається помилок, тим більше ймовірність порушення властивості. Властивість S18-WBS-0324 з максимальною гіпотезою про несправності 5 не вдалося завершити через помилки брак пам'яті. На момент виникнення помилки кількість вирізаних наборів перевищила 1,5

мільйона. На практиці неможливо вручну просіяти кілька тисяч наборів зрізів, але замість цього аналітик буде фільтрувати комбінації, які з достатньою ймовірністю виникнуть, виходячи з меж усікання. У наступному підрозділі (Створення мінімальних наборів скорочень: Імовірнісний аналіз) ми обговоримо використання межі усікання з допомогою ймовірнісного аналізу. Імовірнісний підхід представляє для розгляду більш реалістичне і корисне кількість наборів розрізів.

3.3.7.3. Створення мінімальних наборів розрізів: імовірнісний аналіз

Як імовірнісний аналіз аналіз, таак т max n використовують один і той же базовий алгоритм генерації мінімального набору розрізів (див. Розділ 3.3), але в імовірнісному аналізі мінімальні набори розрізів скорочуються, щоб включати тільки ті комбінації несправностей, ймовірність одночасного виникнення яких перевищує заданий поріг у гіпотезі.

Імовірного аналізу для WBS був привласнений поріг верхнього рівня для кожного властивості, як зазначено в AIR6110 і показано в таблиці 1. Ймовірність виникнення несправностей, пов'язаних з різними компонентами, що була задана згідно з документом AIR6110 [183]. У таблиці показано назва властивості і пов'язана з ним ймовірність. Генерація мінімальних наборів скорочень надала всі набори, які порушують це властивість, сукупні ймовірності яких (за умови незалежності) перевищують порогове значення. Кількість комплектів на потужність зазначено в таблиці.

Як показано в таблиці 3, кількість допустимих комбінацій значно зменшується при заданому імовірнісному порозі порівняно з просто комбінаціями несправностей певних потужностей. Наприклад, в одному контракті (ненавмисне гальмування коліс всіх коліс) було отримано більше мільйона мінімальних наборів скорочень, якщо розглядати його з точки зору аналізу max N, але після врахування ймовірностей в таблиці 3 видно, що вірогідними факторами, що сприяють виникненню небезпеки, є мінімальні набори скорочень потужності один. Імовірнісний аналіз виключив з розгляду багато тисяч наборів розрізів.

В таблиці 3 властивість 0321 має межа скорочення 1,0 × 10-9 з 8 окремими точками відмови. Якщо ця властивість має катастрофічну класифікацію, ці окремі точки відмови повинні бути усунені. Аналогічно з наборами перерізів n = 2, існує в загальній складності 3665 комбінацій, які аналітик безпеки повинен вивчити вручну. В рамках цього аналізу існує кілька способів вирішення проблеми кількості наборів вирізів. Один з них полягає в повторному вивченні

того, як моделюються несправності (наприклад, об'єднання двох режимів відмови клапана в один, оскільки відмова відкрити і відмова закрити не можуть відбутися одночасно), а інший - в переоцінці конструкції моделі, яка докладно обговорюється в наступному підрозділі (Використання результатів аналізу для зміни конструкції).

Таблиця 3.2 Результати набору мінімального розрізу WBS для гіпотези n max.

Результати набору мінімального розрізу WBS для гіпотези n max.					
Хар-ка	n= 1	n= 2	n= 3	n= 4	n= 5
0321	7	0	0	256	57,600
0322-R	75	0	0	0	0
0322-L	75	0	0	0	0
0323	182	0	0	0	0
0324	8	3665	28 694	883 981	-
0325-WX	33	0	0	0	0

Таблиця 3.3 Результати набору минимального розрізу WBS для вероятностной гіпотези.

Результати набору мінімального розрізу WBS для вероятностной гіпотези.					
Хар-ка	n= 1	n= 2	n= 3	n= 4	n= 5
$0321{:}5.0 \times 10^{-7}$	0	0	256	0	0
$0322\text{-}R{:}5.0 \times 10^{-7}$	75	0	0	0	0
$0322\text{-}L{:}5.0 \times 10^{-7}$	75	0	0	0	0
$0323{:}1.0 \times 10^{-9}$	182	0	0	0	0
$0324{:}1.0 \times 10^{-9}$	8	3665	0	0	0
$0325\text{-}W1{:}1.0 \times 10^{-9}$	33	0	0	0	0

3.3.7.4. Подання результатів аналізу мінімальних наборів розрізів

Результати аналізу генерації мінімальних наборів розрізів можуть бути представлені в одній з наступних форм.

1. Мінімальні набори вирізів можуть бути представлені в письмовій формі із зазначенням загальної кількості для кожного властивості, потужності кожного і рядків описи, що показують властивість та інформацію про несправності. Зразок цього висновку показаний на рис. 16.

2. Інформація про мінімальному наборі зрізів може бути представлена у формі підрахунку. Це не містить детальної інформації про несправності, а замість цього дає тільки підрахунок наборів зрізів для кожного властивості. Це корисно у великих моделях з великою кількістю наборів вирізів, так як це зменшує розмір текстового файлу. Приклад цього типу висновку показаний на рис. 17.

```
Minimal Cut Sets for property violation:
property lustre name: safety___GUARANTEE1
property description:  lemma: (S18-WBS-R-0322-left) Asymmetrical left braking.
Total 18 Minimal Cut Sets found for this property
Probability of failure for the overall property: 3.201E-4

Minimal Cut Set # 1
Cardinality 1
original fault name, description: Accumulator_Failed,
"(Accumulator) Stuck nondet fault."
lustre component, fault name: phys_sys,
phys_sys_fault__independently__active__accumulator__fault_1
probability: 5.0E-5

Minimal Cut Set # 2
Cardinality 1
original fault name, description: HydraulicPiston_Failed,
"(HydraulicPiston) Stuck nondet fault."
lustre component, fault name: wheel_brake3,
wheel_brake3_fault__independently__active__normal_hyd_piston__fault_1
probability: 3.3E-5
```

Рис. 3.16. Детальний висновок мінімального набіра розрізів.

```
Minimal Cut Sets for property violation:
property lustre name: safety___GUARANTEE1
property description:  lemma: (S18-WBS-R-0322-left) Asymmetrical left braking.
Total 18 Minimal Cut Sets
Cardinality 1 number: 18

Minimal Cut Sets for property violation:
property lustre name: safety___GUARANTEE2
property description:  lemma: (S18-WBS-R-0322-right) Asymmetrical right braking
Total 18 Minimal Cut Sets
Cardinality 1 number: 18

Minimal Cut Sets for property violation:
property lustre name: safety___GUARANTEE0
property description:  lemma: (S18-WBS-R-0321) Never loss of all wheel braking
Total 6 Minimal Cut Sets
Cardinality 1 number: 6
```

Рис. 3.17. Підсумковий висновок мінімального набіра розрізів.

3.3.8. | Використання результатів аналізу для внесення змін у конструкцію

В даній роботі використовуємо одну вимогу вищого рівня WBS, щоб проілюструвати, як Програма по безпеці може використовуватися для виявлення недоліків конструкції і як несправності можуть вплинути на поведінку системи (S18-WBS-0323: Ніколи не допускайте випадкового гальмування при заблокованих всіх колесах). Це опис властивостей безпеки

можна докладно знайти в таблиці 1. При виконанні аналізу композиційних несправностей max n з $n = 1$ було показано, що ця конкретна несправність є єдиною точкою для відмови властивості безпеки. Контрприклад показаний на рис. 18, показує активну несправність датчика педалі.

Щоб усунути цю проблему, можна додати надмірність для обробки одного несправного датчика за допомогою трьох датчиків. Загальний вихідний сигнал системи датчиків може використовувати схему голосування для визначення достовірності показань датчика. Існує декілька можливих схем голосування, одна з яких-голосування більшістю голосів. При наявності трьох датчиків це зменшує проблему з однією точкою відмови. Нові поведінкові контракти додаються в сенсорну систему для моделювання поведінки надмірності і голосування.

У разі датчика педалі в WBS була реалізована остання з двох стратегій, описаних вище. В модель була додана сенсорна система, яка містила три датчика педалей. Вихід цієї підсистеми був обмежений з використанням схеми голосування більшістю голосів. При наступних запусках аналізу (незалежно від того, який тип запуску використовувався) в системі була підтверджена стійкість щодо відмови одного датчика педалі. На рис. 19 показані ці архітектурні зміни, внесені в модель.

Як видно з цього єдиного прикладу, така велика система, як WBS, виграла б від багатьох ітерацій цього процесу. Крім того, якщо модель буде змінена навіть трохи з боку розробки системи, це автоматично вплине на аналіз безпеки, і будь-які негативні результати будуть показані при наступних запусках аналізу. Це ефективно усуває будь-які непорозуміння між командами розробки та аналізу систем і створює нову гарантію щодо змін моделі.

Для отримання додаткової інформації про типи моделей несправностей, які можуть бути створені, а також дані про результати аналізу див. Керівництво користувача, розташоване в репозиторії GitHub [202]. Цей репозиторій також містить усі моделі, використовувані в цьому проекті.

Problems | Properties | AADL Property Values | Classifier Information | AGREE Results | AGREE Counterexample | Console

Name	Step 1	Step 2
pedal_sensor_R		
pedal_sensor_R		
lemma: (S18-WBS-0323) Never inadvertent braking with all wheels locked	true	false
(SensorPedalPosition) Inverted boolean fault		
(pedal_sensor_L__fault_1)	false	false
(pedal_sensor_R__fault_1)	true	true
ALL_WHEELS_BRAKE	true	true
ALL_WHEELS_STOPPED	false	false
BRAKE_AS_NOT_COMMANDED	false	false
HYD_PRESSURE_MAX	true	true
PEDALS_NOT_PRESSED	true	false
POWER	false	true
SPEED	true	true
W1ROLL	true	true

Рис. 3.18. ПОГОДЖЕНИЙ контрприклад для властивості безпеки при випадковому гальмуванні.

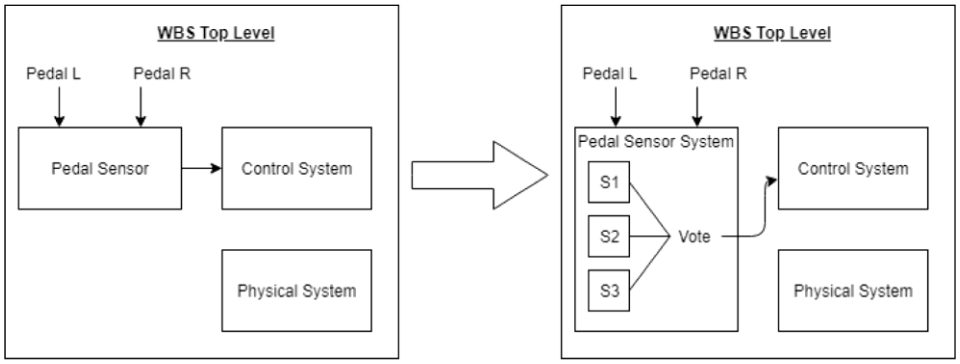


Рис. 3.19. Зміни в архітектурному моделі для усунення несправностей.

3.4. | Обговорення

Підхід, викладений у цьому розділі, не покликаний замінити аналітика безпеки, а швидше надати аналітикові додаткове уявлення про розробляються складних критичних системах. Інші аспекти оцінки безпеки, які паралельні якісному і кількісному аналізу безпеки, підтримуваного нашим підходом, будуть як і раніше вирішуватися традиційними способами (наприклад, рівні гарантії розробки/проектування або рівні цілісності компонентів). Наш внесок не служить заміною експертних знань аналітика з безпеки або охоплює весь процес оцінки, а замість цього забезпечує автоматизований і всеосяжний аналіз для перевірки вимог безпеки при наявності несправностей і отримання доказів для процесу оцінки. Це особливо корисно для все більш складних систем авіоніки з інтенсивним використанням програмного забезпечення, де ручної аналіз стає все більш складним для всебічного перерахування всіх можливих причинно-наслідкових зв'язків відмов.

Відсутність точних моделей архітектури системи і режимів її відмов часто змушує аналітиків безпеки докладати значні зусилля для збору архітектурних відомостей про поведінку системи з декількох джерел. Зазвичай володіючи знаннями про систему предметної області, але не володіючи докладними знаннями про те, як розробляються програмні додатки, інженери-практики з безпеки вважають складним і трудомістким процесом отримання інформації про поведінку програмних додатків, розміщених в системі, та їх вплив на загальну безпеку системи.

Одна з наших цілей-впровадити розроблені нами інструменти в роботу інженерів з безпеки, які проводять оцінку безпеки цифрових систем повітряних суден. Тому нам необхідно зрозуміти, як інструменти і моделі будуть вписуватися в існуючий процес оцінки і сертифікації безпеки. В нашій області інтересів поточний процес оцінки безпеки на системному рівні заснований на ARP 4754A [201] і ARP4761 [204]. Заснований на моделі підхід до аналізу безпеки був запропонований Джоші та ін. в [170,182,188]. При такому підході Модель системи аналізу безпеки (SASM) є центральним елементом процесу аналізу безпеки, а традиційні артефакти аналізу безпеки, такі як дерева несправностей, автоматично генеруються інструментами, які аналізують SASM.

Зміст і структура SASM істотно розрізняються в різних концепціях MBSA. Ми можемо провести відмінності між підходами по декількох різних напрямках. По-перше, чи поширюються вони явно з допомогою користувацьких методів розповсюдження, які ми називаємо Моделюванням логіки збоїв (FLM), або з допомогою існуючого поведінкового моделювання, яке ми називаємо Моделюванням ефекту збою (FEM). Наступне питання полягає в тому, призначені моделі і позначення для аналізу безпеки порівняно з тими, які розширюють існуючі моделі систем (ESM).

Для підходів FEM існує кілька додаткових вимірювань. Один вимір включає в себе питання про те, можна чи каузальные або не каузальные моделі. Моделі без причинного зв'язку допускають одночасне (в часі) двостороннє поширення помилок, що дозволяє більш природно відображати деякі типи відмов (наприклад, зворотний потік в сегментах труби), але їх складніше аналізувати. Останнє вимірювання включає в себе визначення того, чи є аналіз композиційним по верствам ієрархічно складених систем або монолітним. Наш підхід є продовженням AADL (Існуючої моделі системи (ESM)), причинного, композиційного, змішаного підходу FLM/FEM.

Такі інструменти, як додаток до моделі помилок AADL, Версія 2 (EMV2) [179], HiP-HOPS для EAST-ADL [183] та Ansys Medini [184], являють собою

підходи ESM на основі FLM. Як обговорювалося раніше, враховуючи безліч можливих помилок, ці взаємозв'язки поширення вимагають значних зусиль користувача і стають більш складними. Крім того, відповідальність за визначення того, чи можуть поширюватися помилки, лягає на аналітика; відсутність поширення призводить до неправильного аналізу. У додатку з безпеки поширення відбувається за допомогою поведінки системи (певного номінальними контрактами) без додаткових зусиль користувача.

На рис. 20 показана довідкова таблиця, в якій перераховані деякі відповідні робочі інструменти, які ми описуємо у решти цього розділу. На рисунку показані важливі особливості наданої підтримки. Тісно пов'язаний з нашою роботою набір інструментів для оцінки безпеки на основі моделей, званий COMPASS (Коректність, проект моделювання і продуктивність аерокосмічних систем) [11]. COMPASS-це набір причинно-наслідкових інструментів на основі FLM/FEM, який використовує мову SLIM, заснований на підмножині AADL, для своїх вхідних моделей [205,206]. У SLIM модель номінальної системи і модель помилок розробляються окремо, а потім перетворюються в розширену модель системи, і перевірка виконується по цій розширеній моделі.

Інша пов'язана з цим робота включає SmartIFlow [172], який являє собою спеціально створений монолітний інструмент аналізу безпеки на основі FEM, не пов'язаний з причинними факторами, що описує компоненти та їх взаємодії з використанням кінцевих автоматів і подій. Перевірка виконується за допомогою явної перевірки моделі стану, яка повертає набори контрприкладів для вимог безпеки при наявності збоїв. Мова аналізу і моделювання безпеки (SAML) [174] - це заснований на FEM, спеціально побудований, монолітний мова аналізу причинного безпеки. AltaRica [185,186] - це заснований на FEM, спеціально побудований, монолітний мова аналізу безпеки з кількома діалектами. Існує один діалект AltaRica, який використовує потокове (причинну) семантику, в той час як останнє оновлення мови (AltaRica 3.0) використовує не причинний семантику. Діалект потоку даних має істотну інструментальну підтримку, в тому числі комерційний інструмент Cecilia OCAS від Dassault [207]. MADe-це інтегрований набір інструментів на основі моделі, який дозволяє користувачам ідентифікувати збої на основі функціональних залежностей, зафіксованих у моделі, і генерує графічні представлення поширення збоїв [208].

Інструменти формальної перевірки, засновані на перевірці моделей, що використовувалися для автоматизації створення артефактів безпеки [209-211–211], але цей підхід має обмеження з погляду масштабованості та доступності

згенерованих дерев несправностей. Була проведена робота щодо пом'якшення цих обмежень шляхом масштабованого створення читаються дерев несправностей [195].

На відміну від відповідної роботи, обговорювалася раніше, додаток з безпеки підтримує перевірку моделей та кількісні міркування, пов'язуючи поведінкові помилки з компонентами, а потім використовуючи механізми поширення та докази нормального поведінки, вбудовані в додаток AGREE AADL. Це дозволяє користувачам міркувати про еволюцію несправностей з плином часу і створювати контрприклади, що демонструють, як несправності компонентів призводять до відмов. Наш підхід розширює і адаптує роботу Джоші та ін [170] на мову моделювання AADL. Інструмент і документація доступні під ліцензією BSD і можуть бути розташовані за адресою: https: //github.com/loonwerks/AMASE/.

EXISTING MBSA TOOLS AND METHODS

	Modeling		Analysis Capabilities			
	Supports shared system/safety model	Support Failure Effect Modeling (no explicit failure propagation)	Support compositional verification in the presence of faults	Support verification of pilot response in the presence of faults	Generate Minimal Cut Sets	Generate Fault Trees
Safety Annex	✓	✓	✓	✓	✓	✓[1]
EMV2	✓				✓	✓
Compass		✓		✓	✓	✓[2]
xSAP		✓		✓	✓	✓[2]
Ansys Medini Analyze	✓				✓	✓
Altarica OCAS		✓			✓[3]	✓
HiP-HOPS	✓				✓	✓
MADe	✓				✓[4]	✓

✓[1]→ Textual fault trees are flat within verification layer (hierarchies align with compositional verification)
✓[2]→ Fault trees are flat (hierarchical fault tree planned for next releases)
✓[3]→ Support via third-party plugins
✓[4]→ Not explicitly documented but theoretically feasible.

Рис. 20. Відповідні інструменти і методи MBSA.

3.5. | Висновок

В даній роботі для розширення мови AADL з підтримкою інструментів для формального аналізу властивостей безпеки системи при наявності несправностей. Номінальна модель доповнено визначеннями несправностей, що дозволяє проводити аналіз безпеки та впровадження системи на основі єдиної загальної моделі. Ми виявили, що тісна інтеграція аналізу поведінкових помилок в AADL забезпечила тісний зв'язок між аналітиками системи та

безпеки. Зміни, внесені в модель системи, були негайно відображені як у номінальному аналізі, так і в аналізі безпеки.

Використання формальних методів підтримує всебічне дослідження впливу поведінки несправних компонентів на стан відмови на системному рівні без необхідності додавати модель окремі специфікації поширення. Як тільки специфікації інтерфейсу були написані з використанням AGREE і на вихідних даних компонентів були визначені несправності, процес перевірки дозволив аналітику відразу побачити, як поширення помилок вплинуло на систему. Вплив активної несправності не вимагалося визначати вручну, щоб побачити поведінку системи при наявності несправностей.

Цей підхід був проілюстрований на прикладі використання авіаційної системи, але може бути застосований при розробці критично важливих систем у багатьох галузях (наприклад, киберфизические системи, атомні електростанції, розробка автомобілів).

Майбутня робота включає компіляцію мінімальних наборів розрізів у форматі графічного дерева несправностей, розширення користувальницького інтерфейсу для спрощення створення моделі несправності і перетворення контрпримера в потік послідовності, що показує, як система змінюється при активації несправностей. Дослідження, представлені в данній роботі, а також внесок майбутньої роботи - все це служить підтримці процесу оцінки безпеки. Ці матеріали не охоплюють весь процес оцінки, а замість цього спрямовані на забезпечення автоматизованого та всеохоплюючого аналізу, а також на збір фактичних даних для оцінки.

Розділ 4 Охорона праці

Охорона праці відіграє важливу роль як суспільний чинник, оскільки, якими би вагомими не були трудові здобутки, вони не можуть компенсувати людині втраченого здоров'я, а тим більше життя.

Тому, в Законі України «Про охорону праці» підкреслюється, що основним принципом державної політики України є пріоритет життя і здоров'я людини над результатами її праці.

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних, лікувально-профілактичних заходів, спрямованих на збереження життя, здоров'я, і працездатності людини в процесі праці.

В даному дипломному проекті «Технічні засоби діагностування та контролю бортових систем інформаційного обміну на літаку», суб'єктом який виконує роботу з дослідження систем інформаційного обміну є інженер-дослідник, який працює в лабораторії.

Мета розділу «Охорона праці» даної дипломної роботи полягає в тому, щоб забезпечити майбутнього інженера-дослідника інформацією, яка дозволить йому зберегти здоров'я та працездатність під час виконання своїх трудових обов'язків.

4.1 | Аналіз умов праці на робочому місці інженера-дослідника у виробничому приміщенні

4.1.1.Організація робочого місця інженера-дослідника

Робоче місце – це частина простору, в якому інженер здійснює трудову діяльність, і проводить більшу частину робочого часу. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, у відношенні простору, розміру та форми забезпечує зручне положення під час роботи і високу продуктивність праці при найменшому фізичному та психічному напруженні. При правильній організації робочого місця продуктивність праці інженера зростає з 8 до 20 відсотків.

Організація робочого місця інженера-дослідника повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 12.2.032 ССБТ, " Робоче місце при виконанні робіт сидячи. Загальна ергономічна вимога", характеру та особливості трудової діяльності.

Головними елементами робочого місця інженера-дослідника є письмовий стіл і крісло. Основним робочим положенням є положення сидячи. Робоче місце для виконання робіт у положенні сидячи організується відповідно до ГОСТ 12.2.032-78.

При роботі в положенні сидячи витримані такі параметри робочого простору: ширина - 700 мм, глибина - 400 мм, висота робочої поверхні столу над підлогою - 700-750 мм. Витримані оптимальними розмірами столу, котрі складають: висота 710 мм; довжина столу 1300 мм, ширина столу 650 мм. Робочий стіл має простір для ніг висотою 600 мм, шириною 500 мм, глибиною на рівні колін – 450 мм, на рівні витятої ноги –650 мм.

Важливим елементом робочого місця інженера-дослідника є крісло, яке має такі основні елементи: сидіння, спинку та стаціонарні або змінні підлокітники. Робоче сидіння задовольняє наступним вимогам: припускає можливість зміни положення тіла, тобто забезпечувати вільне переміщення корпуса і кінцівок тіла один відносно одного; дозволяє виконувати регулювання висоти в залежності від зросту інженера-дослідника (у межах від 400 до 550 мм), має злегка увігнуту поверхню, мати невеликий нахил назад.

Також одним з моментів продуктивної роботи інженера-дослідника є раціональне розміщення на робочому місці документації, канцелярських приладдя, що має забезпечити інженеру-досліднику зручну робочу позу, найбільш економічні рухи і мінімальну траєкторію переміщення предмета роботи на даному робочому місці.

4.1.2. Вид виробничого приміщення та основні його характеристики

Робоче місце інженера-дослідника знаходиться у лабораторії на третьому поверсі шестиповерхової будівлі та має такі параметри:

1. Довжина приміщення 8 м;
2. Ширина приміщення 4.5 м;
3. Висота приміщення 3 м.
4. Загальна площа приміщення 36 кв. м.
5. Об'єм приміщення 108 куб. м.

У виробничому приміщенні розташовано три робочих місця. Згідно з ДБН В.2.2.-28-2010 «Будівлі адміністративного та побутового призначення»: площу приміщень слід приймати з розрахунку не менше 8 м^2 на робоче місце працівника. У приміщенні 3 робочих місця 8 м^2 х 4 =32 м^2 , що менше загальної площі приміщення 36 кв. м, отже, площа приміщення відповідає вимогам.

Робочі місця з комп’ютером відносно світлових отворів доцільно

розташовувати таким чином, щоб природне світло падало збоку, переважно зліва. Робочі місця з комп'ютером повинні розташовуватися на відстані не менше 1м від стін зі світловими прорізами; відстань між бічними поверхнями столів з комп'ютерами має бути не меншою за 1,2м; відстань між тильною поверхнею одного комп'ютера та екраном іншого не повинна бути меншою ніж 2,5м. Прохід між рядами робочих місць не повинен бути меншим за 1м.

Конструкція робочого місця інженера-дослідника (при роботі сидячи) має забезпечувати підтримання оптимальної робочої пози з такими ергономічними характеристиками:

- ступні ніг - на підлозі або на підставці для ніг;
- стегна - в горизонтальній площині;
- передпліччя - вертикально;
- лікті - під кутом 70-90^{o} до вертикальної площини;
- зап'ястя - зігнуті під кутом не більше 20^{o} відносно горизонтальної площини;
- нахил голови - 15-20^{o} відносно вертикальної площини.

Комп'ютер розташований на робочому місці так, що поверхня екрану знаходилася на відстані 400-700 мм від очей інженера. Користування Комп'ютером основний вид діяльності, тому він розміщується на основному робочому столі, з лівого боку.

4.1.3. Небезпечні та шкідливі виробничі фактори, що можуть впливати на інженера-дослідника при проведенні дослідницької діяльності за допомогою Комп'ютера.

При проведенні дослідницької діяльності за допомогою Електронно-обчислювальних машин, на людину впливають наступні небезпечні і шкідливі виробничі фактори:

- низька чи занадто висока освітленість виробничого приміщення (штучне та природне освітлення);
- вплив електричного струму;
- високий рівень статичної електрики;
- неналежний стан мікроклімату: температура, вологість, швидкість руху повітря, теплове випромінювання.
- вібрації та шум;

- електромагнітне випромінювання.

4.2. | Аналіз небезпечних та шкідливих виробничих факторів, що впливають на інженера-дослідника

4.2.1 Аналіз освітленості лабораторії

Робота користувачів комп‘ютерів характеризується значним напруженням зорового аналізатора, тому важливе значення має забезпечення раціонального освітлення робочих приміщень.

Згідно «Правил охорони праці під час експлуатації електронно- обчислюваних машин», освітлення у приміщеннях з комп’ютерами має бути змішаним (природне і штучне).

Природне світло повинно проникати через бічні світлопрорізи, зорієнтовані, як правило, на північ чи північний схід. Вікна приміщень повинні мати регульовані пристрої для відкривання, а також жалюзі, штори зовнішні, зовнішні козирки тощо.

У лабораторії де виконуються роботи інженером-дослідником спостерігається нестача природного світла обумовлена тим, що робоче місце знаходиться далеко від джерела природного освітлення. Джерелом потрапляння природного освітлення у приміщення лабораторії представлене одностороннє пряме освітлення через 2 вікна, розміром 1х1,5 метра. Напрямок розміщення вікон північно-західний. Коефіцієнт природної освітленості ~ 1,35%, що не відповідає нормативним значенням коефіцієнта природної освітленості ДБН В.2.5-28-2006 «Природне і штучне освітлення». Тому у приміщенні лабораторії на робочому місці інженера-дослідника використовується змішане освітлення.

Штучне освітлення передбачається в усіх виробничих та побутових приміщеннях, де недостатньо природного світла, а також для освітлення приміщень у темний період доби. При організації штучного освітлення потрібно забезпечити сприятливі гігієнічні умови для здорової роботи і одночасно враховувати економічні показники.

Штучне освітлення здійснюється за допомогою системи загального рівномірного освітлення і через екрани комп’ютерів. Застосування світильників без розсіювачів та екранних сіток забороняється. Згідно з Державними будівельними нормами ДБН-В.2.5-28-2006 «Природне і штучне освітлення» на робочому місці інженера-дослідника нормативне значення освітленості має бути в межах 300 - 500 Лк, а фактичне значення освітленості складає 220-270 Лк. Це пов’язано із застарілістю системи освітлення.

Рівень освітлення повинен бути таким, щоб забезпечити можливість ергономічного проведення будь-яких робіт, оптимально – 400 Лк. Для штучного освітлення в даному випадку найкраще підходять LED-лампи, які мають один з найвищих показників світловіддачі.

В нашому випадку, на робочому місці інженера-дослідника, в технічному приміщенні використовуються світильники з використанням двох люмінесцентних ламп на кожному з світильників, які розміщені не над робочими місцями, а посередині приміщення, які в сумі дають рівень освітлення близько 270 Лк. Тому необхідно розробити заходи з підвищення рівня штучного освітлення на робочому місці приблизно до 400 Лк, так як рівень освітлення у приміщенні має задовольняти ряду вимог.

4.2.2 Захист від враження електричним струмом

Вибір, розміщення, виконання і клас ізоляції застосовуваних машин, апаратів і іншого електроустаткування виробляється відповідно до вимог державних стандартів системи стандартів безпеки праці (ССБТ) и "Правил устройства электроустановок ПУЭ – 76".

При проведенні дослідницької діяльності за допомогою Комп’ютера інженер-дослідник піддається небезпеці ураження електричним струмом. Для виключення можливого впливу електричного струму на інженера- дослідника корпуси Комп’ютерів повинні бути заземлені.

Штучні заземлення споруджують з вертикальних і горизонтальних заземлювачів. У якості вертикальних заземлювачів використовують сталеві стрижні – пруток і кутова сталь довжиною 2,5...3 м, а в якості самостійних горизонтальних заземлювачів і для зв'язку вертикальних – смугову сталь і сталевий пруток. Найменші розміри заземлювачів: діаметр пруткових не оцинкованих – 10 мм, оцинкованих – 6 мм; перетин пруткових не оцинкованих заземлювачів – 48 мм^2; товщина прямокутних заземлювачів (смугова сталь) і полиць кутової сталі – 4 мм.

4.2.3 Захист від статичної електрики

Джерелами електростатичного поля можуть бути будь-які поверхні або предмети, які легко електризуються за рахунок тертя: килими, лінолеум, лаковані покриття, одяг із синтетичної тканини, взуття, тощо. Крім того, джерелом електростатичних зарядів є сам Комп’ютер. На екранах Комп’ютерів накопичується електростатичний заряд і виникає електромагнітне поле, яке характеризується напруженістю.

Напруженість електростатичного поля залежно від типу Комп'ютера коливається від 8 до 75 кВ/м. Відповідно ГОСТ 12.1.045-84 "ССБТ. Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля" напруженість електростатичного поля на робочому місці не повинна перевищувати 20 кВ/м.

Поверхневий електростатичний потенціал ПК відповідно СН №1757-77 "Санитарно-гигиенические нормы допустимой напряженности электростатического поля" та СНиП 3.32-007-98 "Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин" не повинен перевищувати 500В.

Захист від електростатики та її небезпечних проявів досягається наступними заходами:

- усунення іонізації повітря нейтралізаторами статичної електрики;
- зменшення електропровідності поверхні шляхом підтримки у приміщенні з Комп'ютерами відносної вологості на рівні 40-60% ;
- застосуванням у приміщенні з комп'ютерами підлоги з антистатичним лінолеумом і проводженням вологого прибирання;
- для зняття статичного заряду кілька разів на день мити руки та обличчя водою, або час від часу торкатися металевих поверхонь, наприклад, зачищених від фарби місць на батареях центрального опалення;
- працівникам бажано носити одяг з природних або комбінованих (природних і штучних) волокон;
- як спецодяг працівникам рекомендуються бавовняні халати;
- періодично при вимкненому комп'ютері протирати ледве змоченим мильним розчином бавовняною ганчіркою пил з поверхонь апаратури.

4.2.4. Мікроклімат робочої зони інженера-дослідника

Параметри мікроклімату можуть змінюватись в широких межах, в той час як необхідною умовою життєдіяльності людини є підтримка постійності температури тіла завдяки властивості терморегуляції, тобто здатності організму регулювати віддачу тепла в навколишнє середовище. Основний принцип нормування мікроклімату є створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем. У санітарних нормах СН-245/71 встановлені величини параметрів мікроклімату, що створюють комфортні умови. Ці норми

встановлюються в залежності від пори року, характеру трудового процесу і характеру виробничого приміщення. Для приміщення лабораторії де працює інженер- дослідник котрий відноситься до категорії Іа (легкі роботи, не потребують фізичної напруги), допустимі та оптимальні значення параметрів мікроклімату приведені в таблиці 1.

Таблиця 4.1 - Норми мікроклімату для приміщень з Комп'ютерами

Пора року	Зона	Температура повітря, °С	Відносна вологість,%	Швидкість руху повітря, м / с
Холодний період	Оптимальна	18 - 21	60 - 40	<0.2
Перехідний період	Допустима	17 - 21	<75	<0.3
Теплий період року	Оптимальна	20 - 25	60 - 40	<0.3
	Допустима	<27 о 13 годині самого жаркого міс.	<75	<0.5

Приміщення з комп'ютерною технікою повинні бути обладнані системами опалення, кондиціонування повітря або ефективною вентиляцією.

У виробничих приміщеннях з перевищеним рівнем тепла необхідно встановити кондиціонер для досягнення оптимального рівня температури повітря.

Повітря, що надходить у приміщення, також, варто очищати від забруднення, у тому числі від пилу та мікроорганізмів.

Для підвищення вологості повітря в приміщеннях з Комп'ютерами варто застосовувати зволожувачі повітря, які заправляються щодня дистильованою або прокип'яченою питною водою.

4.3. | Розробка заходів з охорони праці

Розробка заходів з охорони праці полягає в створенні рекомендацій з розташування оптимальної кількості світильників необхідного типу в приміщенні для створення комфортних умов, що задовольняють всім нормам. Для покращення освітлення у приміщенні лабораторії необхідно виконати реконструкцію встановленої системи штучного освітлення. Потрібно змінити тип ламп, а також їх кількість та потужність.

Для даного приміщення необхідно встановити світильники з світлодіодними лампами (LED-лампи). В LED-лампах електричний струм перетворюється безпосередньо в світло і теоретично це можна зробити без великих енерговтрат. LED-лампи більш міцніші, і більш надійний, їх строк служби може досягати 500 тисяч годин, що майже в 5 - 10 раз більше, ніж у люмінесцентних ламп. Також LED-лампи низьковольтні, а тому безпечніші. В роботі будемо використовувати світильники з світлодіодними лампами Crystal 218 LED термін служби яких 50 000 годин, світловий потік 3200 Лм. Живлення системи освітлення здійснюється по мережі електроживлення з напругою 220В і частотою 50 Гц.

4.4 | Пожежна безпека

При виникненні пожежі на інженера-дослідника можуть впливати небезпечні чинники: відкритий вогонь та іскри; підвищена температура повітря, предметів, обладнання; дим, знижена концентрація кисню; обвалення і пошкодження будівель, споруд, установок, вибух.

Основними причинами пожежі та вибуху в приміщені лабораторії можуть бути наступні фактори:

- несправність та перенавантаження електричного обладнання;
- необережне ставлення до вогню (паління, використання відкритого вогню в недозволених місцях, залишання без нагляду електрообладнання);
- порушення правил пожежної безпеки;
- несправність виробничого обладнання.

Згідно з НАПБ А.01.001-2004 «Правила пожежної безпеки в Україні», для усунення цих причин необхідна підвищена дисципліна та встановлений суворий протипожежний режим. У виробничих приміщеннях повинні бути встановлені надійні засоби попереднього сповіщення небезпеки виникнення пожежі, та розміщенні схеми евакуації.

За пожежною і вибухонебезпечності приміщення належить до категорії «В», класу 2.

В приміщенні лабораторії встановлено систему пожежної сигналізації «ППКП Тирас-4П» з двома димовими сигналізаторами пожежі СПД-3, тепловим датчиком FT-A2S, також повинен бути встановлений пожежний сповіщувач, що реагує на дим IPS-H20P.

Основним критерієм для вибору вогнегасника в приміщенні є величина осередку можливої пожежі. Оскільки величина осередку пожежі в даному у приміщенні не значна, то достатньо встановлення двох переносних порошкових вогнегасників ВП-6.

4.5. | Розрахунок штучного освітлення

Розглянемо вплив освітлення на організм людини. Особливу увагу необхідно приділити питанню освітлення на робочому місці.

Рівень освітленості встановлюється в залежності від категорії зорових робіт. При роботі з ЕОМ він складає не менш E_{min} = 400 Лк.

Виробниче освітлення регулюється нормативно-технічними документами ГОСТ12.1.046-85, СНиП II-4-79. Освітлення на робочому місці повинно бути сполученим (природне і штучне світло). Природне освітлення повинно бути бічним. Коефіцієнт природньої освітленості повинний відповідати нормативним рівням по СНиП II-4-79: при виконанні робіт з категорії високої зорової точності – не нижче 1,5, при зоровій роботі середньої точності – не нижче 1.

Штучне освітлення варто здійснювати у виді комбінованої системи освітлення з використанням LED-ламп у світильниках загального освітлення. Вони повинні забезпечувати рівномірну освітленість за допомогою відбитого чи розсіяного світлорозподілу.

Визначимо норму загального штучного освітлення (кількості необхідних світильників) для забезпечення нормованої освітленості приміщення, застосувавши метод використання коефіцієнта світлового потоку. Основна розрахункова формула має вигляд:

$$F = \frac{E_{min} \cdot S \cdot К_з \cdot z}{N \times \eta} \qquad (4.1)$$

де:

- F – світловий потік лампи у світильнику, лм;
- Emin – норма (мінімум) освітленості, лк;
- S – площа приміщення, м^2;
- Кз – коефіцієнт запасу, що враховує старіння ламп і забруднення світильників (для LED ламп - =1.5);
- z – коефіцієнт нерівномірності освітлення; z = 1.2;
- N – число світильників, обумовлене з умови рівномірності освітлення;
- η – коефіцієнт використання світлового потоку;

Виконаємо розрахунок всіх компонент, що увійшли до формули (4.1).

1. Як вище було сказано, площа приміщення лабораторії складає - S = 36 м^2
2. Норма освітленості на робочих поверхнях в лабораторії складає 400 Лк, так як розряд зорових робіт рівний – IV.

3. Вибирається схема розміщення світильників в залежності від ширини приміщення (кількість рядів світильників): в даному технічному приміщенні розташовані дві лампи по центру приміщення.
4. Визначаємо індекс приміщення за наступною формулою:

$$\lambda = \frac{A \cdot B}{H_p \cdot (A + B)} \tag{4.2}$$

де:

- А – довжина приміщення *А=8м* ;
- В – ширина приміщення *В=4.5м*;
- Н ρ – висота підвісу світильників над робочою поверхнею, м.

Висоту підвісу світильників над робочою поверхнею (H ρ) знаходимо за допомогою формули:

$$H_p = H - h_n - h_c \tag{4.3}$$

де:

- H – висота приміщення, *Н=3м*;
- h_n – висота робочої поверхні над підлогою, h_n*=0.85м*;
- h_c – відстань світлового центру світильника від стелі h_c*=0.15м*.

Визначимо висоту підвісу світильників, підставивши вихідні значення у формулу (5.2), отримаємо:

$$H_p = 3 - 0.85 - 0.15 = 2(\text{м}) \tag{4.4}$$

Далі визначимо значення індексу приміщення λ за формулою (5.2), підставивши в неї вище визначенні параметри, отримаємо:

$$\lambda = \frac{8 \cdot 4.5}{2 \cdot (8 + 4.5)} = 1.44 \tag{4.5}$$

За відомим індексом приміщення λ і коефіцієнтам світлового потоку від підлоги – 30% (0,3), від стін – 40% (0,4) і від стелі – 60% (0,6) визначаємо для світильника значення коефіцієнта використання світлового потоку (η):

η = 0,8

Підставляємо у формулу світлового потоку (5.1) розраховані значення параметрів, отримуємо:

$$F = \frac{400 \cdot 36 \cdot 1.5 \cdot 1.2}{0.8} = 32400 \text{ (Лм)}.$$

(4.6)

Тепер визначимо кількість світильників, необхідну для освітлення приміщення за наступною формулою (5.7).

(4.7)

$$N = \frac{F}{E_л} = \frac{32400}{3200} = 10.$$

Таким чином, щоб забезпечити світловий потік F =32400 Лм необхідно використовувати 10 світильників.

Електрична потужність одного світильника Crystal 218 LED = 36 Вт.

Загальна потужність усієї освітлювальної системи в приміщенні лабораторії:

(4.8)

$$W_{заг} = W_п \cdot N = 36 \cdot 10 = 360\ Вт$$

Висновок: при достатньому природному освітленні (світлий час доби, ясна погода), кількості і сумарної площі світлових прорізів досить для забезпечення необхідної освітленості робочого приміщення.

У випадку недостатності природного освітлення необхідно задіяти джерела штучного освітлення (розрахунок показало, що досить мати 10 світильників Crystal 218 LED і чистими плафонами.

4.6. | Висновок

Для покращення освітлення проведена модернізація штучного освітлення та запропоновані більш ефективніші лампи (10 ламп Crystal 218 LED), світловий потік кожної з яких складає 3200 Лм, що дозволяє досягти оптимального значення необхідної освітленості робочого місця Ен=400Лк, строк служби ламп при цьому 50 тисяч годин. До того ж вони стійкі до механічних пошкоджень та низьковольтні, а отже - безпечні. Крім цього не важко помітити, що при застосуванні LED ламп зменшується навантаження на мережу.

Також для покращення мікроклімату в приміщенні лабораторії та підвищення продуктивності праці інженера-дослідника необхідно встановити кондиціонер та використовувати зволожувачі повітря.

Розділ 5. Охорона навколишнього середовища

Тема «Сталий розвиток зеленої авіаційної промисловості на шляху до комплексної системи підтримки»

Анотація: Сталий розвиток стає все більш важливим для зеленої авіаційної промисловості. Щоб сприяти сталому розвитку зеленої авіаційної промисловості, дослідники в усьому світі намагалися досліджувати та експериментувати з новими ініціативами. У цьому дослідженні досліджуються тенденції розвитку зеленої авіаційної промисловості з допомогою системи аналізу даних на основі библиометрии. Візуалізація кластера, часового поясу та тимчасової шкали використовується для визначення поточних тенденцій розвитку зеленої авіаційної промисловості в рамках трьох тем: шум, вплив на навколишнє середовище та екологію. У відповідь на екологічні проблеми слід створити комплексну систему підтримки зеленої авіаційної промисловості, яка передбачає залучення зацікавлених сторін, включаючи бізнес-стратегії, інноваційні технології, екологічну політику і державну підтримку. У цьому дослідженні оцінюється інноваційний потенціал та потенціал сталого розвитку зеленої авіаційної промисловості. Потім розглядаються ролі вдосконалення стратегії інтеграції технологій, підтримки політики і участі громадськості у формуванні інтегрованої системи підтримки. Схема цієї системи може виявитися корисною для сталого розвитку зеленої авіаційної промисловості.

5.1. | Введення

Неухильне розвиток глобалізації призвело до різкого зростання попиту на транспорт, що робить сталий розвиток авіаційної промисловості все більш актуальним (Хиннен та ін., 2017; Кляйнер, 2007; Лассен, 2010). Екологічні проблеми (наприклад, шум і забруднення повітря), пов'язані з авіаційною промисловістю, наростають (Чуї та ін., 2018; Фальк і Хагстен, 2020; Махашабде та ін., 2011). Частка людей, які терміново госпіталізуються або помирають від серцево-судинних захворювань, збільшується в середньому на 3,5%, коли шум в аеропорту збільшується на десять децибел (Коррейя та ін., 2013). Авіаційна промисловість відповідає за 13% викидів вуглецю з усіх джерел транспорту, а чистий ефект викидів оксиду азоту, який збільшує концентрацію озону, становить приблизно 24% (Чи та ін., 2009). За оцінками, щорічні темпи зростання діяльності авіаційної промисловості складуть 5%, і, отже, викиди вуглекислого газу до 2050

року будуть у сім-вісім разів вище, ніж у 1990 році (Міжнародна асоціація повітряного транспорту, 2009). Ці проблеми збільшують зовнішні витрати авіаційної промисловості і впливають на сталий розвиток суспільства та економіки. Зелена авіаційна промисловість з'явилася як відповідь на ці екологічні проблеми.

Для авіаційної промисловості "зелений" означає екологічно чистий режим розвитку, в якому орієнтовані на енергозбереження та охорону довкілля, намагаючись зменшити негативні зовнішні ефекти авіаційної промисловості (Хагман та ін., 2015). Сталий розвиток зеленої авіаційної промисловості зосереджено на захисті екологічного середовища і раціональному використанні ресурсів, що створює умови для майбутнього розвитку авіаційної промисловості (Весперманн і Виттмер, 2011). На екологічну авіаційну галузь впливають технологічні розробки та екологічна політика (Карагианнис та ін., 2019; Чжан та ін., 2020). На це також впливають бізнес-стратегії, які поступово сприяють розвитку зеленої авіаційної промисловості в інтересах кращого майбутнього (Перес-Вальс та ін., 2015; Шеу, 2014).

Наукові публікації в області зеленої авіаційної промисловості різко зросли за останнє десятиліття, як з точки зору кількості спеціалізованих журналів, так і статей на журнал (Чен, 2017; Хаберл та ін., 2019). Між тим, завдяки досягненням в області технологій та методології, різноманітність тем в рамках досліджень в галузі зеленої авіаційної промисловості також зросла. У бібліометричного області дослідники вже давно використовують дані публікацій для вивчення закономірностей і тенденцій наукового співробітництва (Фаррух та ін., 2020; Ферассо та ін., 2020; Филсер та ін., 2020; Кабонго, 2020). Однак библиометрические дослідження в області досліджень зеленої авіаційної промисловості залишаються обмеженими. Для кращого розвитку зеленої авіаційної промисловості та сприяння відповідним дослідженням у галузі зеленої авіаційної промисловості важливо виявити закономірності її розвитку і майбутні тенденції. Застосовуючи библиометрию в області зеленої авіаційної промисловості, створюється система наукового аналізу, а саме система аналізу даних на основі библиометрии (BDAS), для вивчення тенденцій зеленої авіаційної промисловості та забезпечення сталого розвитку цієї галузі. Він може запропонувати нові дослідницькі ідеї і стати важливою довідкою для вчених.

У цьому дослідженні BDAS розроблена для вивчення тенденцій в області екологічно чистої авіаційної промисловості і прискорення сталого розвитку цієї галузі. Визначено три ключові теми про тенденції розвитку зеленої авіаційної

промисловості, і для майбутнього розвитку зеленої авіаційної промисловості впроваджено уявна інтегрована система підтримки, що включає бізнес-стратегії, інноваційні технології, екологічну політику та участь громадськості. Інша частина цього дослідження організована наступним чином. BDAS побудований для визначення тенденцій розвитку зеленої авиационною промисловості у розділі 2 представлены результати використання BDAS для демонстрації поточного розвитку зеленої авіаційної промисловості, а в розділі 3 пропонується комплексна система підтримки. Інновації і стійкість зеленої авіаційної промисловості спочатку оцінюються в розділі 4, а потім обговорюються ключові елементи інтегрованої системи підтримки. Нарешті, висновки представлені в розділі 5.

5.2 | МЕТОДОЛОГІЯ

У зв'язку з швидким зростанням досліджень в області "зеленої" авіаційної промисловості визначити найбільш актуальні напрямки досліджень і тенденції розвитку непросто. Тому вивчення літератури має важливе значення для визначення найбільш значимих напрямків досліджень, особливо в галузі транспортних систем. Інструменти бібліометричного аналізу можуть досліджувати процес розвитку зеленої авіаційної промисловості і майбутні напрямки. У цьому дослідженні BDAS побудований для вивчення взаємозв'язку між науковими працями, опублікованими в області зеленої авіаційної промисловості (в розбивці по роках і вибраними ключовими словами).

5.2.1 | BDAS

На рисунку 1 показана організація пропонованих BDAS, яка передбачає використання двох програмних пакетів: Web of Science (WoS) і CiteSpace. Наведено приклад процедури пропонованої системи, що включає модулі збору даних, аналізу даних і візуалізації. У модулі збору даних більша частина інформації отримана з первинного огляду літератури. Модуль аналізу даних використовує простір цитат для візуалізації та аналізу наукових публікацій для виявлення нових тенденцій. Вимоги до формату даних CiteSpace засновані на стандарті бази даних WoS і оновлюються з урахуванням змін у форматі даних в базі даних Інституту наукової інформації (ISI) (Чен, 2017). Модуль візуалізації надає спеціальне опис ходу виконання, яке може бути використане для виконання аналізу множинних, розподілених за часом і динамічних складних мереж, а також для визначення гарячих точок і тенденцій у певній галузі.

Велика частина відповідної інформації отримана модулем збору даних з вихідної бази даних літератури, ISI WoS, яка була обрана з-за її широкого доступу до основним дослідницьким баз даних. У BDAS пошук ключових слів здійснюється з допомогою "Xi + Yi" з розширеним пошуком, де "Xi" - це ключові слова, такі як екологічність, екологічність, емісія повітряних суден і шум, а "Yi" - ключові слова, що відносяться до авіаційної промисловості, авіакомпаніям і повітряному транспорту. Всього було завантажено 616 статей, пов'язаних із зеленою авіаційною промисловістю. Після ретельного видалення не відносяться до справи і дублюючих статей первісна запис була уточнена до 539 статей. CiteSpace обробив дані за період 1990-2019 років (публікації до кінця 2019 року), побудував мережі методом обрізки "Мінімального сполучного дерева" і вибрав ключові слова у якості типів вузлів для визначення діапазонів в заданих межах. Коефіцієнти подібності Jaccard були обрані в якості заходів подібності посилань, і для вилучення міток кластера в цьому дослідженні був обраний алгоритм TF*IDF. Аналіз на основі ключових слів, тип аналізу співпадінь, має важливе значення для розуміння динаміки розвитку знань. У модулі візуалізації були використані різноманітні вистави для мережевого аналізу, такі як кластер і тимчасова шкала, що розкривають тенденції з літератури по зеленій авіаційної промисловості.

5.2.2 | Наукове картографування

Наукове картографування є важливою бібліометричного технологією для вивчення концептуальної структури конкретної галузі досліджень (Кобо та ін., 2011). Цей підхід дозволяє відстежувати конкретну область і розмежовувати області досліджень для виявлення становлення і розвитку. Виходячи з знань предметної області, наукове картографування описує процес і структуру розвитку наукових знань, виявляючи складні взаємозв'язки між мережевою структурою, групами знань і загальною еволюцією (Чен та ін., 2010). Оскільки інструменти візуалізації CiteSpace можуть визначати відмінні особливості спектра, вони широко використовуються в дослідженнях. Теорія наукового розвитку Куна є однією з моделей, яка обґрунтувала застосування простору цитування для відображення наукових знань, що базується на функції прогнозування простору цитування і може прогнозувати майбутнє стан певної галузі досліджень (Чен, 2017).

Кун (2012) стверджує, що науковий розвиток-це процес історичної наукової революції. Наука не характеризувалася відповідною парадигмою на донаучной стадії, але для наукової революції необхідна нова парадигма, оскільки наука розвивається шляхом постійного вирішення проблем. При входженні в нову парадигму це перетворилося в нову норму. Отже, застосовуючи простір

цитування способу наукового розвитку, структура карти знань, створена накопиченням, поширенням і перетворенням кластерів наукового цитування, вказує на передові області і підкреслює еволюцію знань.

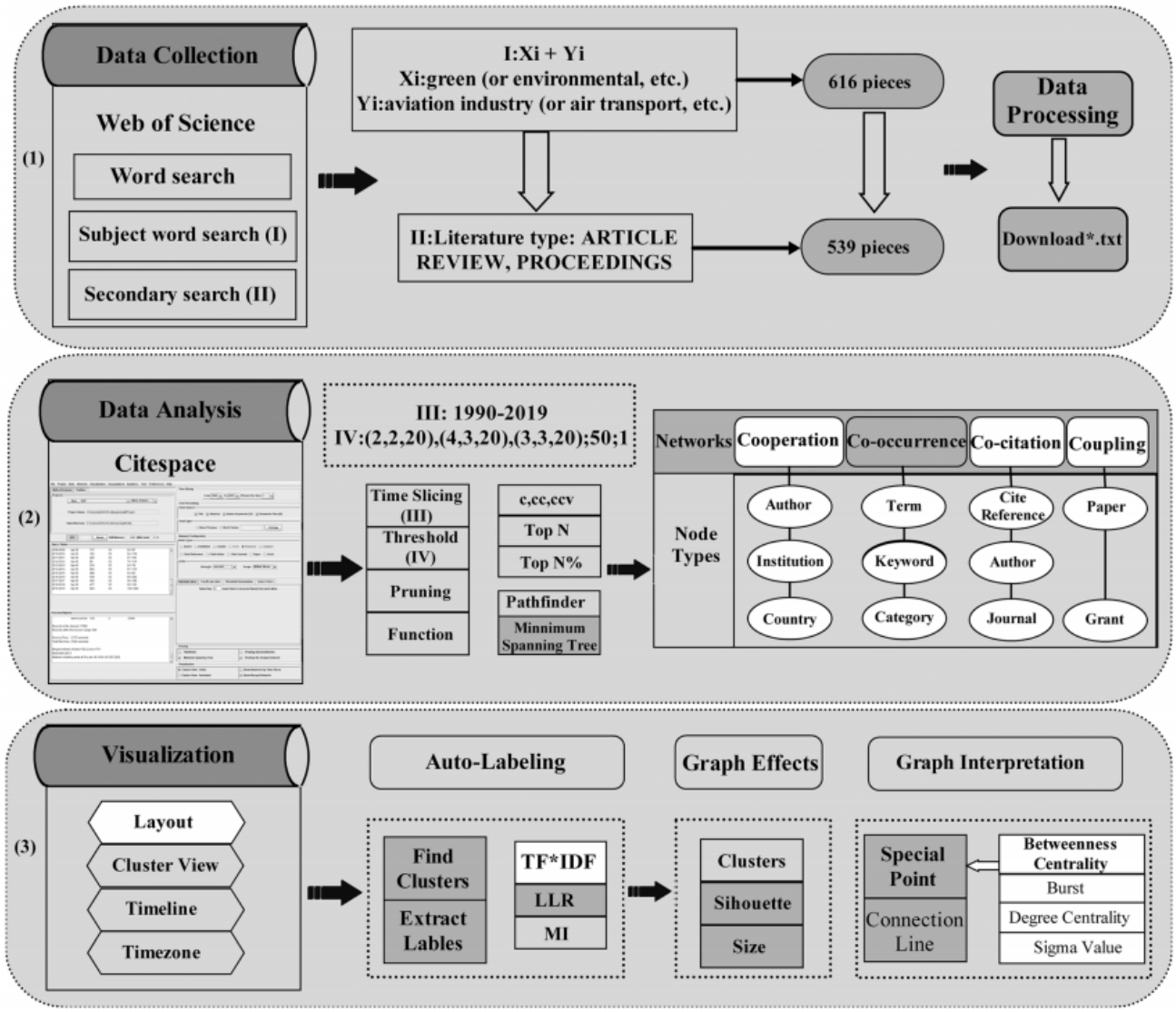


Рис.1 Процес аналізу літератури за допомогою системи аналізу библиометрических даних (BDAS)

Простір цитування може визначати кластери спільного цитування в літературі і відстежувати тенденції досліджень. Основними технологіями CiteSpace є алгоритми спектральної кластеризації та вибору функцій, що включають візуалізацію результатів для розуміння тенденцій досліджень та їх еволюції. Подання тимчасової шкали CiteSpace дозволяє візуалізувати і ідентифікувати досягнення зеленої авіаційної промисловості в різні роки публікації. Кластери в CiteSpace використовуються для визначення траєкторій для моделювання тенденцій розвитку зеленої авіаційної промисловості.

5.2.3 | Кластеризація даних і аналіз візуалізації

Основною особливістю CiteSpace є візуалізація результатів пошуку, яка дозволяє нам розуміти пов'язані тенденції і зміни, у той час як кластерний аналіз CiteSpace виявляє взаємозв'язки між ключовими словами з відповідних статей. На рисунку 2 показаний графік Подання кластера" , створений CiteSpace.

Двома важливими показниками в CiteSpace для вимірювання характеристик загальних структурних мереж є модульність і міра силуету. Відносно високе значення модульності (0,9352) показує розумно розділені кластери в мережі, як показано на рисунку 2. Міра силуету може перевірити, правильно чи згруповані дані. Якщо значення міри силуету наближається до 1, дані правильно кластеризованы; якщо значення близьке до -1, дані неправильно кластеризованы. Дані знаходяться на межі двох природних кластерів, якщо міра силуету близька до 0. Показник силуету вказує на те, що кластерна однорідність (0,4704) тенденцій зеленої авіаційної промисловості відносно висока.

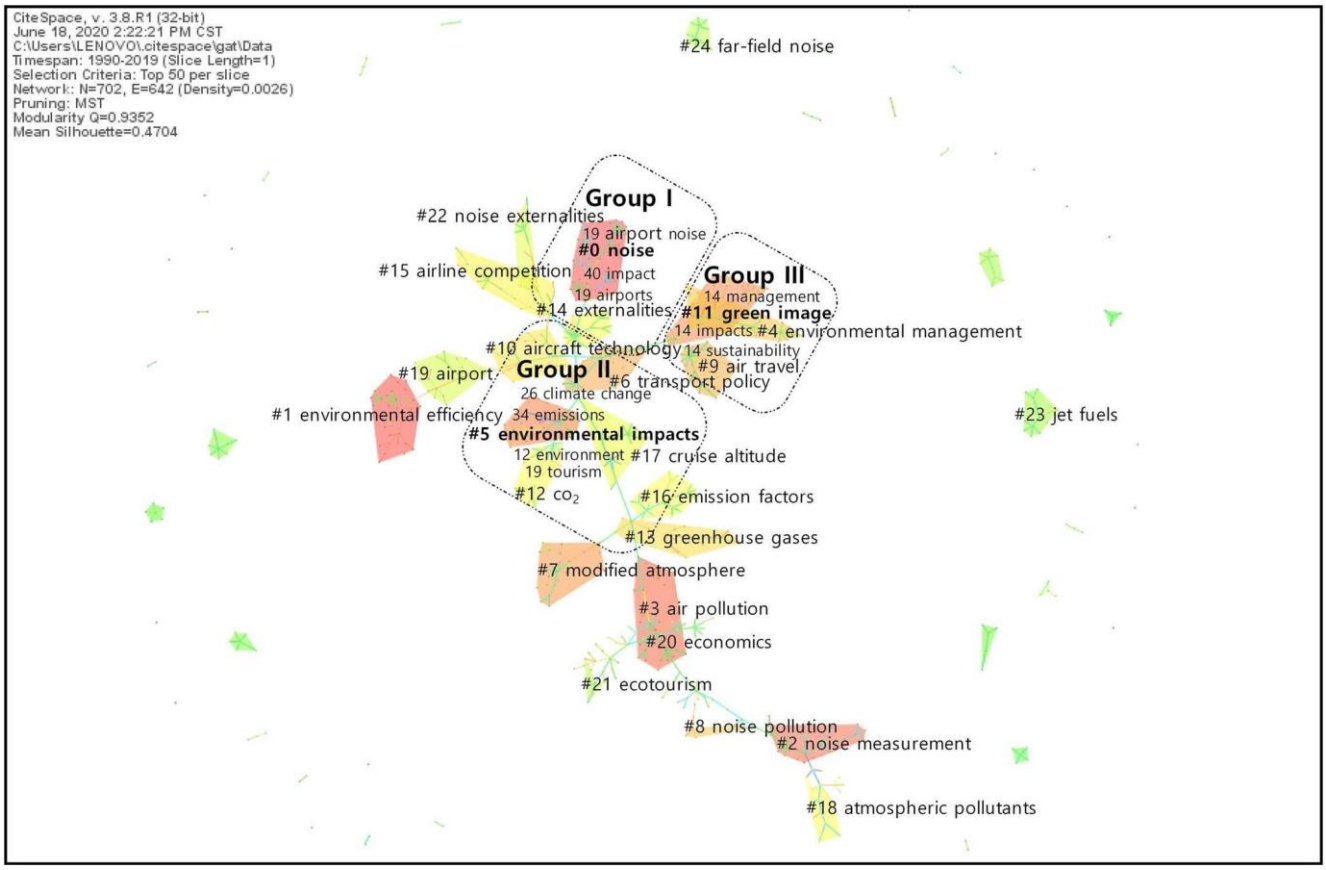


Рис. 2 Ключові слова спільне поява і кластерний подання відповідних досліджень в області зеленої авіаційної промисловості

ТАБЛИЦЯ 1 Огляд вибраних кластерів і пов'язаних з ними ключових слів

Cluster ID	Size	Silhouette	Mean (year)	Label (LLR)	Keywords in cluster
0	35	0.812	2001	Noise	40 impact; 19 airports; 19 aircraft noise
5	26	0.756	2005	Environmental impacts	34 emissions; 26 climate change; 19 tourism; 12 environment
11	23	0.739	2009	Green image	14 management; 14 impacts; 14 sustainability

З допомогою автоматичної маркування знайдено 82 кластера. Вимірювання силуету і розміру застосовуються для визначення найбільш відповідних кластерів. Виділяються три кластери: "шум", "вплив на навколишнє середовище" і "зелене зображення", які мають високі силуети (ID: 0, 5 і 11; силуети: 0,812, 0,756 і 0,739) і відносно великі за розміром (35, 26 і 23). Оскільки ключові слова трьох кластерів різні, вони розглядаються як репрезентативні для різних ключових елементів зеленої авіаційної промисловості. Три кластери та пов'язані з ними ключові слова об'єднані в три групи: група I–III, перераховані в таблиці 1 (також показано на рисунку 2). У цій таблиці числа, що передує ключовими словами, є частотоюслів.

На малюнках 3 і 4 показані графіки "Вид тимчасової шкали" і "Вид часового поясу" відповідних досліджень для зеленої авіаційної промисловості. На графіку "Подання тимчасової шкали" показано кластери разом з горизонтальними часовими лініями (Чен, 2017). Кожен кластер відображається зліва направо. Легенда про час публікації відображається у верхній частині подання. Графік Подання часового поясу" підкреслює часові закономірності між дослідним фронтом і його інтелектуальною базою, включаючи масив вертикальних смуг в якості часових поясів (Чен, 2006). Часові пояси розташовані в хронологічному порядку зліва направо. Алгоритм компоновки являє собою модифікований алгоритм вбудовування пружини, так що горизонтальне переміщення елемента обмежено його часовим поясом, але його вертикальне переміщення повністю визначається його зв'язками з елементами в інших годинних поясах (Чен, 2006). Мета полягає в тому, щоб зробити спеціальність легко впізнаваною. В цілому, "Подання тимчасової шкали" відображає кластери разом з горизонтальними часовими лініями, в той час як "Уявлення часового поясу" показує кластери з вертикальними смугами в якості часових поясів.

Виходячи з цих цифр, кластери змінюються з плином часу, причому три групи виникають у різні періоди: за "Шумом" слід "Вплив на навколишнє середовище", а потім "екологію", що свідчить про розвиток зеленої авіаційної промисловості.

На основі цього початкового аналізу визначено три етапи "зеленої" авіаційної промисловості.

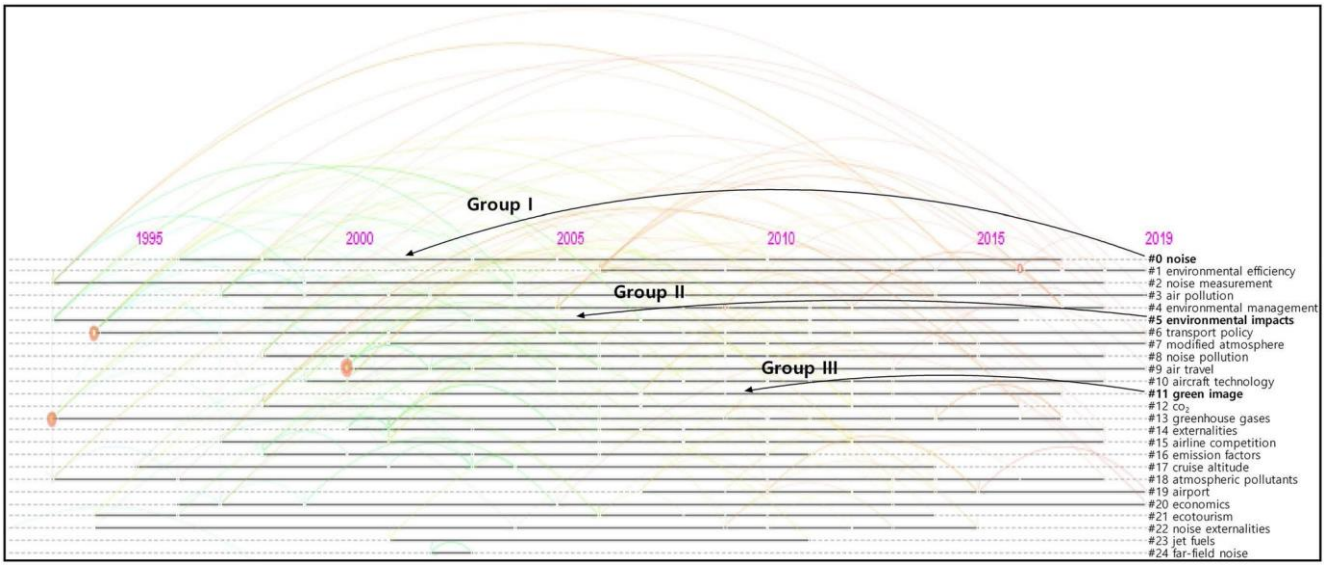


Рис. 3 Кластери системи аналізу даних на основі библиометрических даних (BDAS) з поданням тимчасової шкали відповідних досліджень в області екологічно чистої авіаційної промисловості

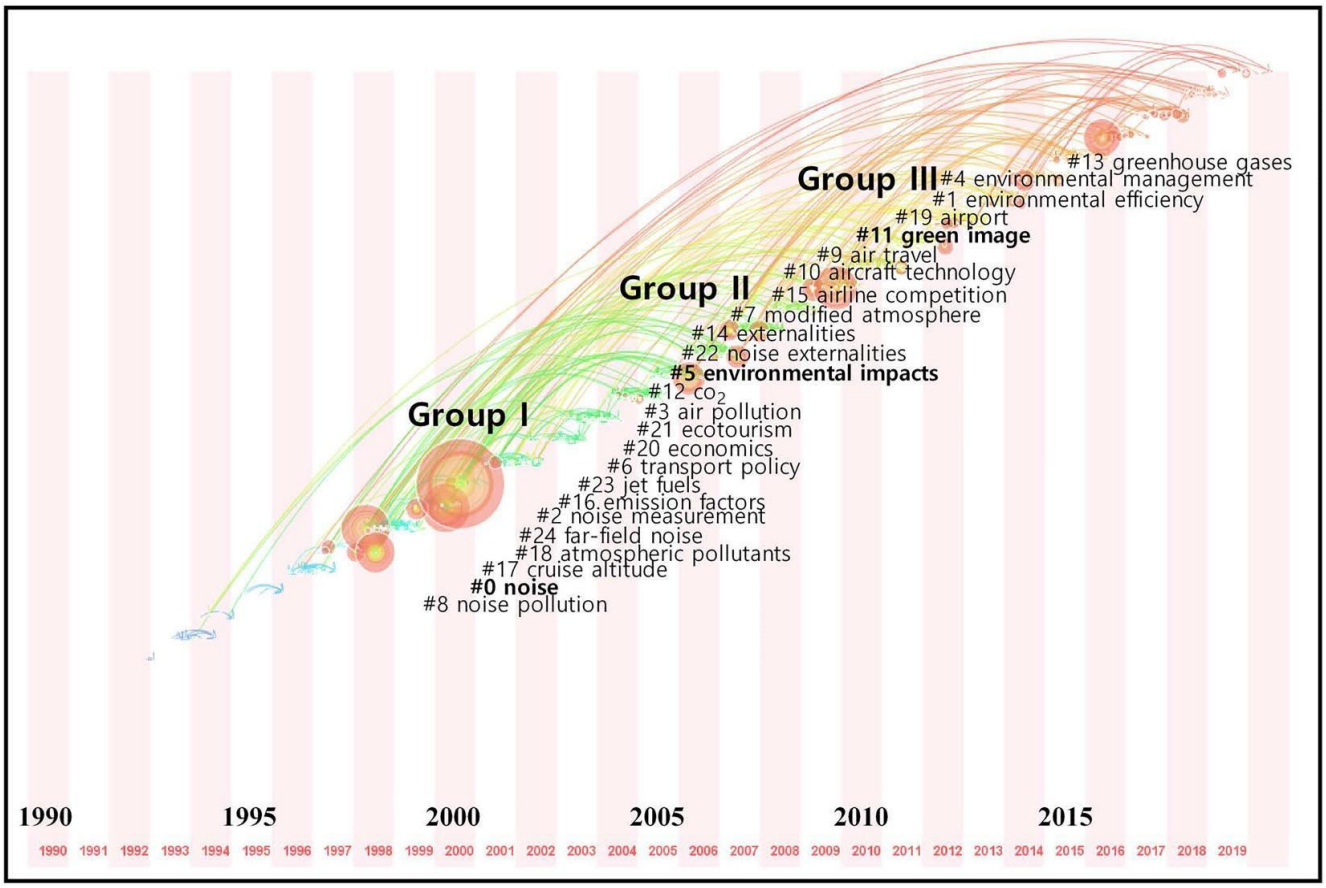


Рис. 4 Кластери системи аналізу библиометрических даних (BDAS) з поданням часових поясів відповідних досліджень в області екологічно чистої авіаційної промисловості

5.3 | Тенденції розвитку зеленої авіаційної промисловості

Для сприяння сталому розвитку "зеленої" авіаційної промисловості і внесення вкладу в літературу в галузі "зеленої" авіаційної промисловості розробляється BDAS, що включає модулі збору, аналізу і візуалізації даних. За даними BDAS, розкриваються тенденції розвитку зеленої авіаційної промисловості. На карті візуальних знань (рис. 2) виділяються три окремих кластера: "шум", "вплив на навколишнє середовище" і "екологію", які є основою для подальшого аналізу. На основі результатів використання BDAS демонструється поточний розвиток зеленої авіаційної промисловості, і в цьому розділі пропонується інтегрована система підтримки.

5.3.1 | Посилення впливу авіаційної промисловості на навколишнє середовище

З розвитком суспільства і економіки попит на комерційну авіацію зріс, що призвело до посилення впливу на навколишнє середовище, включаючи шум, якість повітря і зміна клімату (Чуї і Чи, 2017; Фридль, 1999; Грампелла та ін., 2017; Шефер і Вайц, 2014).

Авіаційний шум є найбільш очевидним впливом діяльності авіаційної промисловості на навколишнє середовище і був основним джерелом скарг на аеропорти, що призвело до сильного протидії спільноти більшості проектів розширення аеропортів (Брістоу і Уордман, 2006; Фейтельсон та ін.; 1996; Захарі та ін., 2010). Розділ 2 вказує, що "шум" є найбільш раннім у трьох різних кластерах. Шум, що вимірюється в децибелах, зазвичай масштабується, щоб повідомити про чутливості людського сприйняття до різних частотах. Скоригований по тону сприйманий рівень шуму і зважена шкала є двома зазвичай використовуваними частотно-зваженими шкалами (Махашабде та ін., 2011). Перший з них враховує сприйняття людиною чистих тонів і інших спектральних неоднорідностей і прийнятий у проектуванні повітряних суден та стандарти сертифікації шуму Міжнародної організації цивільної авіації (ІКАО). Останній вимірює різні частоти у відповідності з чутливістю людського вуха і, як правило, використовується для створення карт зон впливу шуму та оцінки впливу шуму. Показники шуму повітряних суден підрозділяються на показники окремих подій і сукупні показники (Махашабде, 2009). Показники однієї події оцінюють прямий

вплив одного руху повітряного судна, в той час як сукупні показники враховують довгострокове вплив авіаційного шуму.

Викиди, вироблені авіаційними двигунами, містять діоксид вуглецю, водяна пара, оксиди азоту, оксиди сірки, монооксид вуглецю, тверді частинки та інші забруднювачі (Гарді та ін., 2016; Чи та ін., 2010; Зельвелинг та ін., 2011). Викиди вуглецю складають близько 70% таких забруднюючих речовин, а на частку водяної пари припадає трохи менше 30%, в той час як інша частина відповідає менше 1% від загального обсягу викидів (Федеральне управління цивільної авіації, 2005). Багато з цих викидів здійснюють прямий або опосередкований негативний вплив на здоров'я. Традиційно аналіз впливу авіації на якість повітря спрямований на викиди при посадці і зльоті нижче 3000 футів (Андерсон та ін., 2007). Проте деякі дослідження показали, що викиди на етапі круїзу (більше 3000 футів) можуть становити значну частину загального впливу якості повітря на здоров'я (Махашабде та ін., 2011). За оцінками, передчасна смертність в результаті глобальних викидів в результаті польотів повітряних суден складає значну частку від загального впливу авіаційної промисловості на здоров'я (Барретт, 2010). Майбутні вимірювання впливу авіації на якість повітря повинні містити загальну інформацію про викиди в ході польотів. Міжурядова група експертів зі зміни клімату (МГЕЗК) визначає радіаційний вплив як "вимірювання впливу фактора на зміну енергетичного балансу введення і виведення в атмосферної системі Землі" (Соломон та ін, 2007). Ефект потепління мається на увазі позитивним радіаційним впливом, а ефект охолодження мається на увазі негативним. У Четвертому оціночному доповіді МГЕЗК підраховано, що в 2005 році радіаційний вплив дозвукових авіації становила приблизно 3% від загального радіаційного впливу діяльності людини (Соломон та ін, 2007).

Стикаючись з вищевказаними екологічними наслідками, зацікавлені сторони авіаційної промисловості шукають стратегії, які збалансують екологічні та економічні інтереси.

Авіаційний шум був першим зовнішнім фактором, який регулювався ІКАО в 1960-х роках (ІКАО, 2005). До 1980-м рокам ІКАО розробила стандарти викидів повітряних суден з допомогою Стандартів та рекомендованої практики (SARPs) (ІКАО, 2006). За останні кілька десятиліть було зроблено безліч заходів по боротьбі з впливом авіації на зміну клімату. ІКАО (2008) заснувала Групу з міжнародної авіації та зміни клімату для надання керівних вказівок з розгляду впливу комерційної авіації на зміну клімату. Європейська комісія видала наказ, що вимагає включення авіації в Систему торгівлі викидами Європейського союзу (Ангер і Келлер, 2010). У Сполучених Штатах Агентство з охорони

навколишнього середовища (2008) оголосило про попередньому повідомленні про правилі, в якому пропонується висловити зауваження громадськості про вплив регулювання парникових газів у відповідності з Законом про чистому повітрі. Агентство також скасував правило, яке вимагало обов'язкової звітності про викиди парникових газів в великих галузях промисловості, включаючи авіацію, для збору даних для прийняття рішень у майбутньому. Оскільки прогнозовані темпи зростання комерційної авіації в найближчі 20-25 років складають приблизно 5% в рік, вплив авіації на навколишнє середовище може бути більш значним порівняно з іншими галузями промисловості (Мец та ін., 2007).

Для поліпшення екологічних показників авіаційної промисловості необхідно добре розуміти компроміси між бізнес-стратегіями, інноваційними технологіями, екологічною політикою і впливом на навколишнє середовище (наприклад, шум, якість повітря і зміна клімату). Комітет з авіаційної охорони навколишнього середовища ІКАО самостійно займався наслідками авіаційного шуму та викидів за допомогою таких заходів, як стандарти сертифікації авіаційного шуму (ІКАО, 2005). Регулюючі рішення засновані на показниках економічної ефективності, які оцінюють скорочення викидів повітряних суден і рівні шуму щодо очікуваних витрат на реалізацію рішень (Махашабде та ін., 2011). Однак у цього підходу є деякі явні недоліки. Екологічні вигоди чітко не оцінені, а невизначеності, пов'язані з аналізом регулювання, обмежені. Необхідний всебічний аналіз для оцінки компромісів між впливом на навколишнє середовище і економічними витратами для досягнення сталого розвитку зеленої авіаційної промисловості.

5.3.2 | Зелений імідж для репутації авіакомпанії

У зв'язку з тим, що вплив авіаційної промисловості на навколишнє середовище привертає все більшу увагу, все більше і більше авіакомпаній зосереджуються на створенні "зеленого " іміджу" для збереження своєї репутації. У 1990-х роках почали з'являтися "зелені" споживачі, коли люди почали усвідомлювати наслідки деградації навколишнього середовища (Хван і Люі, 2020). Поведінка споживачів відображає це явище. Споживачі, які турбуються про екологічні проблеми (наприклад, забруднення повітря та зміни клімату), концентруються на усунення природного збитку, купуючи екологічно чисті продукти/послуги. Імідж бренду є найбільш значущим чинником, що визначає рішення про покупку екологічно чистих продуктів/послуг (Дирсехан і Куртулус, 2018). Концепція зеленого іміджу відноситься до сприйняття компанії у свідомості клієнтів, що пов'язує компанію з екологічними зобов'язаннями (Майєр та ін, 2012). Багато компанії прагнули створити зелений імідж, щоб скористатися його перевагами для досягнення успіху в бізнесі. Продукти/послуги компанії можуть бути розпізнані від

конкурентів зеленим зображенням завдяки його символічної ролі в уявленні бренду компанії (Хван і Хван, 2019). Таким чином, імідж зеленого бренду глибоко впливає на купівельну поведінку споживачів.

Створення зеленого іміджу також важливо для авіакомпаній (Хагманн та ін., 2015; Лу і Ван, 2018). Авіакомпанія Asiana Airlines, велика авіакомпанія Південної Кореї, що займається діяльністю з енергозбереження за рахунок скорочення викидів, отримала нагороди Корейського індексу стійкості за свою програму управління навколишнім середовищем. United Airlines займалася екологічним менеджментом і домоглася значного прогресу в області інновацій в літаках, що дозволило підвищити паливну економічність на 20%. Авіакомпанія також встановила топливосберегающие крильця, знижують середні викиди вуглекислого газу на 600 тонн для кожного літака в рік і підвищують паливну економічність майже на 2%. Ця авіакомпанія отримала золоту нагороду екологічної авіації "Авіакомпанія року", надану журналом World aviation industry за її зусилля по охороні навколишнього середовища (Хван і Хван, 2019). Багато авіакомпанії розробили екологічно чисті стратегії для створення зеленого іміджу. Авіакомпанія з "зеленим" іміджем докладає зусиль для реалізації бізнес-стратегій щодо поліпшення екологічних показників. Таким чином, метою авіакомпанії з "зеленим" іміджем є усунення впливу авіаційної промисловості на навколишнє середовище

Оскільки авіапасажири все більше турбуються про екологічні проблеми і все більше прагнуть купувати екологічно чисті продукти/послуги, створення зеленого іміджу стає необхідним для авіакомпаній, щоб отримати конкурентну перевагу і чітко виділитися серед конкурентів. "Стаючи зеленими", вони можуть значно підвищити свою репутацію, передаючи елементы навколишнього середовища в бренд.

5.3.3 | Майбутнє розвиток зеленої авіаційної промисловості

Очікується, що увага до екологічним іміджу для підвищення репутації авіакомпаній і зниження впливу авіаційної промисловості на навколишнє середовище буде продовжуватися, оскільки концепція "екологічності" приймається все більшим числом людей у всьому світі. Грунтуючись на існуючих технологіях, очікується, що екологічна авіаційна промисловість буде все більше орієнтуватися на захист навколишнього середовища (Паркер, 2009; Ву та ін., 2018). У майбутньому може бути розроблена інтегрована система підтримки зеленої авіаційної промисловості, в якій беруть участь авіакомпанії, інженери, уряди і громадськість. У відповідь на екологічні виклики необхідна система

підтримки розвитку зеленої авіаційної промисловості. Отже, поява інтегрованої системи підтримки зеленої авіаційної промисловості стає центральною проблемою для досягнення сталого соціального розвитку. Запропонована комплексна система сприяння сталому розвитку зеленої авіаційної промисловості показана на рисунку 5.

Малюнок 5. Хоча деякі авіакомпанії, уряду і вчені запропонували стратегії досягнення сталого розвитку зеленої авіаційної промисловості, повністю функціональні інтегровані системи підтримки зеленої авіаційної промисловості поки недоступні. У запропонованій комплексній системі підтримки зеленої авіаційної промисловості докладно обговорюються майбутні моделі використання енергії в авіаційному секторі та пов'язані з цим екологічні наслідки. Попередні обговорення, проведені в літературі з видобутку корисних копалин і пов'язаних з цим питань розвитку, свідчать про нагальну необхідність комплексної системи підтримки зеленої авіаційної промисловості, оскільки сталий розвиток зеленої авіаційної промисловості в найближчому майбутньому бажано для створення чистого, високоефективного суспільства.

В майбутньому інтегровані системи підтримки зеленої авіаційної промисловості повинні передбачати бізнес-стратегії, інноваційні технології, екологічну політику, ресурси та екологічні характеристики, які в сукупності враховують економічні, екологічні та соціальні потреби сталого розвитку. Екологічна стійкість є найбільш фундаментальною потребою. Розумні, цілеспрямовані і безперервні бізнес-стратегії є важливою частиною системи, оскільки відповідні бізнес-стратегії можуть підвищити ефективність авіаційної промисловості. Технологічні досягнення в галузі авіації навряд чи можуть бути досягнуті без допомоги інженерів, яких можна було б заохочувати за допомогою належних бізнес-стратегій. Державна підтримка може бути корисною для розвитку комплексної системи. Належна екологічна політика може сприяти технологічному прогресу та зміни ринкового попиту в цілях підтримки стійкої конкуренції. Активна громадська підтримка може допомогти в реалізації нових бізнес - стратегій, інноваційних технологій та екологічної політики. Іншими словами, тенденції розвитку зеленої авіаційної промисловості будуть пов'язані з цілеспрямованими бізнес-стратегіями, інноваційними технологіями, екологічною політикою і державною підтримкою, з упором на створення комплексної системи підтримки. Отже, інтерактивне управлінське, технологічне, політичне спрямування за участю громадськості є неминучою тенденцією розвитку зеленої авіаційної промисловості.

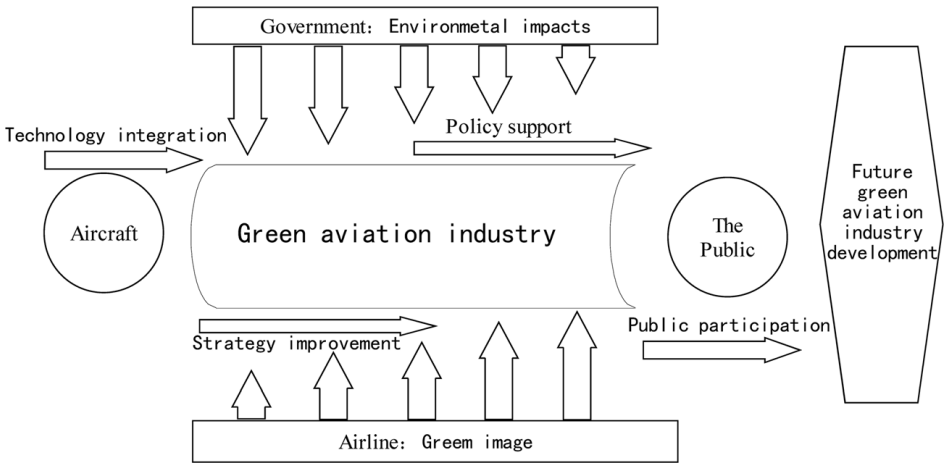


Рис. 5 Інтегрована система підтримки зеленої авіаційної промисловості

5.4 | Дискусія

Хоча тенденції розвитку зеленої авіаційної промисловості вже розглядалися в літературі, потрібен подальший аналіз для сприяння сталому розвитку. У цьому розділі спочатку оцінюється поточний стан зеленої авіаційної промисловості. Потім забезпечується кілька ключових елементів інтегрованої системи, що включає бізнес-стратегії, інноваційні технології, екологічну політику та участь громадськості.

5.4.1 | Оцінка "зеленої" авіаційної промисловості

Інновації та сталий розвиток є важливими темами для зеленої авіаційної промисловості. Перше є основою розвитку, а друге-основою для майбутнього просування.

5.4.1.1 | Інновації

Будучи рушійною силою соціального розвитку, інновації можуть забезпечити ефективність та захист навколишнього середовища в галузі авіаційної промисловості. Починаючи з Роджерса Еверетта (1995), дифузія розглядалася як інноваційний процес, який призводить до перетворень у функціях і структурі соціальних систем. Однак поширення не є лінійним, а відповідає інтерактивного процесу обміну інформацією і переговорів. На інновації впливає попит (Едлер, 2013). При розгляді шуму, впливу на навколишнє середовище і зеленого іміджу необхідні інновації для створення зеленого іміджу в зеленій авіаційної промисловості.

Інновації можуть сприяти швидкому розвитку зеленої авіаційної промисловості. Як наукомістка галузь, авіація повинна розглядати інноваційні технології в якості центрального елемента для сприяння переходу до режиму

розвитку "зеленої" авіаційної промисловості та оптимізації структури промисловості. Підвищення рівня науково-технічних інновацій сприяє підвищенню безпеки, ефективності управління, контролю витрат і рівня обслуговування всієї галузі, тим самим підвищуючи основну конкурентоспроможність. Для реалізації розвитку зеленої авіаційної промисловості необхідно надавати важливе значення науковим і технологічним інноваціям, а також надавати політичну підтримку розробки та просування нових технологій. Тому необхідні численні зусилля для сприяння інноваціям для майбутньої зеленої авіаційної промисловості.

5.4.1.2 | Стійкість

Сталий розвиток є найважливішою потребою для майбутнього соціального розвитку і основною проблемою зеленої авіаційної промисловості (Лоу та ін., 2012). Екологічні, технологічні і соціальні зміни мають важливе значення для досягнення майбутньої стійкості. Авіаційна промисловість викликає шум, забруднення навколишнього середовища і викиди вуглекислого газу, все це перешкоджає стійкості зеленої авіаційної промисловості. Однак стійкість пов'язана не тільки з екологічними проблемами, але і з соціальним та економічним розвитком, оскільки забруднення навколишнього середовища та зміна клімату істотно впливають на здоров'я, добробут, культуру і засоби до існування. Таким чином, для досягнення режиму "зеленої" авіаційної промисловості необхідна довгострокова перспектива з переходом до нових бізнес-стратегій, інноваційних технологій, екологічної політики та енергетиці.

Економічна стійкість зеленої авіаційної промисловості забезпечує гарантію екологічної і соціальної стійкості, що, в свою чергу, сприяє економічної стійкості. Для досягнення стійкості зеленої авіаційної промисловості в першу чергу слід приділяти увагу безпеці. Тільки за умови безпечного транспортування слід вживати зусиль щодо зниження впливу на навколишнє середовище. Стратегії розвитку авіакомпаній повинні бути спрямовані не тільки на підвищення прибутку, але і на реалізацію соціальної відповідальності, створення позитивного іміджу і, таким чином, забезпечення сталого розвитку. Авіакомпанії повинні побудувати "зелений" спосіб життя і постійно прагнемо зменшити вплив на навколишнє середовище авіації в різних способів, таких як розширення використання літальних апаратів з підвищеною економії палива продуктивністю і низьким рівнем шуму, що виключає літальними апаратами з високою витратою палива, зміцнення і розвиток передових енерго-і ресурсозберігаючих технологій, підвищення ефективності використання палива, зниження забруднення навколишнього середовища, приділяючи пильну увагу до зміни клімату.

5.4.2 | Ключові елементи інтегрованої системи

Для сприяння сталому розвитку зеленої авіаційної промисловості в майбутньому виділяються деякі ключові елементи комплексної системи підтримки.

5.4.2.1 | Вдосконалення стратегії

Необхідно удосконалити бізнес-стратегії авіакомпаній для сталого розвитку зеленої авіаційної промисловості. Зміцнення бізнес-стратегій є ключовим фактором для успішних авіакомпаній, оскільки зелені зображення стають центральними елементами впізнаваності бренду. Ефективні бізнес-стратегії авіаційної промисловості є основною основою для "зеленого" розвитку і сприяють скороченню викидів вуглецю, економічним вигодам, соціального забезпечення, охорони навколишнього середовища і підвищення операційної ефективності (Дейлі, 2016; Кейванпур та ін., 2017). Авіакомпанії повинні оптимізувати свої бізнес-стратегії, знизити вплив авіаційної промисловості на довкілля і підвищити коефіцієнт використання ресурсів.

Авіакомпаніям необхідно оптимізувати свої маршрутні мережі та знизити експлуатаційні витрати на авіацію. Вони можуть підвищити ефективність авіаційної промисловості за рахунок співпраці з урядами, аеропортами та виробниками. Крім того, авіакомпанії можуть знизити витрату палива за рахунок оптимізації повітряних маршрутів, скорочення дальності польоту, точного розрахунку пропускної спроможності палива і скорочення процесу ковзання по землі при забезпеченні безпеки, підвищення ефективності експлуатації та скорочення викидів вуглекислого газу. Крім того, авіакомпаніям слід розробити методи оцінки ефективності роботи і провести аналіз витрат і вигод для підвищення операційної ефективності. Викиди вуглецю і шум, які перевищують норми, призводять до витрат, у той час як зниження впливу на навколишнє середовище принесе суттєві вигоди, наприклад, за рахунок продажу додаткових квот на викиди в системах торгівлі викидами.

5.4.2.2 | Інтеграція технологій

Технології є однією з фундаментальних рушійних сил розвитку "зеленої" авіаційної промисловості. Нові технологічні траєкторії і нові рішення зазвичай випливають з технологічних потоків між відповідними секторами, щоб зробити еволюцію та інтеграцію здійсненними. Гріштейн і Ханна (1997) використовують інтеграцію технологій для опису взаємодоповнюваності технологій, що сприяють

інноваціям. Хаклин (2007) зазначає, що злиття технологій, засноване на безперервних інноваціях, є важливим джерелом проривних інновацій.

Інтеграція інноваційних технологій може знизити вплив на навколишнє середовище і сприяє сталому розвитку зеленої авіаційної промисловості. Наприклад, літаки нового покоління і модифікації льотних двигунів, як правило, підвищують ефективність літаків приблизно на 50%. Двигун є основою літака, що визначає його економічну ефективність та надійність. Розробка нових двигунів дозволить значно підвищити паливну економічність. Зниження ваги літака може бути досягнуто за рахунок зменшення фарбування літака, зміни дизайну кабіни та бортового обладнання, а також використання композитних матеріалів. З точки зору конструкції, невеликі крила з закінченнями можуть бути використані для зниження опору повітря в польоті і зниження витрати палива. Для розвитку зеленої авіаційної промисловості авіакомпаніям необхідно впроваджувати більш досконалі моделі літаків з належним урахуванням відносних витрат і вигод. Крім того, авіакомпаніям необхідно активно співпрацювати з виробниками літаків для розробки і виробництва нових альтернативних видів палива, знижують викиди забруднюючих речовин та вуглецю. Інтеграція цих інноваційних технологій (наприклад, сучасних літаків і біопалива) може сприяти сталому розвитку зеленої авіаційної промисловості.

5.4.2.3 | Підтримка політики

Сталий розвиток зеленої авіаційної промисловості потребує політичної підтримки, яка може значно прискорити весь процес. Авіаційна промисловість-це галузь, що характеризується високим ступенем міжнародної конкуренції. Урядам необхідно розробити нову екологічну політику авіаційної промисловості, визнати промислові тенденції і зрозуміти промислові технології та інформацію, отриману науково - дослідними і дослідно-конструкторськими установами. Для досягнення сталого розвитку зеленої авіапром, передовий досвід і уроки, витягнуті з існуючих, пов'язаних з екологічної політики (наприклад, шум податку в Нідерландах, "міхур" політики США та ЄС ЄТС), повинні бути витягнуті (шо, 2001; Гервину, 2009; Ангер, 2010; Сгуриди і співавт., 2011; Гегг і співавт., 2015). Крім того, необхідно координувати вплив ринкової економіки і всієї галузі в цілому для сприяння сталому розвитку зеленої авіаційної промисловості.

Уряди повинні відігравати активну роль в якості інститутів управління галуззю. Вони можуть направляти інтеграцію технологій авіаційної промисловості, сприяти вдосконаленню стратегії авіакомпаній і заохочувати участь громадськості шляхом надання державних послуг, створення справедливої конкурентної

ринкового середовища, захисту законних прав споживачів і розробки ефективної екологічної політики. Урядам слід також сприяти структурній перебудові авіаційної промисловості, перетворення режимів розвитку і реформ для усунення перешкод на шляху сталого розвитку зеленої авіаційної промисловості. Необхідно визнати роль державного регулювання для підвищення ефективності оперативних втручань. Крім того, урядам необхідно просувати нові системи державної політики. Впровадження нових систем державної політики є важливим засобом для урядів забезпечити макроекономічний контроль і поліпшити екологічну політику.

5.4.2.4 | Участь громадськості

Громадськість, якій могли б керувати уряду, грає безліч ролей у зеленій авіаційної промисловості (Райлі та ін., 2010; Сантос та ін., 2018). По-перше, фізичні особи є потенційними пасажирами для авіаперевезень, а також потенційними відправниками і покупцями авіаперевезень. Крім того, окремі особи також можуть бути прихильниками зеленої авіаційної промисловості. Переваги окремих осіб певною мірою визначають рішення авіакомпаній, екологічну політику урядів і розвиток технологій, і все це також впливає на їх переваги. Таким чином, екологічна авіаційна промисловість буде краще розвинена, якщо окремі особи будуть ефективно брати участь у її сталий розвиток.

Уряди можуть заохочувати окремих осіб до участі у сталому розвитку зеленої авіаційної промисловості. Люди можуть просувати концепцію зеленого споживання. Більше число потенційних клієнтів, які дотримуються концепції зеленого споживання, може спонукати авіакомпанії приділяти більше уваги своєму зеленому іміджу. Авіакомпанії з "зеленим" іміджем можуть залучати приватних осіб за допомогою реклами, тим самим переміщаючи частку ринку з авіакомпаній без зеленого іміджу на авіакомпанії з "зеленим" іміджем, що підштовхне перших до прийняття заходів по підвищенню іміджу свого бренду. Ці авіакомпанії можуть надавати кошти для вдосконалення технологій (наприклад, скорочення фарбування літаків, раціонального дизайну салону і бортового обладнання, а також використання композитних матеріалів), що буде сприяти подальшому сталому розвитку зеленої авіаційної промисловості. Люди можуть дізнатися про розвиток технологій, пов'язаних із зеленою авіаційною промисловістю, з різних джерел (наприклад, веб-сторінок, телевізійних рекламних роликів і новин), які можуть розширити концепцію зеленого споживання людей. Інтеграція громадськості з урядами, технологіями і авіакомпаніями для

формування інтегрованої системи зеленої авіаційної промисловості допоможе успішно просувати сталий розвиток зеленої авіаційної промисловості.

5.5 | ВИСНОВКИ

У цьому дослідженні тенденція розвитку зеленої авіаційної промисловості була вивчена шляхом поєднання бібліометричного аналізу з вивченням літератури. З використанням відповідних технологій і програмного забезпечення була створена система BDAS зі збором і візуалізацією даних для визначення відповідних координаторів досліджень. З допомогою BDAS були визначені тенденції в області ключових слів для оцінки трьох тем "зеленої" авіаційної промисловості. Траєкторія дослідження показала, що шум привернув увагу на ранньому етапі, за яким послідувало вплив на навколишнє середовище, що також пояснює шум. В останні кілька років концепція зеленого іміджу стала популярною як для авіакомпаній, так і для пасажирів. Уявна інтегрована система підтримки впроваджується в якості майбутнього сталого розвитку зеленої авіаційної промисловості. Оцінка зеленої авіаційної промисловості була проведена на основі інновацій і стійкості. Крім того, були обговорені деякі ключові елементи інтегрованої системи підтримки зеленої авіаційної промисловості.

Оскільки авіаційна промисловість швидко розвивається по мірі розвитку суспільства та економіки, необхідно докласти численні зусилля для стримування її зростаючого впливу на навколишнє середовище. У цьому документі були побудовані BDA для вивчення тенденцій розвитку зеленої авіаційної промисловості та визначено три ключові теми: "шум", "вплив на навколишнє середовище" і "екологія". У відповідь на екологічні проблеми була впроваджена інтегрована система підтримки зеленої авіаційної промисловості з участю авіакомпаній, інженерів, урядів та громадськості. Результати дослідження показали, що ця система, що включає бізнес-стратегії авіакомпаній, інноваційні технології, екологічну політику та участь громадськості, може бути корисною для майбутнього сталого розвитку зеленої авіаційної промисловості, яка потребує підтримки всіх зацікавлених сторін.

У цьому дослідженні є кілька обмежень, оскільки дані дослідження були отримані від ISI WOS, а библиометрические дослідження в області аналогічних тим (наприклад, низьковуглецевий авіація) залишалися обмеженими. Однак, оскільки результати досліджень, наведені в цій статті, були засновані на об'єктивних даних, можна було бачити, що вони дають розумну оцінку стану зеленої авіаційної промисловості. Крім того, необхідні подальші дослідження.

Наприклад, існує необхідність в подальшій роботі, яка може розвинути результати досліджень, що містяться в цій статті. Крім того, в майбутньому слід вивчити тенденції розвитку деяких аналогічних тим (наприклад, низьковуглецевий авіація).

Загальні висновки

В даній дипломній роботі був проведений аналіз основних перспектив розвитку технічних засобів діагностування та контролю бортових систем інформаційного обміну літальних апаратів. В першому розділі були проаналізовані особливості архітектури DIMA, а також проаналізовано і висунуто на обговорення дослідження і розробки трьох ключових технологій в системі DIMA за останні роки. Нарешті, були висунуті тенденція розвитку технології DIMA future. В другому розділі був запропонований новий метод аналізу на основі моделей з безліччю обмежень для процесу динамічної реконфігурації IMA, при внесенні в систему змін чи при виникненні збою в системі. В третьому розділі для розширення мови AADL з підтримкою інструментів для формального аналізу властивостей безпеки системи при наявності несправностей, номінальну модель було доповнено визначеннями несправностей, що дозволяє проводити аналіз безпеки та впровадження системи на основі єдиної загальної моделі. було виявлено, що тісна інтеграція аналізу поведінкових помилок в AADL забезпечила тісний зв'язок між аналітиками системи та безпеки. Зміни, внесені в модель системи, були негайно відображені як у номінальному аналізі, так і в аналізі безпеки.

1. Huagang Xiong and Zhonghua Wang, Advanced Avionics Integration Techniques, National Defense Industry PressBerlin, 2009. pp. 1.
2. R. Fuchsen, IMA NextGen: A new technology for the Scarlett program, Aerospace and Electronic Systems Magazine, IEEE, Vol. 25, No. 10, pp. 10–16, 2010.
3. R. Wolfg and M. Jakovlievic. Distributed IMA and DO-297: Architectural, Communication And Certifcation Attributes. In IEEE 27th DASC, 2008.
4. G. Warden. Application of a Distributed Integrated Modular Avionics Test Bed to Sikorsky Aircraft. http://www.tttech.com, 2010. 5. T. Rogalski, S. Samolej and A. Tomczyk. ARINC 653 Based Time-Critical Application for European SCARLETT Project. In AIAA guidance, navigation, and control conference, pages 8–11, 2011.
5. T. Rogalski, S. Samolej and A. Tomczyk. ARINC 653 Based Time-Critical Application for European SCARLETT Project. In AIAA guidance, navigation, and control conference, pages 8–11, 2011.
6. T. Robati, A. Gherbi, A. E. Kouhen and J. Mullins, Design and simulation of distributed IMA architectures using TTEthernet: a model-driven approach, Journal of Ambient Intelligence & Humanized Computing, Vol. 8, No. 3, pp. 1–11, 2017.
7. . Q. Zhou, Z. Xiong, Z. Zhan, T. You and N. Jiang. The mapping mechanism between Distributed Integrated Modular Avionics and data distribution service. In International Conference on Fuzzy Systems & Knowledge Discovery, pages 2502–2507, 2016.
8. D. de Niz, K. Lakshmanan and R. Rajkumar. On the Scheduling of Mixed-Criticality Real-Time Task Sets. In IEEE 30th International Conference on Real-Time Systems Symposium, pages 291–300, 2009.
9. ARINC 653-1-2003. Avionics Application Software Standard Interface. ARINC Specifcation 653, 2003.
10. S. Saewong, R. Rajkumar and J. Lehoczky. Analysis of Hierarchical Fixed Priority Scheduling. In Proceedings of the Euromicro Conference on Real-Time Systems. pages 173–181, NY, 2002. IEEE.
11. L. Almeida and P. Pedreiras. Scheduling within temporal partitions: Response-time analysis and server design. In in the 4th ACM International Conference on Embedded Software, pages 9:95–103, Italy, 2004. Pisa
12. R. I. Davis and A. Burns. Resource Sharing in Hierarchical Fixed Priority Pre-Emptive Systems. In IEEE 27th International Conference on Real-Time Systems Symposium, Rio de Janeiro, Brazil, pages 257–270, 2006.
13. K. Lakshmanan, D. De Niz and R. Rajkumar, et al. Resource allocation in distributed mixed-criticality cyber-physical systems. In IEEE 30th International Conference on Distributed Computing Systems (ICDCS), pages 169–178, 2010.
14. K. Lakshmanan, D. de Niz and R. Rajkumar. Mixed-criticality task synchronization in zero-slack scheduling. In IEEE 17th Real-Time and Embedded Technology and Applications Symposium (RTAS), pages 47–56, 2011.
15. . H. W. Jin and S. Han, Temporal partitioning for mixed-criticality systems, Emerging Technologies & Factory Automation, Vol. 19, No. 6, pp. 1–4, 2011.

16. D. Tamas-Selicean and P. Pop. Design Optimization of MixedCriticality Real-Time Applications on Cost-Constrained Partitioned Architectures. In IEEE 30th International Conference on Real-Time Systems Symposium, pages 24–33, 2011.
17. D. Tamas-Selicean and P. Pop, Design Optimization of MixedCriticality Real-Time Embedded Systems, ACM Transactions on Embedded Computing Systems, Vol. 14, No. 3, pp. 1–8, 2015.
18. . Chuancai Gu, Nan Guan, Jinming Yu, et al. Partitioned Scheduling Policies on Multi-Processor Mixed-Criticality Systems. Journal of Software, pp. 284–297, 2014.
19. Roman Trüb, Georgia Giannopoulou, Andreas Tretter and Lothar Thiele, Implementation of Partitioned Mixed-Criticality Scheduling on a Multi-Core Platform, ACM Transactions on Embedded Computing Systems (TECS), Vol. 16, No. 5, pp. 1–21, 2017.
20. J. Barhorst, T. Belote, P. Binns, J. Hoffman, J. Paunicka, P. Sarathy, J. S. P. Stanfll, D. Stuart and R. Urzi. White paper: A research agenda for mixed-criticality systems. http://www.cse. wustl.edu/~cdgill/CPSWEEK0_MCAR, 2009.
21. . S. Vestal. Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance. In IEEE Proceedings Real-Time Systems Symposium. pages 239–243, 2007.
22. A. Mok. Fundamental design problems of distributed systems for the hard real-time environment. Cambridge, MA, USA, Tech. Rep., 1983.
23. N. Audsley. Optimal Priority Assignment and Feasibility of Static Priority Tasks with Arbitrary Start Times. Technical Report YCS 164, University of York. 1991.
24. F. Dorin, P. Richard, M. Richard, et al., Schedulability and sensitivity analysis of multiple criticality tasks with fxed-priorities, Real-Time Systems, Vol. 46, No. 3, pp. 305–331, 2010.
25. S. Baruah and S. Vestal. Schedulability analysis of sporadic tasks with multiple criticality specifcations. In Real-Time Systems, 2008. ECRTS'08. Euromicro Conference on. pages 147–155, 2008. IEEE.
26. S. K. Baruah, A. Burns and R. I. Davis. Response-time analysis for mixed criticality systems. In Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd. pages 34–43, 2011, IEEE.
27. S. Baruah, V. Bonifaci, G. D'Angelo, et al., Scheduling real-time mixed-criticality jobs, Computers, IEEE Transactions on, Vol. 61, No. 8, pp. 1140–1152, 2012.
28. S. Baruah, H. Li and L. Stougie. Towards the design of certifable mixed-criticality systems. In Real-Time and Embedded Technology and Applications Symposium (RTAS), 2010 16th IEEE. pages 13–22, 2010. IEEE.
29. H. Li and S. Baruah. Load-based schedulability analysis of certifable mixed-criticality systems. In Proceedings of the Tenth ACM International Conference on Embedded Software. pages 99–108, 2010, ACM.
30. H. Li and S. Baruah. An algorithm for scheduling certifable mixed-criticality sporadic task systems. In Real-Time Systems Symposium (RTSS), 2010 IEEE 31st. pages 183–192, 2010, IEEE.

31. N. Guan, P. Ekberg, M. Stigge, et al. Efective and efcient scheduling of certifable mixed-criticality sporadic task systems. In Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd. pages 13–23, 2011, IEEE.
32. S. Baruah and G. Fohler. Certifcation-cognizant time-triggered scheduling of mixed-criticality systems. In Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd. pages 3–12, 2011, IEEE.
33. S. K. Baruah, V. Bonifaci, G. D'Angelo, et al. Mixed-criticality scheduling of sporadic task systems. In Algorithms–ESA 2011. pages 555–566, Berlin, 2011, Springer.
34. S. Baruah, V. Bonifaci, G. D'Angelo, et al. The preemptive uniprocessor scheduling of mixed-criticality implicit-deadline sporadic task systems. In Real-Time Systems (ECRTS), 2012 24th Euromicro Conference on. pages 145–154, 2012, IEEE.
35. P. Ekberg and W. Yi. Outstanding Paper Award: Bounding and Shaping the Demand of Mixed-Criticality Sporadic Tasks. In Real-Time Systems (ECRTS), 2012 24th Euromicro Conference on. pages 135–144, 2012, IEEE.
36. A. K. Mok, X. Feng and D. Chen. Resource partition for real-time systems. In Real-Time Technology and Applications Symposium, 2001. Proceedings. Seventh IEEE. pages 75–84, 2001, IEEE.
37. H. Li and S. Baruah. Global mixed-criticality scheduling on multiprocessors. In Real-Time Systems (ECRTS), 2012 24th Euromicro Conference on. pages 166–175, 2012, IEEE.
38. R. M. Pathan. Schedulability analysis of mixed-criticality systems on multiprocessors. In Real-Time Systems (ECRTS), 2012 24th Euromicro Conference on. pages 309–320, 2012, IEEE.
39. F. Santy, L. George, P. Thierry, et al. Relaxing mixed-criticality scheduling strictness for task sets scheduled with FP. In Real-Time Systems (ECRTS), 2012 24th Euromicro Conference on. pages 155–165, 2012, IEEE.
40. J. Yao, J. Wu, Q. Liu, Z. Xiong and G. Zhu, System-Level Scheduling of Mixed-Criticality Trafcs in Avionics Networks, IEEE Access, Vol. 4, pp. 5880–5888, 2017.
41. M. Spuri. Holistic Analysis of Deadline Scheduled Real-Time Distributed Systems, RR-2873, INRIA, France, 1996.
42. M. Klein, T. Ralya, B. Pollak, et al., A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems, Kluwer Academic PublisherNorwell, 1993.
43. J. C. Palencia Gutiérrez, J. J. Gutiérrez García and M. González Harbour. On the schedulability analysis for distributed hard realtime systems. In Real-Time Systems, 1997. Proceedings., Ninth Euromicro Workshop on. pages 136–143, 1997, IEEE.
44. O. Redell and M. Sanfridson. Exact best-case response time analysis of fxed priority scheduled tasks. In Real-Time Systems, 2002. Proceedings. 14th Euromicro Conference on. pages 165–172, 2002, IEEE.

45. P. E. Hladik and A. M. Déplanche. Best-case response time analysis for precedence relations in hard real-time systems. In RealTime Systems Symposium Work-in-Progress Session. 2003.
46. J. J. G. García, J. C. P. Gutiérrez and M. G. Harbour. Schedulability analysis of distributed hard real-time systems with multipleevent synchronization. In Real-Time Systems, 2000. Euromicro RTS 2000. 12th Euromicro Conference on. pages 15–24, 2000, IEEE.
47. J. C. Palencia and M. G. Harbour. Exploiting precedence relations in the schedulability analysis of distributed real-time systems. In Real-Time Systems Symposium, 1999. Proceedings. The 20th IEEE. pages 328–339, 1999, IEEE.
48. K. Tindell. Adding time-ofsets to schedulability analysis. Technical Report UCS 221, Department of Computer Science, University of York, 1994.
49. J. C. Palencia and M. González Harbour. Schedulability analysis for tasks with static and dynamic ofsets. In Real-Time Systems Symposium, 1998. Proceedings., The 19th IEEE. pages 26–37, 1998, IEEE.
50. T. Pop. Scheduling and Optimisation of Heterogeneous Time/ Event-Triggered Distributed Embedded Systems. Linköping, 2003.
51. T. Pop. Analysis and Optimisation of Distributed Embedded Systems with Heterogeneous Scheduling Policies. Linköping, 2007.
52. M. A. O. Yugang, Yongjun ZHANG, Shiyao JIN. An Improved Schedulability Analysis Algorithm of Hard Real-Time Distributed System, Journal of Software, Vol. 12, No. 2, pp. 298–302, 2001.
53. . O. Redell and M. Torngren. Calculating exact worst case response times for static priority scheduled tasks with ofsets and jitter. In Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE. pages 164–172, 2002, IEEE.
54. Yao Chen, Qiao Li, Jun Lu and Huagang Xiong. Improved schedulability analysis for multiprocessor mixed-criticality systems. Journal of Beijing University of Aeronautics and Astronautics, pp. 1918–1926, 2015.
55. Hong Mu. Research and Implementation of Real-time System Schedulability Analysis and Simulation Tools, Master Thesis of University of Electronic Science and Technology, 2017.
56. Pujie Han, Zhengjun Zhai, Brian Nielsen and Ulrik Nyman. A Modeling Framework for Schedulability Analysis of Distributed Avionics Systems, In Proceedings of MARS/VPT, pages 150–168, 2018.
57. X. L. Teng and H. Pham, A software-reliability growth model for N-version programming systems, IEEE Transactions on Reliability, Vol. 51, No. 3, pp. 311–321, 2002.
58. X. Cai, M. R. Lyu and M. A. Vouk. An experimental evaluation on reliability features of N-version programming. In Proceedings of the International Symposium on Software Reliability Engineering (ISSRE 2005), pages 161–170, 2005.

59. . H. Yamachi, Y. Tsujimura, Y. Kambayashi, et al., Multi-objective genetic algorithm for solving N-version program design problem, Reliability Engineering and System Safety, Vol. 91, No. 9, pp. 1083–1094, 2006.
60. F. Vargas, R. D. R. Fagundes and D. J. Barros. Experimental results of a recovery block scheme to handle noise in speech recognition systems. In Proceedings of the 11th Asian Test Symposium (ATS'02), pages 224–229, 2002.
61. W. L. Yeung and S. A. Schneider, Design and verifcation of distributed recovery blocks with CSP, Formal Methods in System Design, Vol. 22, No. 3, pp. 225–248, 2003.
62. N. Navet, Y. Q. Song and F. Simonot, Worst-case deadline failure probability in real-time applications distributed over controller area network, Journal of Systems Architecture, Vol. 46, No. 7, pp. 607–617, 2000.
63. R. Dobrin, H. Aysan and S. Punnekkat. Maximizing the fault tolerance capability of fxed priority schedules. In Embedded and RealTime Computing Systems and Applications, 2008. RTCSA'08. 14th IEEE International Conference on. pages 337–346, 2008, IEEE.
64. P. K. Saraswat, P. Pop and J. Madsen, Task migration for faulttolerance in mixed-criticality embedded systems, ACM SIGBED Review, Vol. 6, No. 3, p. 6, 2009.
65. H. Aysan, R. Dobrin and S. Punnekkat. Task-Level Probabilistic Scheduling Guarantees for Dependable Real-Time Systems-A Designer Centric Approach. In 2011 14th IEEE International Symposium on. pages 281–287, 2011, IEEE.
66. Chongjie Dong and Yuqiang Chen. Real-Time Scheduling Algorithm of Dynamic with Fault-Tolerant in Heterogeneous Distributed Systems. Journal of System Simulation, pp. 1132–1140, 2017.
67. Junlong Zhou, Min Yin, Zhifang Li, Kun Cao and Jianming Yan, Fault-Tolerant Task Scheduling for Mixed-Criticality Real-Time Systems, Journal of Circuits, Systems and Computers, Vol. 26, No. 1, pp. 1–17, 2017.
68. H. Sariowan. A service curve approach to performance guarantees in integrated service networks. Ph.D. Dissertation, Univ Calif San Diego. 1996.
69. D. Bertsekas and R. Gallager, Data Networks, vol. 2nd, Prentice HallUpper Saddle River, 1992.
70. Rene L. Cruz, A Calculus for Net work Delay, Part I: Network Elements in Isolation, IEEE Transaction on Information Theory, Vol. 37, No. 1, pp. 114–131, 1991.
71. Rene L. Cruz, A Calculus for Net work Delay, Part II: Network Analysis, IEEE Transaction on Information Theory, Vol. 37, No. 1, pp. 132–141, 1991.
72. C. S. Chang, Performance Guarantees in Communication Networks, Springer-VerlagNew York, 2000.
73. J.-Y. Le Boudec and P. Thiran, Network Calculus. LNCS2050 ed., SpringerBerlin, 2004.
74. Y. Jiang, A basic stochastic network calculus, ACM SIGCOMM Computer Communication Review, Vol. 36, No. 4, pp. 123–134, 2006.
75. Y. Jiang and Y. Liu, Stochastic Network Calculus, SpringerHeidelberg, 2008.

76. M. Fidler, Survey of deterministic and stochastic service curve models in the network calculus, Communications Surveys & Tutorials, IEEE, Vol. 12, No. 1, pp. 59–86, 2010.
77. Luxi Zhao, Qiao Li, Wanqing Lin and Huagang Xiong, Stochastic network calculus for analysis of latency on TTEthernet network, Acta Aeronautica ET Astronautica Sinica, Vol. 37, No. 6, pp. 1953–1962, 2016.
78. Xuan Zhou, Feng He and Tong Wang. Using network calculus on worst-case latency analysis for TTEthernet in preemption transmission mode. In IEEE 10th International Conference on Signal Processing and Communication Systems (ICSPCS), pages 1–7, 2016.
79. S. Martin and P. Minet. Schedulablility Analysis of Flows Scheduled with FIFO: Application to the EF Class. In Workshop on Parallel and Distributed Real-Time Systems, 2006.
80. . H. Bauer, J. L. Scharbarg and C. Fraboul. Applying and Optimizing Trajectroy Approach for Performance Evaluation of AFDX Avionics Network. In ETFA 2009.
81. H. Bauer, J.-L. Scharbarg and C. Fraboul, Improving the WorstCase Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach, IEEE Transactions Industrial Informatics, Vol. 6, pp. 521–533, 2010.
82. H. Bauer, J.-L. Scharbarg and C. Fraboul, Applying trajectory approach to AFDX avionics network. In Proceedings of the 14th International Conference Emerging Technology Factory Automation, Mallorca, pages 1–8, 2009.
83. H. Bauer, J. L. Scharbarg and C. Fraboul, Improving the WorstCase Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach, IEEE Transaction Industrial Informatics, Vol. 6, pp. 521–533, 2010.
84. . M. Vojnovic and J. Le Boudec, Stochastic analysis of some expedited forwarding networks. In Proceedings of the Infocom, New York, 2002.
85. H. Charara, J. L. Scharbarg, J. Ermont, et al. Methods for bounding end-to-end delays on an AFDX network. In IEEE 18th Euromicro Conference on Real-Time Systems, pages 197–202, 2006.
86. J. L. Scharbarg and C. Fraboul, Methods and tools for the temporal analysis of avionic networks, INTECH Open Access PublisherQazvin, 2010.
87. C. Canew and R. Guerra Global View of Methods for Evaluating End-To-End Delays on AFDX. In 5th Real-Time Systems Seminar. pages 6, 2011.
88. H. Charara and C. Fraboul. Modelling and simulation of an avionics full duplex switched ethernet. In Telecommunications, 2005. advanced industrial conference on telecommunications/service assurance with partial and intermittent resources conference/elearning on telecommunications workshop. aict/sapir/elete 2005. proceedings, pages 207–212, 2005, IEEE.
89. J. L. Scharbarg and C. Fraboul. Simulation for end-to-end delays distribution on a switched ethernet. In 2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pages 1092–1099, 2007.

90. M. Adnan, J. L. Scharbarg, J. Ermont, et al. Model for worst case delay analysis of an AFDX network using timed automata. In 2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pages 1–4, 2010.
91. Airlines Electronic Engineering Committee. Avionics Application Software Standard Interface Part 1-Required Services; ARINC Document ARINC Specification 653 P1-3; Aeronautical Radio, Inc.: Annapolis, MD, USA, 2010.
92. STANAG, NATO. 4626-2005 Modular and Open Avionics Architecture (Part I: Architecture); North Atlantic Organization: Brussels, Belgium, 2005; pp. 24–34.
93. Jolliffe, G. Producing a safety case for IMA blueprints. In Proceedings of the 24th Digital Avionics Systems Conference, Washington, DC, USA, 30 October–3 November 2005; Volume 2; IEEE: Piscataway, NJ, USA, 2005.
94. López-Jaquero, V.; Montero, F.; Navarro, E.; Esparcia, A.; Catal'n, J.A. Supporting ARINC 653-based dynamic reconfiguration. In Proceedings of the 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, Helsinki, Finland, 20–24 August 2012; IEEE: Piscataway, NJ, USA, 2012.
95. Bieber, P.; Noulard, E.; Pagetti, C.; Planche, T.; Vialard, F. Preliminary design of future reconfigurable IMA platforms. ACM Sigbed Rev. 2009, 6, 1–5.
96. Hilbrich, R.; van Kampenhout, R. Dynamic reconfiguration in NoC-based MPSoCs in the avionics domain. In Proceedings of the 3rd International Workshop on Multicore Software Engineering, ACM, New York, NY, USA, May 2010.
97. Ding, M. Research on Reconfiguration and Verification Methods for Integrated Modular Avionics. Ph.D. Thesis, Northwest University, Xi'an, China, 2019.
98. Shukla, J.; Das, B.; Pant, V. Stability constrained optimal distribution system reconfiguration considering uncertainties in correlated loads and distributed generations. Int. J. Electr. Power 2018, 99, 121–133.
99. Ellis, S.M. Dynamic software reconfiguration for fault-tolerant real-time avionic systems. Microprocess. Microsyst. 1997, 21, 29–39.
100. van Vliet, J.C. Software Engineering-Principles and Practice, 3rd ed.; Wiley: Hoboken, NJ, USA, 2008.
101. SAE. AS5506A: Architecture Analysis and Design Language (AADL) Version 2.0; SAE: Warrendale, PA, USA, 2009.
102. . SAE. AS5506 Annex: Behavior Specification V2.0; SAE: Warrendale, PA, USA, 2011.
103. Yang, Z.; Hu, K.; Ma, D.; Bodeveix, J.; Pi, L.; Talpin, J. From AADL to Timed Abstract State Machines: A verified model transformation. J. Syst. Softw. 2014, 93, 42–68.
104. Walker, M.; Reiser, M.O.; Tucci-Piergiovanni, S.; Papadopoulos, Y.; Lönn, H.; Mraidha, C.; Parker, D.; Chen, D.; Servat, D. Automatic optimisation of system architectures using EAST-ADL. Syst. Softw. 2013, 86, 2467– 2487.

105. Feiler, P.H.; Gluch, D.P. Model-Based Engineering with AADL: An introduction to the SAE Architecture Analysis & Design Language; Addison-Wesley: Boston, MA, USA, 2012.
106. Bozzano, M.; Cimatti, A.; Katoen, J.P.; Nguyen, V.Y.; Noll, T.; Roveri, M. Safety, dependability and performance analysis of extended AADL models. Comput. J. 2010, 54, 754–775.
107. Hugues, J.; Zalila, B.; Pautet, L.; Kordon, F. From the prototype to the final embedded system using the Ocarina AADL tool suite. ACM Trans. Embed. Comput. Syst. (TECS) 2008, 7, 42.
108. Chkouri, M.Y.; Robert, A.; Bozga, M.; Sifakis, J. Translating AADL into BIP-application to the verification of real-time systems. In Proceedings of the International Conference on Model Driven Engineering Languages and Systems, Toulouse, France, 28 September–3 October 2008; Springer: Berlin/Heidelberg, Germany, 2008.
109. Zhang, F.; Zhao, Y.; Ma, D.; Niu, W. Formal Verification of Behavioral AADL Models by Stateful Timed CSP. IEEE Access 2017, 5, 27421–27438.
110. Zhao, Z.; Zhang, J.; Sun, Y.; Liu, Z. In Modeling of Avionic Display System for Civil Aircraft Based on AADL, 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 4121–4126.
111. Liu, Z.; Zhao, Z. In Modeling and Schedulability Verfication of IMA Partitioning Based on AADL, 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 417–420.
112. Liu, W. AADL Model Transformation and Verification. Master's Thesis, Shaanxi Normal University, Xi'an, China, 2013.
113. Wu, Y.; Li, S. AADL Model Based on TPN. Comput. Technol. Dev. 2014, 24, 88–91.
114. Hadad, A.S.A.; Ma, C.; Ahmed, A.A.O. Formal Verification of AADL Models by Event-B. IEEE Access 2020, 8, 72814–72834.
115. Sendall, S.; Kozaczynski, W. Model transformation: The heart and soul of model-driven software development. IEEE Softw. 2003, 20, 42–45.
116. Cuadrado, J.S.; Guerra, E.; de Lara, J. Static Analysis of Model Transformations. IEEE Trans. Softw. Eng. 2017, 43, 868–897.
117. Hu, K.; Zhang, T.; Yang, Z.; Tsai, W. Exploring AADL verification tool through model transformation. J. Syst. Architect. 2015, 61, 141–156.
118. Chkouri, M.Y.; Robert, A.; Bozga, M.; Sifakis, J. Translating AADL into BIPapplication to the Verification of Real-Time Systems, Models in Software Engineering; Springer: Berlin/Heidelberg, Germany, 2009; pp. 5–19.
119. Berthomieu, B.; Bodeveix, J.P.; Chaudet, C.; Dal Zilio, S.; Filali, M.; Vernadat, F. Formal Verification of AADL Specifications in the Topcased Environment, Reliable Software Technologies–Ada-Europe 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 207–221.
120. Rugina, A.-E.; Kanoun, K.; Kaâniche, M. A System Dependability Modeling Framework Using AADL and GSPNs, Architecting Dependable Systems IV; Springer: Berlin/Heidelberg, Germany, 2007; pp. 14–38.

121. Bozzano, M.; Cavada, R.; Cimatti, A.; Katoen, J.-P.; Nguyen, V.Y.; Noll, T.; Olive, X. Formal verification and validation of AADL models. In Proceeding of Embedded Real-Time Software and Systems 2010 (ERTS 2010), Toulouse, France, 19-21 May 2010.
122. Kabir, S.; Papadopoulos, Y. Applications of Bayesian networks and Petri nets in safety, reliability, and risk assessments: A review. Saf. Sci. 2019, 115, 154–175.
123. Rajkumar, S.M.; Chakraborty, S.; Dey, R.; Deb, D. Online Delay Estimation and Adaptive Compensation in Wireless Networked System: An Embedded Control Design. Int. J. Control Autom. Syst. 2020, 18, 856–866.
124. Luan, W.; Qi, L.; Zhao, Z.; Liu, J.; Du, Y. Logic Petri Net Synthesis for Cooperative Systems. IEEE Access 2019, 7, 161937–161948.
125. Jensen, K. Coloured Petri Nets. In Petri Nets: Central Models and Their Properties; Springer: Berlin/Heidelberg, Germany, 1987; pp. 248–299.
126. Jensen, Kurt. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use; Springer Science & Business Media: Berlin, Germany, 2013; Volume 1.
127. Huber, P.; Jensen, K.; Shapiro, R.M. Hierarchies in coloured Petri nets. In Proceedings of the International Conference on Application and Theory of Petri Nets, Bratislava, Slovakia, 24–29 June 2018; Springer: Berlin/Heidelberg, Germany, 1989.
128. Marsan, M.A.; Balbo, G.; Conte, G.; Donatelli, S.; Franceschinis, G. Modelling with Generalized Stochastic Petri Nets; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1994.
129. Ajmone Marsan, M.; Conte, G.; Balbo, G. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. ACM Trans. Comput. Syst. (TOCS) 1984, 2, 93–122.
130. Chiola, G.; Marsan, M.A.; Balbo, G.; Conte, G. Generalized stochastic Petri nets: A definition at the net level and its implications. IEEE Trans. Softw. Eng. 1993, 19, 89–107.
131. Bugarin, A.J.; Barro, S. Fuzzy reasoning supported by Petri nets. IEEE Trans. Fuzzy Syst. 1994, 2, 135–150.
132. Li, Y.; Chen, Y.; Tang, N.; Yang, L. 2016. Modeling and analysis of failure mechanism dependence based on petri net. In Proceedings of the Prognostics and System Health Management Conference, Chengdu, China, 19-21 October 2016; pp. 1–7.
133. Wieland, C.; Schmid, O.; Meiler, M.; Wachtel, A.; Linsler, D. Reliability computing of polymer-electrolytemembrane fuel cell stacks through petri nets. J. Power Sources 2009, 190, 34–39. 44.
134. Sunanda, B.E.; Seetharamaiah, P. Modeling of safety-critical systems using petri nets. ACM SIGSOFT Softw. Eng. Notes 2015, 40, 1–7.
135. Li, W.; He, M.; Sun, Y.; Cao, Q. A novel layered fuzzy Petri nets modelling and reasoning method for process equipment failure risk assessment. J. Loss Prevent. Proc. 2019, 62, 103953.
136. Gonçalves, P.; Sobral, J.; Ferreira, L.A. Unmanned aerial vehicle safety assessment modelling through petri Nets. Reliab. Eng. Syst. Safe 2017, 167, 383–393.

137. Liu, R. Reliability Modeling of Integrated Modular Avionics System Platform Using AADL, and GSPN Analysis Method. Master's Thesis, Civil Aviation University of China, Tianjin, China, 2016.
138. Li, Z.; Wang, S.; Zhao, T.; Liu, B. A hazard analysis via an improved timed colored petri net with time– space coupling safety constraint. Chin. J. Aeronaut. 2016, 29, 1027–1041.
139. Arena, D.; Criscione, F.; Trapani, N. Risk assessment in a chemical plant with a CPN-HAZOP Tool. IFACPapersOnLine 2018, 51, 939–944.
140. Patel, R.; Gojiya, A.; Deb, D. Failure Reconfiguration of Pumps in Two Reservoirs Connected to Overhead Tank Innovations in Infrastructure. In Innovations in Infrastructure; Springer: Singapore, 2019; pp. 81–92.
141. Kapoor, D.; Deb, D.; Sahai, A.; Bangar, H. Adaptive failure compensation for coaxial rotor helicopter under propeller failure 2012. In Proceedings of the American Control Conference (ACC), Montreal, QC, Canada, 27–29 June 2012; pp. 2539–2544.
142. Committee, A.E. ARINC 664 Aircraft Data Networks, Part7: Avionics Full Duplex Switched Ethernet (AFDX) Network; Aeronautical Radio, Inc.: Annapolis, MD, USA, 2005.
143. Prisaznuk, P.J. ARINC 653 role in integrated modular avionics (IMA). In Proceedings of the 2008 IEEE/AIAA 27th Digital Avionics Systems Conference, St. Paul, MN, USA, 26–30 October 2008; IEEE: Piscataway, NJ, USA, 2008.
144. Zhang, F.; Chu, W.; Fan, X.; Wan, M. Research on architecture of integrated modular avionics [J]. Electron. Opt. Control 2009, 9, 013.
145. Reis, J.G.; Wanner, L.; Fröhlich, A.A. A framework for dynamic real-time reconfiguration. In Proceedings of the 2015 Euromicro Conference on Digital System Design (DSD), Funchal, Portugal, 26–28 August 2015; IEEE: Piscataway, NJ, USA, 2015.
146. Aeronautical Radio. Avionics Application Software Standard Interface; ARINC653: Annapolis, MD, USA, 2010.
147. Montano, G.; McDermid, J. Human Involvement in Dynamic Reconfiguration of Integrated Modular Avionics, Avionics. In Proceedings of the 27th Digital Avionics Systems Conference, St. Paul, MN, USA, 26–30 October 2008; IEEE: Piscataway, NJ, USA, 2008.
148. Zhou, Q.; Gu, T.; Hong, R.; Wang, S. An AADL-based design for dynamic reconfiguration of DIMA. In Proceedings of the 2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC), East Syracuse, NY, USA, 5–10 October 2013; IEEE: Piscataway, NJ, USA, 2013.
149. Montano, G.; Norridge, P.; Sullivan, W.; Topping, C.; Wishart, A.; Bubenhagen, F.; Fiethe, B.; Michalik, H.; Osterloh, B.; et al. Dynamically Reconfigurable Processing Module for Future Space Applications. In Proceedings of the DASIA 2010 Data Systems In Aerospace, Budapest, Hungary, 1-4 June 2010; Volume 682.
150. Suo, D.; An, J.; Zhu, J. A new approach to improve safety of reconfiguration in integrated modular avionics. In Proceedings of the 2011 IEEE/AIAA 30th Digital

Avionics Systems Conference (DASC), Seattle, WA, USA, 16–20 October 2011; IEEE: Piscataway, NJ, USA, 2011.
151. Arshad, N. Dynamic reconfiguration of software systems using temporal planning. Ph.D. Thesis, University of Colourado, Boulder, CO, USA, 2003.
152. Montano, G. Dynamic reconfiguration of safety-critical systems: Automation and human involvement. Ph.D. Thesis, University of York, York, UK, 2011.
153. Quan, Z.; Wang, S. IMA reconfiguration modelling and reliability analysis based on AADL. In Proceedings of the 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent, Hong Kong, China, 4–7 June 2014; IEEE: Piscataway, NJ, USA, 2014.
154. Suo, D.; An, J.; Zhu, J. AADL-based modelling and TPN-based verification of reconfiguration in integrated modular avionics. In Proceedings of the 2011 18th Asia Pacific Software Engineering Conference (APSEC), Ho Chi Minh, Vietnam, 5–8 December 2011; IEEE: Piscataway, NJ, USA, 2011.
155. . Aerospace, S.A.E. SAE Architecture Analysis and Design Language (AADL); Annex Volume 2: Annex F: ARINC653 Annex; SAE International: USA, 2009.
156. Feiler, P.H.; Gluch, D.P.; Hudak, J.J. The Architecture Analysis & Design Language (AADL): An Introduction. No. CMU/SEI-2006-TN-011; Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst: Pittsburgh, PA, USA, 2006.
157. Aerospace, S.A.E. SAE Architecture Analysis and Design Language (AADL); Annex Volume 2: Annex D: Behaviour Model Annex; SAE International: USA, 2009.
158. Murata, T. Petri nets: Properties, analysis and applications. Proc. IEEE 1989, 77, 541–580.
159. Petri, C.A. Kommunikation mit Automaten. Bonn: Institute fur Instrumentelle Mathematik, Schriften des IIM Nr.3, 1962. Also, English Translation: Communication with Automata, Tech. Rep. RADC-TR-65–377, 1966, Volume 1.
160. Jensen, K. Coloured Petri nets: A high level language for system design and analysis. In Proceedings of the International Conference on Application and Theory of Petri Nets, Bonn, Germany, June 1989; Springer: Berlin/Heidelberg, Germany, 1989.
161. Kristensen, L.M.; Christensen, S.; Jensen, K. The practitioner's guide to coloured Petri nets. Int. J. Softw. Tools Technol. Transf. (STTT) 1998, 2, 98–132.
162. Jensen, K.; Munkegade, N. An Introduction to the Theoretical Aspects of Coloured Petri Nets. In Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, Germany, 1994; Volume 803.
163. Jensen, K.; Kristensen, L.M.; Wells, L. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. Int. J. Softw. Tools Technol. Transf. 2007, 9, 213–254.
164. Van der Aalst, W.M. The application of Petri nets to workflow management. J. Circuits Syst. Comput. 1998, 8, 21–66.
165. STANAG, NATO. 4626-2005 Modular and Open Avionics Architecture (Part VI: Guidelines for System Issues); Volume 4: System Configuration/Reconfiguration page: 7–20; North Atlantic Organization: Brussels, Belgium, 2005.

166. Feiler, P.H.; Lewis, B.A.; Vestal, S. The SAE Architecture Analysis & Design Language (AADL) a standard for engineering performance critical systems. In Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, Munich, Germany, 4–6 October 2006; IEEE: Piscataway, NJ, USA, 2006.
167. Aerospace, S.A.E. SAE Architecture Analysis and Design Language (AADL); Annex Volume 1: Annex E: Error Model Annex; SAE International: USA, 2006.
168. SEI AADL Team. An Extensible Open Source AADL Tool Environment (OSATE); Software Engineering Institute: Pittsburgh, PA, USA, 2006.
169. Beaudouin-Lafon, M.; Mackay, W.E.; Jensen, M.; Andersen, P.; Janecek, P.; Lassen, M.; Lund, K.; Mortensen, K.; Munck, S.; Ratzer, A.; et al. CPN/Tools: A tool for editing and simulating coloured petri nets ETAPS tool demonstration related to TACAS. In Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Genova, Italy, 2–6 April 2001; Springer: Berlin/Heidelberg, Germany, 2001.
170. Joshi A, Miller SP, Whalen M, Heimdahl MP. A proposal for model-based safety analysis. In: Proceedings of 24th digital avionics systems conference. 2005.
171. Bozzano M, Cimatti A, Griggio A, Mattarei C. Efficient anytime techniques for model-based safety analysis. In: Computer aided verification. 2015.
172. Hönig P, Lunde R, Holzapfel F. Model based safety analysis with smartiflow. Information 2017;8(1).
173. Lisagor O, Kelly T, Niu R. Model-based safety assessment: review of the discipline and its challenges. In: The proceedings of 2011 9th international conference on reliability, maintainability and safety. 2011.
174. Gudemann M, Ortmeier F. A framework for qualitative and quantitative formal model-based safety analysis. In: HASE 2010. 2010.
175. Friedenthal S, Moore A, Steiner R. A practical guide to SysML: the systems modeling language. Morgan Kaufmann; 2014.
176. Helle P. Automatic SysML based safety analysis. In: Proceedings of the 5th international workshop on model based architecting and construction of embedded systems. 2012. p. 19–24.
177. Mhenni F, Nguyen N, Choley J-Y. Automatic fault tree generation from SysML system models. In: 2014 IEEE/ASME international conference on advanced intelligent mechatronics. IEEE; 2014, p. 715–20.
178. AS5506C. Architecture analysis & design language. SAE International; 2017.
179. Feiler P, Hudak J, Delange J, Gluch D. Architecture fault modeling and analysis with the error model annex, version 2. Technical report CMU/SEI-2016-TR-009, Software Engineering Institute; 2016.
180. Bozzano M, Cimatti A, Katoen J-P, Nguyen VY, Noll T, Roveri M. The COMPASS approach: Correctness, modeling and performability of aerospace systems. In: Computer safety, reliability, and security. Springer Berlin Heidelberg; 2009.

181. MathWorks. The mathWorks inc. Simulink product web site. 2004, http://www.mathworks.com/products/simulink. [Accessed 30 September 2017].
182. Joshi A, Heimdahl MP. Model-based safety analysis of simulink models using SCADE design verifier. In: SAFECOMP. In: LNCS, vol. 3688. 2005. p. 122.
183. Chen D, Mahmud N, Walker M, Feng L, Lönn H, Papadopoulos Y. Systems modeling with EAST-ADL for fault tree analysis through hip-HOPS*. IFAC Proc Vol 2013;46(22):91–6.
184. Ansys medini tool. 2020, https://www.ansys.com/products/systems/ansys-medi ni-analyze/medini-analyze-capabilities. [Accessed 17 November 2020].
185. Prosvirnova T, Batteux M, Brameret P-A, Cherfi A, Friedlhuber T, Roussel JM, et al. The AltaRica 3.0 project for model-based safety assessment. IFAC 2013;46(22).
186. Bieber P, Farges J-L, Pucel X, Sèjeau L-M, Seguin C. Model Based Safety Analysis for co-assessment of operation and system safety: application to specific operations of unmanned aircraft. In: ERTS2. 2018.
187. Stewart D, Liu J, Heimdahl M, Whalen M, Cofer D, Peterson M. The safety annex for architecture analysis and design language. In: 10th edition European congress embedded real time systems. 2020.
188. Joshi A, Heimdahl MP. Behavioral fault modeling for model-based safety analysis. In: Proceedings of the 10th IEEE high assurance systems engineering symposium. 2007.
189. Feiler P, Gluch D. Model-based engineering with AADL: An introduction to the SAE architecture analysis & design language. Addison-Wesley Professional; 2012.
190. Cofer DD, Gacek A, Miller SP, Whalen MW, LaValley B, Sha L. Compositional verification of architectural models. In: NFM 2012, vol. 7226. 2012. p. 126–40.
191. Halbwachs N, Caspi P, Raymond P, Pilaud D. The synchronous dataflow programming language lustre, vol. 79, no. 9. IEEE; 1991, p. 1305–20.
192. Gacek A, Backes J, Whalen M, Wagner L, Ghassabani E. The JKind model checker. Lecture notes in computer science, vol. 10982, Chan: Springer; 2018.
193. AIR 6110. Contiguous aircraft/system development process example. SAE; 2011.
194. Bozzano M, Cimatti A, Pires AF, Jones D, Kimberly G, Petri T, et al. Formal design and safety analysis of AIR6110 wheel brake system. In: CAV 2015, Proceedings, Part I. 2015. p. 518–35.
195. Bozzano M, Cimatti A, Mattarei C, Tonetta S. Formal safety assessment via contract-based design. In: Automated technology for verification and analysis. 2014.
196. Stewart D, Whalen M, Cofer D, Heimdahl MP. Architectural modeling and analysis for safety engineering. In: IMBSA 2017. 2017. p. 97–111.
197. Ghassabani E, Gacek A, Whalen MW. Efficient generation of inductive validity cores for safety properties. 2016, CoRR, abs/1603.04276.
198. Ghassabani E, Whalen MW, Gacek A. Efficient generation of all minimal inductive validity cores. In: 2017 formal methods in computer aided design. 2017. p. 31–8.

199. Stewart D, Liu J, Heimdahl M, Whalen M, Cofer D, Peterson M. Architectural modeling and analysis for safety engineering. NASA final report, 2019, https: //github.com/loonwerks/AMASE/tree/master/doc/AMASE_Final_Report_2019.
200. Backes J, Cofer D, Miller S, Whalen MW. Requirements analysis of a quadredundant flight control system. In: NFM. In: LNCS, vol. 9058. 2015. p. 82–96.
201. SAE ARP 4754A. Guidelines for development of civil aircraft and systems. SAE International; 2010.
202. Stewart D, Liu J, Whalen M, Cofer D, Peterson M. Safety annex for AADL repository. GitHub; 2018, https://github.com/loonwerks/AMASE. [Accessed 17 October 2020].
203. Driscoll K, Sivencrona H, Zumsteg P. Byzantine fault tolerance, from theory to reality. In: SAFECOMP. In: LNCS. 2003.
204. SAE ARP 4761. Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. SAE International; 1996.
205. Bozzano M, Cimatti A, Roveri M, Katoen JP, Nguyen VY, Noll T. Codesign of dependable systems: A component-based modeling language. In: 2009 7th IEEE/ACM international conference on formal methods and models for co-design. 2009.
206. Bozzano M, Cimatti A, Katoen J-P, Yen Nguyen V, Noll T, Roveri M. Model-based codesign of critical embedded systems. 507, 2009.
207. Bieber P, Bougnol C, Castel C, Heckmann JP, Kehren C, Metge S, et al. Safety assessment with AltaRica - lessons learnt based on two aircraft system studies. In: 18th IFIP world computer congress. 2004.
208. PHM Technology. MADE for model-based FTA. 2021, https://www.phmtechnology.com/assets/downloads/default/MADe%20for%20Model-based%20FTA.pdf. [Accessed 12 March 2021].
209. Bozzano M, Cimatti A, Lisagor O, Mattarei C, Mover S, Roveri M, et al. Symbolic model checking and safety assessment of AltaRica models. In: Science of computer programming, Vol. 98. 2011.
210. Bozzano M, Cimatti A, Tapparo F. Symbolic fault tree analysis for reactive systems. In: ATVA. 2007.
211. Bittner B, Bozzano M, Cavada R, Cimatti A, Gario M, Griggio A, et al. The xSAP safety analysis platform. In: TACAS. 2016.