

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ ННЦ ЗФН _____
(повна назва)

Кафедра _____ Програмної інженерії _____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

_____ другий (магістерський) _____
(рівень вищої освіти)

Дослідження методів побудови інформаційних систем підтримки прийняття
рішень реального часу
(тема)

Виконав: студент 2 курсу, групи ПЗСзм-18-1
спеціальності 121- Інженерія програмного
забезпечення _____

(код і повна назва спеціальності)

освітньо-професійної програми Програмне
забезпечення систем _____

(повна назва освітньої програми)

_____ Мерчанський Є.В. _____

(прізвище, ініціали)

Керівник _____ проф. Дудар З.В. _____

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____

З.В.Дудар

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121– Інженерія програмного забезпечення

(код і повна назва)

Освітньо-професійна програма Програмне забезпечення систем

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 20 ____ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Мерчанському Євгенію Вадимовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів побудови інформаційних систем підтримки прийняття рішень реального часу

затверджена наказом по університету від «__» _____ 2019 р № __ Стз

заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії

10 грудня 2019 р.

3. Вихідні дані до роботи проаналізувати існуючі алгоритми, що використовуються для вимоги до веб-сервісу, що буде розроблено, середовище проектування Microsoft VisualStudio 2017, мова розробки C#, сервер IIS Express

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, опис запропонованих варіантів оптимізації, використовувані методи та алгоритми, опис розробленої програмної системи, опис застосованих оптимізацій, аналіз можливих застосувань

5. Консультанти розділів роботи

Найменування розділу	Консультант (посаду, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. Дудар З.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	10 жовтня 2019 р.	
2.	Огляд існуючих методів	27 жовтня 2019 р.	
3.	Проектування та розробка ПЗ	15 листопада 2019 р.	
4.	Підготовка пояснювальної записки	25 листопада 2019 р.	
5.	Спецчастина	26 листопада 2019 р.	
6.	Підготовка презентації та доповіді	30 листопада 2019 р.	
7.	Попередній захист	10 грудня 2019 р.	
8.	Нормоконтроль, рецензування	11 грудня 2019 р.	
9.	Занесення диплома в електронний архів	12 грудня 2019 р.	
10.	Допуск до захисту в зав. кафедри	14 грудня 2019 р.	
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання _____ 2019 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Дудар З.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи: 87 с., 32 рис., 3 додатки, 28 джерел.

ДИНАМІЧНІ СТРУКТУРИ, ТЕМПОРАЛЬНА ЛОГІКА, МЕРЕЖІ ПЕТРІ, СИСТЕМИ РЕАЛЬНОГО ЧАСУ.

Об'єктом дослідження магістерської роботи є складні динамічні системи реального часу.

Метою роботи є розробка методів і відповідних програмних засобів, що дозволяють створювати імітаційні моделі складних динамічних структур

Методи розробки базуються на інструментах роботи мережами Петрі.

У результаті роботи було проаналізовано предметну галузь, поставлено задачу та спроектовано програмну систему, що реалізує метод розфарбованих мереж Петрі.

DYNAMIC STRUCTURES, TEMPORAL LOGIC, PETRI NETWORKS, REAL-TIME SYSTEMS.

The object of the master's study is complex dynamic real-time systems.

The purpose of the work is to develop methods and appropriate software that allow you to create simulation models of complex dynamic structures

The development methods are based on the Petri tools.

As a result, the subject area was analyzed, a task was set and a software system was implemented that implements the method of colored Petri nets.

ЗМІСТ

Вступ.....	7
1 Аналіз досліджень в галузі моделювання динамічних систем.....	10
1.1 Підходи до розробки динамічних інтелектуальних систем	10
1.2 Динамічні інтелектуальні системи як підклас динамічних систем	12
1.3 Інтелектуальні системи підтримки прийняття рішень реального часу ..	14
1.4 Завдання моделювання складних динамічних систем	17
1.5 Аналіз підходів до моделювання складних динамічних систем	18
1.6 Постановка задач дослідження	20
2 Опис проведених теоретичних досліджень	22
2.1 Аналіз підходів до представлення й моделювання часу в складних динамічних системах	22
2.2 Вибір формального апарата імітаційного моделювання	26
2.3 Моделі на Мережах Петрі	29
3 Аналіз результатів дослідження	33
3.1 Аналіз методів моделювання процесів у складних динамічних системах	33
3.2 Модифікації мереж Петрі для моделювання процесів	34
3.3 Модифікація РМП РЧ із підтримкою якісних темпоральних залежностей	38
3.4 Верифікація моделей процесів, створених на основі мереж Петрі	43
4 Програмна реалізація базових інструментальних засобів.....	46
4.1 Розробка прототипу інструментарію	46
4.2 Організація функціонування моделі	52
4.3 Розробка інструментарію в середовищі Microsoft Visual Studio	54
4.4 Зберігання результатів	57
5. Опис можливості використання отриманих результатів.....	64

Висновки	66
Перелік джерел посилання	68
Додаток А Листінг	71
Додаток Б Слайди презентації.....	74
Додаток В Відгук і рецензії	84

ВСТУП

У цей час активно розробляються методи й програмні засоби проектування математичного й програмного забезпечення інтелектуальних систем підтримки прийняття рішень (ІСППР), включаючи найбільш складних їхніх представників – ІСППР реального часу (ІСППР РЧ). ІСППР РЧ призначені для допомоги особам, що приймають рішення (ОПР), при керуванні й моніторингу складних динамічних систем (ДС) – технічних (технологічних), транспортних, організаційних і інших – в умовах досить твердих часових обмежень і при наявності різного типу невизначеності у вступників інформації. До таких систем можна віднести, зокрема, об'єкти енергетики. Важливим завданням, що виникає при проектуванні й розробці ІСППР РЧ, є розробка методів і програмних засобів моделювання складних ДС, включаючи засобу представлення й оперування часовими (темпоральними) залежностями, як кількісними, так і якісними. У ряді сучасних і дорогих комерційних інструментальних комплексів для побудови складних ДС (G2, Rtworks і ін.) є засоби відображення фактора часу й темпоральних залежностей, однак вони досить примітивні й не відповідають сьогоднішнім вимогам.

Слід зазначити, що й ІСППР РЧ ставляться до класу складних ДС – динамічних інтелектуальних систем (ДІС), заснованих на використанні методів штучного інтелекту й пошуку (висновку) рішень на основі знань. Одним з основних блоків ДІС є підсистема моделювання, використовувана як для імітації процесів у складних ДС, так і для аналізу наслідків прийнятих рішень і вибору найкращих рекомендацій для ОПР. Питаннями моделювання складних ДС, у тому числі в плані використання засобів моделювання в інтелектуальних системах типу ДІС і ІСППР.

Завдання дослідження процесів і закономірностей, які визначають функціонування складних ДС у різних ситуаціях, і розробки методів і програмних засобів моделювання таких процесів, є досить актуальними, оскільки надійність і

передбачуваність поведінки складних ДС найчастіше є більш важливими властивостями, чому продуктивність, модифікованість і т.п. Це пов'язане з істотним ризиком виникнення помилок на етапі проектування ДС і дуже високою ціною прояву цих помилок на стадії експлуатації. Однак класичний підхід – аналітичний аналіз ДС як фізичної системи, описуваної, наприклад, диференціальними рівняннями, – погано застосуємо в силу високої складності подібних систем.

Наразі стійкий інтерес проявляється до методів розробки та аналізу імітаційних моделей (ІМ) складних ДС. У якості базового формалізму для створення ІМ, орієнтованих на використання в складі ІСППР РЧ, пропонується апарат на основі мереж Петрі (МП). Попередній аналіз показав, що деякі модифікації МП, що використовують конструкції модульності й ієрархічності (так звані «МП високого рівня») є перспективним базисом для таких моделей.

Об'єктом дослідження є складні ДС, для моніторингу й керування якими використовуються ДІС типу ІСППР РЧ. Такі системи являють собою сукупність взаємодіючих компонентів, кожний з яких у будь-який момент часу перебуває в деякому стані.

Метою роботи є розробка методів і відповідних програмних засобів, що дозволяють створювати ІМ складних ДС, для керування й моніторингу яких використовуються ДІС типу ІСППР РЧ, на основі темпоральної модифікації РМП.

Для досягнення поставленої мети в дисертаційній роботі було потрібно розв'язати наступні науково-практичні завдання:

- аналіз існуючих підходів до моделювання систем і темпоральних залежностей і застосовності цих підходів для розробки й аналізу моделей складних ДС, що є об'єктом дослідження;
- розробка методів і алгоритмів функціонування ІМ складних ДС на основі апарата РМП із можливістю представлення й оперування темпоральною інформацією;
- розробка методів аналізу й верифікації моделей складних ДС, для моніторингу й керування якими використовуються ДІС типу ІСППР РЧ;

- програмна реалізація розроблених методів і алгоритмів у плані їх включення в програмне забезпечення ДС типу ІСППР РЧ;
- експериментальна апробація розроблених методів і інструментальних програмних засобів.

Для рішення поставлених завдань використані методи імітаційного моделювання, моделі й методи представлення часу (темпоральної логіки, вирахування), а також теорія автоматів, теорія мереж Петрі і їх модифікацій, методи розробки програмного забезпечення.

- запропонований формалізм РМП реального часу з підтримкою темпоральної логіки Аллена (РМП РЧ ТЛА), орієнтований на моделювання процесів у складних ДС, що й дозволяє (на відміну від базового формалізму РМП РЧ) використовувати в якості захисних функцій переходів мережі вираження логіки Аллена, що визначають темпоральні зв'язки між місцями мережі;

- запропоновані оригінальний метод моделювання ДС на основі РМП РЧ ТЛА й ряд відповідних алгоритмів – визначення допустимості переходу, розв'язання конфлікту переходу, спрацьовування переходу, що мають поліноміальну оцінку складності, що дозволяє їхнє застосування в ІСППР РЧ із досить твердими часовими обмеженнями;

- обґрунтована можливість застосування підходу на основі Model Checking для верифікації моделей процесів у ДС і запропонований алгоритм верифікації ІМ на основі РМП РЧ ТЛА, орієнтований на використання в ІСППР РЧ.

1 АНАЛІЗ ДОСЛІДЖЕНЬ В ГАЛУЗІ МОДЕЛЮВАННЯ ДИНАМІЧНИХ СИСТЕМ

1.1 Підходи до розробки динамічних інтелектуальних систем

У процесі інформатизації суспільства особливу роль відіграють інтелектуальні системи (ИС) – основний продукт досліджень у штучному інтелекті (ШІ). Сьогодні термін «інтелектуальний» став досить уживаним у контексті інформаційних технологій, однак його вживання далеке не завжди має однозначний сенс.

Для розуміння цілей і завдань напрямку ШІ необхідно уточнити феномен інтелекту природнього. Можна виділити ряд ключових здатностей суб'єкта, сукупність яких і формує його «інтелектуальність», у тому числі [1]: здатність виділяти істотне в наявних знаннях, тобто впорядковувати їх (необхідний аспект інтуїції); здатність до цілепокладання планування поведінки, тобто породження послідовностей «ціль → план → дія»; здатність до відбору знань (посилок висновків, релевантних мети міркування); здатність витягати наслідки з наявних знань, тобто здатність до міркування, яке може містити як правдоподібні висновки, використовувані для висування гіпотез, так і достовірні висновки, використовувані при доказах; здатність до аргументованого прийняття рішень, що використовує впорядковані знання (представлення знань) і результати міркувань, відповідні до поставленої мети; здатність до рефлексії – оцінці знань і дій; наявність пізнавальної цікавості, тобто суб'єкт, що пізнає, повинен бути здатний ставити запитання й шукати на нього відповідь; здатність і потреба знаходити пояснення як відповідь на запитання «чому?»; здатність до синтезу пізнавальних процедур, що утворюють евристику рішення завдань і розгляду проблем; здатність до навчання й використанню пам'яті; здатність до раціоналізації ідей: прагнення уточнити їх як поняття; здатність до створення цілісної картини предмета мислення, що поєднує знання, релевантні поставленої цілі (тобто створення деякої «теорії»); здатність адаптації в умовах зміни життєвих ситуацій і

знань, корекція «теорій» і поведінки. У роботі будемо дотримуватися комплексного визначення ІС, даного в роботі [1]: ІС – комп'ютерна система, яка є засобом автоматизації ключових здатностей природнього інтелекту. Архітектура такої ІС наведена на рис. 1.1.

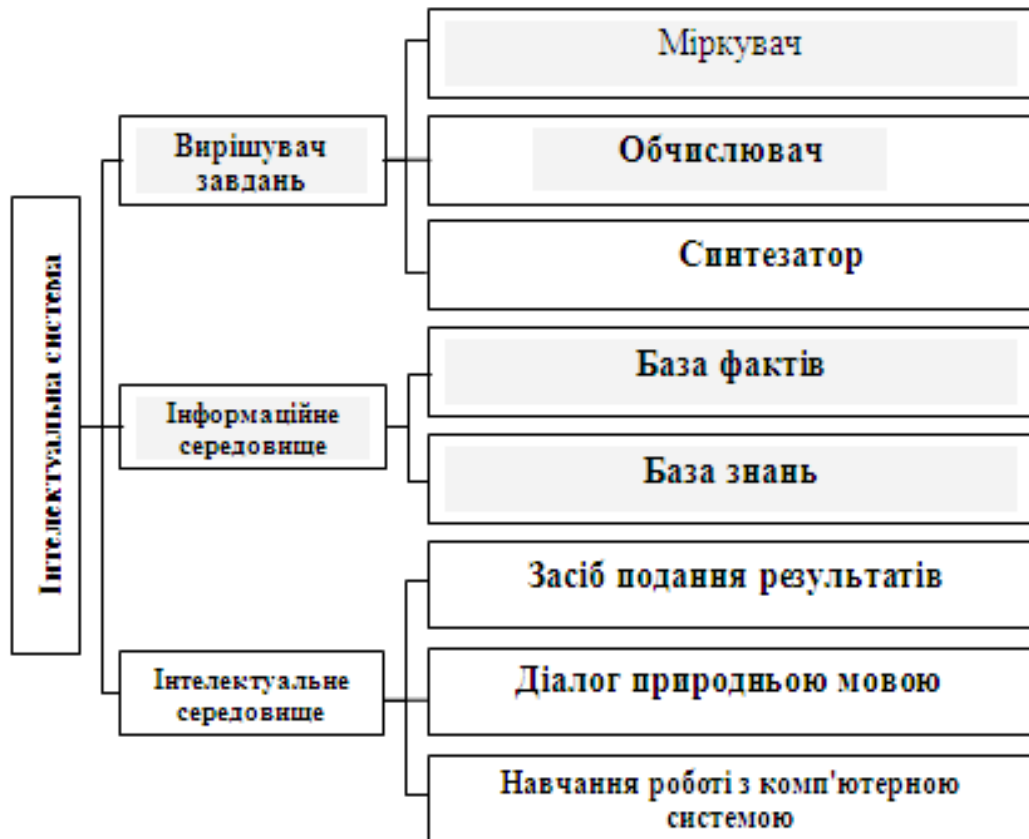


Рисунок 1.1 – Архітектура ІС згідно [1]

Найчастіше ІС застосовуються для рішення складних завдань, де основна складність рішення пов'язана з використанням слабо-формалізованих знань фахівців-практиків і де логічна (значеннєва) обробка інформації превалює над обчислювальною, наприклад: розуміння природньої мови; аналіз, діагностика й прийняття рішень у складні (проблемних) ситуаціях; навчання; аналіз візуальної інформації; оперативно-диспетчерське керування й ін.

1.2 Динамічні інтелектуальні системи як підклас динамічних систем

Відомо досить багато різних варіантів класифікації ІС, одним з найбільш використовуваних серед яких є розподіл ІС на статичні й динамічні [4, 5]. Для статичних ІС передбачається, що вихідна інформація й БЗ системи не міняються в процесі пошуку (висновку) рішення, для динамічних ІС ця вимога може не виконуватися.

Статичні ІС, зокрема статичні ІСППР, орієнтовані на пошук рішення в статичних предметні (проблемних) областях, характеристики яких не міняються в процесі функціонування ІС (процесі пошуку рішення). А динамічні ІС (ДІС) – на пошук рішення в динамічних областях, характеристики яких, як правило, змінюються в процесі пошуку рішення, і для яких необхідно враховувати фактор часу й часові (темпоральні) залежності. Відомо, що поняття ДС виникає, коли необхідно досліджувати розвиток системи в часі шляхом установалення взаємозв'язку між значеннями параметрів системи в різні моменти часу [6]. Формально ДС можна задати набором:

$$DS \equiv \langle T, X, U, \Omega, Y, \Gamma, n, \eta \rangle, \quad (1.1)$$

де T – множина моментів часу: деяке упорядковане підмножина множини речовинних чисел, $T \in \mathfrak{R}_0$;

\bar{X} – множина станів;

U – множина миттєвих значень вхідних впливів;

$\Omega = \{\omega \mid \omega: T \rightarrow U\}$ – непорожня множина припустимих вхідних впливів;

Y – множина миттєвих значень вихідних величин;

$\Gamma = \{\gamma \mid \gamma: T \rightarrow Y\}$ – \ множина вихідних величин;

$\varphi: T \times T \times X \times \Omega \rightarrow X$ – перехідна функція стану, значеннями якої служать стану $\varphi(\tau, \tau_0, x(\tau_0), \omega) = x(\tau) \in X$, у яких виявляється система в момент часу $\tau \in T$, якщо в початковий момент часу $\tau_0 \in T$ вона була в початковому стані $x(\tau_0) \in X$ і на неї діяв вхідний вплив $\omega \in \Omega$;

$\eta: T \times X \rightarrow Y$ – вихідне відображення, що визначає вихідні величини.

ДІС – це такі системи, які, з одного боку, задовольняють наведеному визначенню ДС, з інший, – їхнього динаміка описується правилами, часто використовуваними в ІС, а стану – темпоральними параметрами й змінн, що мають як кількісний, так і якісний (символьний) характер [13]. При дослідженні поведінки багатьох складних ДС (технічних, організаційно-технічних, соціальних, екологічних і ін.) і розробці ІС (ІСППР) для допомоги ОУР при керуванні такими ДС, необхідно використовувати підходи й методи ШІ. Існуючі в таких системах закони функціонування (поведінки) описуються найчастіше якісними параметрами й настільки складні, що не допускають (або роблять практично неприйнятним) застосування традиційного аналітичного представлення, але можуть задаватися за допомогою емпіричних або експертних правил. У зв'язку із цим класичні засоби моделювання на основі строгих математичних моделей є непридатними або неефективними (неадекватними реальній системі).

Слід зазначити, що сучасні дослідження в області ДІС спрямовані на створення як потужних універсальних, так і спеціалізованих інструментальних засобів різного призначення для промислової розробки ДІС у сфері комерційних і промислових додатків у різних галузях науки й техніки.

Згідно [7], можна виділити наступні основні типи ДІС:

- динамічні інтегровані експертні системи;
 - розподілені інтелектуальні системи, у тому числі побудовані за принципом багатоагентних систем;
 - інтелектуальні системи підтримки прийняття рішень (ІСППР).
- Представники різних типів ДІС мають свої відмітні ознаки (див., наприклад, [6]). При цьому їх поєднує можливість функціонування в динамічних предметні (проблемних) областях і необхідність обліку фактора часу й темпоральних залежностей. Зупинимось докладніше на найбільш складному підкласі ІСППР – ІСППР РЧ [6].

1.3 Інтелектуальні системи підтримки прийняття рішень реального часу

ІСППР РЧ – системи підтримки прийняття рішень, в основі яких лежать методи ШІ і які призначені для допомоги ОУР при керуванні складними об'єктами (ОК – об'єктами керування, ДС) і процесами різної природи в умовах досить твердих часових обмежень [2, 5]. В [2] запропоновано конструювати ІСППР РЧ як інтегровані ІС семіотичного типу, що поєднують строгі, формальні методи й моделі пошуку рішень із нестрогими, евристичними методами й моделями, що базуються на знаннях фахівців-експертів, моделях людських міркувань, імітаційних моделях, некласичних логіках і накопиченому досвіді [9].

Формально ІСППР РЧ семіотичного типу може бути задана набором [6, 12]:

$$SS = \langle M, R(M), F(M), F(SS) \rangle,$$

де: $M \equiv \{M_1, M_2, \dots, M_n\}$ – множина формальних або логіко-лінгвістичних моделей, що реалізують певні інтелектуальні функції;

$R(M): S \rightarrow M$ – множина правил вибору необхідної моделі або сукупності моделей у поточній ситуації, де S – множина можливих станів (яке може бути й відкритим), або деяка множина узагальнених ситуацій, наприклад, штатних, аномальних або аварійних, при влученні в які відбувається зміна моделі;

$F(M) \equiv \{F(M_1), F(M_2), \dots, F(M_n)\}$ – множина правил модифікації моделей M .

$F(SS)$ – правило модифікації властиво системи SS – її базових конструкцій $M, R(M), F(M)$ і, можливо, самого правила $F(SS)$;

$F(SS)$ реалізує ряд відображень $S \times M \rightarrow M', S \times R(M) \rightarrow R'(M), S \times F(M) \rightarrow F'(M), S \times F(SS) \rightarrow F'(SS)$, де $S' \cap S = \emptyset$, тобто правила модифікації даного типу використовуються в ситуаціях, коли наявних множин моделей, правил вибору й правил модифікації недостатньо для пошуку рішень у сформованій проблемній ситуації S' . Для модифікації $F(SS)$ можуть бути використані як внутрішні засоби, так і зовнішні метазнання, що відбивають прагматичний аспект проблемної ситуації.

У перерахованих моделях і правилах вибору присутній фактор часу, кожна з моделей системи орієнтована на обробку деякого типу невизначеності (Не-Фактора [10, 11]). Зокрема, моделі й методи, що використовують для оперування неповної й суперечливої інформації апарат наближених множин, ефективні в ситуаціях максимальної невизначеності, коли немає ніякої додаткової інформації про проблему [12].

У міру вступу додаткової інформації (і зменшення, таким чином, ступені невизначеності), наприклад, у вигляді ймовірностей для правил, може бути застосований інтегрований підхід на базі нечіткої або імовірнісної логік, що підвищує ступінь правдоподібності видаваних системою рекомендацій ОПР. При наявності повної інформації про проблемну ситуацію рішення можуть бути достовірними й отриманими в за допомогою математичних методів оптимізації.

Одним з головних завдань ІСППР РЧ при керуванні складними ОК типу ДС є допомога ОПР для втримання об'єкта в штатному (нормальному) режимі функціонування. Для цього необхідно реалізувати функції, обумовлені сучасними прийманнями керування в аномальних і критичних ситуаціях, на основі методів атракторів і теорії стійкості, когнітивної графіки, методів обробки невизначеності на основі нечітких і неklasичних логік і ін.

Узагальнена архітектура ІСППР РЧ як ДІС семіотичного типу з (рис. 1.2)

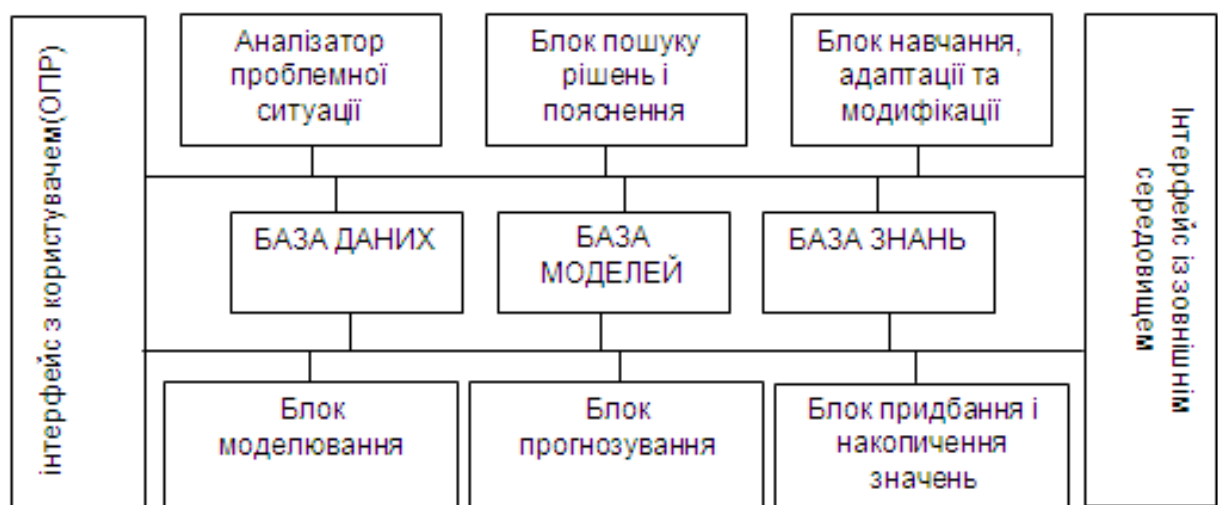


Рисунок 1.2 – Узагальнена архітектура ІСППР РЧ

розподіленим інтелектом, що полягає із взаємодіючих між собою інтелектуальних блоків (модулів, агентів), що виконують відповідні інтелектуальні функції, представлена на рис. 1.2 [5].

ІСППР РЧ включає ряд традиційних для експертних систем блоків: базу даних (БД), базу знань (БЗ), блоки аналізу ситуації, пошуку рішень і пояснення, придбання й нагромадження знань, адаптації й модифікації. Крім цього, в ІСППР РЧ присутні база моделей, а також блоки моделювання й прогнозування, які використовуються для аналізу наслідків прийнятих рішень і вибору найкращих рекомендацій для ОПР.

Блок моделювання моделює поведінка ОК. Він може виступати як агент даних про стан об'єкта на етапах тестування системи й властиво прийняття рішень для порівняння з даними, що надходять від датчиків.

Блок прогнозування здійснює функції прогнозування аномальних ситуацій і наслідків керуючих впливів. Прогнозування здійснюється по команді ОПР на основі даних про поточний стан ОК, що надходять із БД, і знань, що зберігаються в БЗ, із застосуванням методів як достовірного, так і правдоподібного висновку.

Блок моделювання також може бути використаний разом із блоком прогнозування для прогнозу аномальних ситуацій і наслідків керуючих впливів. Базу моделей і блок моделювання можна виділити в окрему підсистему моделювання.

Таким чином, при розробці перспективної ІСППР РЧ необхідно включити до її складу модель ОК (одну або трохи). У загальному випадку ОК задовольняють визначенню ДС . У роботі розглядається завдання моделювання складних ДС, для моніторингу й керування якими призначені ІСППР РЧ, з необхідністю обліку темпоральних залежностей.

1.4 Завдання моделювання складних динамічних систем

Віднесення об'єкта (системи) до розряду «простих» або «складних» досить умовно й визначається тими завданнями, які коштують перед дослідником.

Розглянемо визначення складної ДС на прикладі ДІС. Як було показано раніше, ДІС орієнтовані на імітацію різних здатностей природнього інтелекту. Завдання, обумовлені цієї глобальною метою, на сьогоднішній день є одними з найбільш складних завдань в області інформаційних технологій. Крім того, як видно з рис. 1.1 і 1.2, структура ДІС задовольняє класичному визначенню складної системи як «системи багаторівневою конструкцією, що є, із взаємодіючих елементів, поєднаних у підсистеми різних рівнів» [4]. Таким чином, більшість ДІС є складними системами.

Технічні об'єкти й системи, для керування й моніторингу яких використовуються ІСППР РЧ, у загальному випадку також можна віднести до складних ДС. Можна виділити наступні ключові особливості таких систем:

- багатокomпонентність: система являє собою сукупність незалежних компонентів, кожний з яких виконує певну функцію;
- мережева організація: взаємодія компонентів один з одним здійснюється шляхом передачі даних (повідомлень);
- необхідність швидкого (оперативного) прийняття рішень у ситуаціях, які розглядаються як провідні до порушень правильності функціонування систем (тобто до аномальних або аварійним ситуаціям).

Також можна виділити ряд властивостей систем [5, 7], які зручно представити у вигляді дихотомій:

- динамічність – статичність: динамічна система, як відзначалося раніше, характеризується тим, що їх вихідні параметри в цей момент часу визначаються характером вхідних впливів колись і тепер (докладніше такі системи описані в попередньому розділі). А якщо ні, то системи називають статичними;

– стаціонарність – нестаціонарність: якщо параметри системи не змінюються в часі, то вона називається стаціонарною, у протилежному випадку система називається нестаціонарною;

– лінійність – нелінійність: для лінійної системи реакція на суму двох іди більш різних впливів еквівалентна сумі реакцій на кожне збурювання окремо, для нелінійної – це умова не виконується;

– безперервність – дискретність: стану системи можуть мінятися безупинно або мати дискретний характер, відповідно, по цих ознаках системи діляться на безперервні й дискретні;

– детермінованість – стохастичність: система, на функціонування якої істотний вплив виявляють випадкові фактори, що обурюють, і впливу, називають стохастичними. Якщо передбачається, що поведінка системи можна описати однозначно або можна ігнорувати вплив випадкових факторів, то така система є детермінованою.

Складні ДС, для керування й моніторингу яких використовуються ІСПП РЧ, у загальному випадку є динамічними нестаціонарними нелінійними безперервними детермінованими системами.

1.5 Аналіз підходів до моделювання складних динамічних систем

Моделювання процесів, що протікають у складній ДС, є ефективним способом переконатися в правильності й надійності функціонування такої системи [5]. Згідно [12], математичне моделювання – це опосередковане практичне або теоретичне дослідження об'єкта, при яким безпосередньо вивчається не сам об'єкт, а деяка допоміжна формальна модель (ФМ), що перебуває в деякій об'єктивній відповідності з пізнаваним об'єктом, здатна заміщати його в певних відносинах, що й дає при її дослідженні, в остаточному підсумку, інформацію про самий модельований об'єкт. Таким чином, ФМ – це

спрощений образ досліджуваного явища, записаний за допомогою формальної символіки. Подібний підхід дозволяє розробляти системи, опираючись на вже розроблені теоретичні викладення, що суттєво полегшує процес. Очевидно, що вибір ФМ визначає якість моделювання й аналізу процесів. При виборі ФМ для розглянутих складних ДС будемо орієнтуватися на наступні класи моделей:

- функціональні моделі, що відображають закономірності функціонування об'єктів і використовувані при проектуванні складних систем для дослідження принципів їх роботи й характеру процесів, що протікають;

- структурні моделі, що відбивають структурні характеристики й використовувані при конструюванні систем у ситуаціях, коли важливим є взаємодія окремих компонентів між собою.

Очевидно, що гібридні (інтегровані) моделі, що поєднують можливості обох перерахованих, можуть мати суттєво більшу виразність. Крім того, для виявлення структурних закономірностей системи (об'єкта) і відображення їх людині (ОПР, експертові) необхідно забезпечити максимальну візуальну виразність моделі.

Інша класифікація ФМ виділяє аналітичні й імітаційні моделі [4,5]. Аналітичні моделі являють собою рівняння або системи рівнянь: алгебраїчні, інтегральні, диференціальні, кінцево-різницеві й інші. Такі моделі, як правило, статичні й підходять для ідеалізованих завдань і об'єктів, що володіють високою спільністю. Даний тип моделей звичайно застосовують для опису фундаментальних властивостей об'єктів. Складні ДС (об'єкти) рідко вдається описати аналітично, оскільки вступає в дію сформульований Л. Заді принцип несумісності: «Чим складніше система, тем менш ми здатні дати точні й у той же час, що мають практичне значення судження про її поведінку. Для систем, складність яких перевершує деякий граничний рівень, точність і практичний зміст стають, що майже виключають один одного характеристиками» [15].

Альтернативою аналітичним моделям є імітаційні моделі (ІМ) [4]. Основна відмінність ІМ від аналітичних моделей полягає в тому, що замість аналітичного опису взаємозв'язків між входами й виходами досліджуваної системи будують модель (алгоритм) послідовність, що відображає, розвитку процесів усередині

досліджуваного об'єкта, а потім «програють» поведінка об'єкта на комп'ютері. ДО НИХ прибігають тоді, коли об'єкт моделювання настільки складний, що адекватно описати його поведінка математичними рівняннями неможливо або дуже складно. ІМ дозволяє розкласти більшу модель на частині (підсистеми), якими можна оперувати окремо й будувати на їхній основі інші, більш прості або, навпаки, більш складні моделі. Досить докладна ІМ системи або керованих з її допомогою об'єктів, що адекватно відображає їх функціонування, сама по собі представляє ДС.

1.6 Постановка задач дослідження

У контексті аналізу процесів, що протікають у ДС, раціонально використовувати ІМ. Враховуючи слабку формалізованість «інтелектуальних» процесів, впливає при необхідності також прибігати до евристичних методів, підходів на основі експертних знань. Перспективний підхід, що базується на інтеграції різних моделей. Варто відзначити, що моделювання таїть у собі можливість прорахунків. Недооблік реальних особливостей існування системи може привести до фатальних для цієї системи наслідків або, навпаки, породити чисто віртуальні системи, практично нереалізовані або не мають практичного змісту, що часто зустрічається в практиці моделювання. Тому необхідно випробування (апробація) створеної моделі для підсумкової оцінки результатів.

У контексті даної роботи ДС розглядаються, насамперед, як інформаційні системи, тому на узагальненому рівні можна виділити наступні процеси:

- уведення інформації із зовнішніх або внутрішніх джерел;
- обробка вхідної інформації й представлення її в зручному для моделювання виді;
- висновок інформації для представлення ОПР або передачі в іншу систему;

– зворотний зв'язок – передача інформації, переробленої людьми або іншою системою, для корекції вхідної інформації й рекомендованих рішень.

У залежності від завдання дослідження й характеристик складної ДС можуть бути виділені інші, більш детальні (локальні) підпроцеси, що визначають функціонування частини системи. Найчастіше немає необхідності в створенні повної докладної моделі всієї ДС – для дослідження певного процесу досить побудувати модель взаємодії декількох компонентів системи, залучених у цей процес.

Для реалізації підсистеми моделювання, що дозволяє аналізувати й верифікувати складні ДС, пропонується розробити базову ІМ – її математичне й програмне забезпечення, – з метою наступного включення створеної моделі до складу ІСППР РЧ. У роботі пропонується підхід на основі графічних моделей, що інтегрує можливості імітаційного й евристичного (експертного) моделювання.

2 ОПИС ПРОВЕДЕНИХ ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

2.1 Аналіз підходів до представлення й моделювання часу в складних динамічних системах

Про важливість наявності засобів представлення часу й темпоральних залежностей у ДІС говориться практично з моменту їх появи [9,10]. Облік часового фактора необхідний при рішенні завдань діагностики й моніторингу складного об'єкта або процесу, планування дій для досягнення поставленої мети, прогнозування наслідків прийнятих рішень, керування в реальному часі й навчання ОПР [11]. Тому завдання обробки часове (темпоральної) інформації й оперування темпоральними залежностями є ключовою при моделюванні складних ДС, для керування й моніторингу якими застосовуються ІСППР РЧ.

Темпоральні залежності можуть бути як кількісними (метричними), якщо для представлення часу використовуються кількісні заходи на часовій осі, так і якісними, якщо використовується тільки відносне положення в часі подій або дій [12].

Моделювання залежностей першого типу звичайно не викликає утруднень, однак при цьому суттєво обмежуються можливості розроблювача моделі й ОПР, тому що складно прямо відобразити темпоральні залежності між різними об'єктами системи, а також урахувати різного типу невизначеності (Не-Фактори або «зашумлені» дані), які властиві більшості ДІС. Використання ж якісних залежностей вимагає включення в модель засобів темпоральних логік. Для моделювання процесів складних ДС і ДІС типу ІСППР РЧ необхідно забезпечити можливість використання обох підходів.

Сучасні методи представлення темпоральних залежностей можна розбити на два основні класи – засновані на моделюванні змін у часі й засновані на явнім моделюванні часу [13] (рис. 2.1).

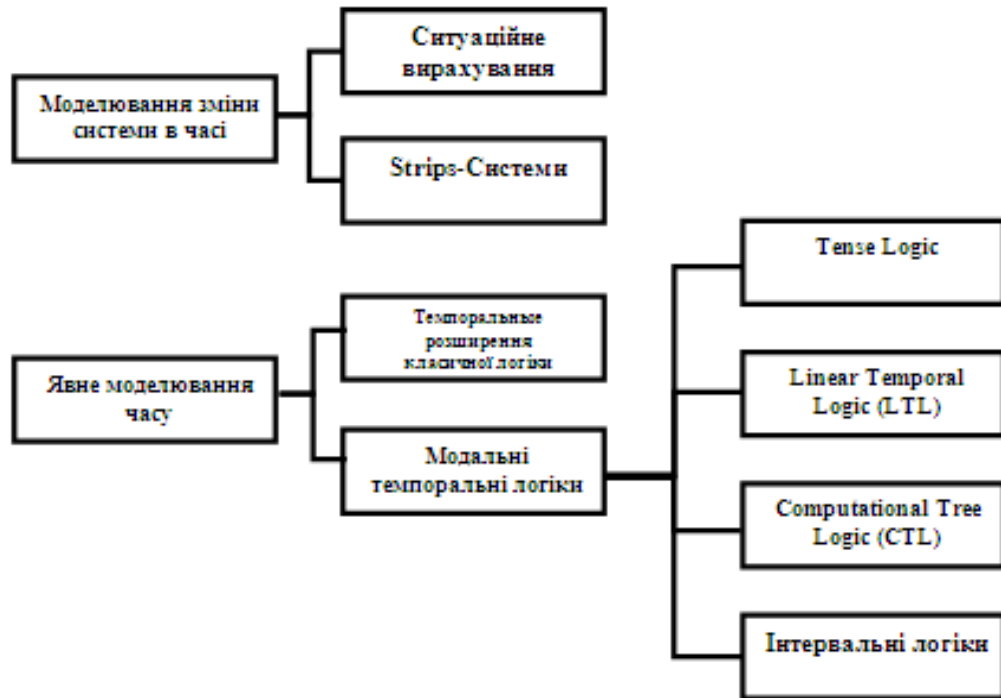


Рисунок 2.1 – Підходи й методи представлення темпоральних залежностей

З відомих підходів у плані моделювання (представлення й оперування) темпоральних залежностей можна виділити керування в часі, інтелектуальне імітаційне моделювання на основі системи РДО [12], динамічну версію інструментального комплексу «АТ-ТЕХНОЛОГІЯ», гібридні нечітко-темпоральні моделі часових рядів [13]. Зокрема, модель керування в часі, запропонована в [14] у контексті розробки перспективних ДІС, пов'язана з динамічним керуванням процесами в часі. У систему включаються правила, в антецеденті яких у тому або іншому виді може бути представлений поточний час системи. Така модель орієнтована на компенсацію збурювань у системі. Вона досить просто реалізована й використовує алгоритми практично прийнятної (поліноміальної) складності, але досить абстрактна, зокрема, для використання в ІСППР РЧ для моніторингу й керування конкретним об'єктом.

У рамках моделювання змін системи в часі час безпосередній не розглядається: у фокусі уваги перебувають зміни стану проблемної області, а прив'язка до моментів часу вторинна. При такому підході звичайно виділяються так звані індикатори змін, поява яких у системі ініціює виконання деяких дій.

Методи, засновані на явним моделюванні часу, характеризуються більшими виразними можливостями. Методи й моделі цього класу можна розрізнити по способу введення фактора часу. При цьому використовуються темпоральні логіки – логіки, у яких істинне значення логічних формул залежить від моменту або інтервалу часу, для якого обчислюються значення цих формул. Основу таких логік становлять або модальні логіки [15], якщо час ураховується в семантиці шляхом ускладнення інтерпретації, або модифікації логіки першого порядку, якщо час ураховується в синтаксисі. Алгоритми висновку в темпоральних логіках, що є розширенням логіки предикатів першого порядку, можуть базуватися на модифікаціях класичних алгоритмів висновку в цих логіках (наприклад, темпоральної резолюції), але ефективність цих алгоритмів досить низька для можливості їх практичного застосування в ІСППР РЧ при моделюванні процесів у складних ДС [14].

У темпоральних логіках у якості часових примітивів використовуються моменти (у крапкових логіках) або інтервали (в інтевальних логіках). Крапкові логіки – досить потужний спосіб опису подій і їх зв'язків. Даний підхід оперує поняттями часових крапок і відносин (« до», «послу», «одночасно») між ними. Час описується крапками, що лежать на часовій осі, яка відображає потік часу. До числа найбільш відомих можна віднести логіку Пріора (Tense Logic), лінійну темпоральну логіку (Linear Temporal Logic, LTL), логіка дерев обчислень (Computational Tree Logic, CTL).

Існує велика кількість модифікацій крапкових темпоральних логік, у тому числі, застосованих при моделюванні часу в ДС [13]. Головний недолік подібного підходу полягає в тому, що, що оперують моментами логіки недостатньо чутливі до значеннєвих відмінностей між дієслівними предикатами, тому що останні виражають інформацію про стан справ не тільки в той або інший момент часу, але й на тому або іншому часовому інтервалі (наприклад, «бути раніше», «не пізніше», «одночасно» і т.п.).

На відміну від крапкових логік, інтервальні логіки оперують часовими інтервалами (періодами) – відрізками часу, що мають деяку тривалість. Інтервали

можуть бути прив'язані до часової осі (тобто із зазначеними моментами початку й кінця), а можуть бути описані тільки тривалістями й відносинами. Стандартний опис інтевальної моделі (інтервальної логіки) було дано Дж. Алленом у вигляді темпоральної логіки Аллена (ТЛА) [16], активно використовуваної в ІС у силу її практичної реалізованості, зокрема, в ІСППР РЧ [16].

Базовими конструкціями ТЛА є наступні:

- часовий інтервал X – це впорядкована пара $\langle X_-, X_+ \rangle$, така, що виконується умова $X_- < X_+$, де X_- і X_+ розглядаються як моменти часу (наприклад, на речовинній осі);
- інтервальна інтерпретація (*I*-Інтерпретація) – функція, що відображає часовий інтервал на числову вісь із обов'язковим виконанням умови $X_- < X_+$;
- множина базисних інтервальних відносин B , обумовлене через відносини між кінцями інтервалів;
- кінцева множина інтервальних формул Θ . множина Θ *I*-Здійсненне, якщо існує *I*-Інтерпретація (називана *I*-Моделлю Θ), яка виконує кожну формулу Θ . Якщо інтервальна формула здійсненна кожною *I*-Моделлю множини формул Θ , то F логічно випливає з $(\Theta \rightarrow F)$;

Основні відносини між темпоральними інтевалами в ТЛА

$$B = \{b, b_i, m, m_i, o, o_i, d, d_i, s, s_i, f, f_i, e\}.$$

ТЛА лягла в основу подальших розробок в області інтервального представлення часу. ТЛА характеризується достатньою виразністю й наявністю поліноміальних алгоритмів висновку, що дозволяє використовувати її в завданні моделювання процесів у складних ДС. До недоліків ТЛА звичайно відносять складність представлення часових крапок, що часто необхідно для природнього людського сприйняття (і для моделювання людських міркувань). Простого об'єднання понять моментів і інтервалів недостатньо через «проблему граничних крапок», для рішення якої був розроблений підхід точково-інтервального представлення, здатного описувати часові крапки, інтервали й відносини між ними [17].

2.2 Вибір формального апарата імітаційного моделювання

На підставі результатів проведеного аналізу можна зробити висновок, що для рішення комплексних завдань моделювання, дослідження, аналізу й верифікації складних ДС доцільно використовувати підхід на основі імітаційного моделювання. Основна складність полягає в адекватному представленні часу й темпоральних залежностей між об'єктами складних ДС. При цьому для рішення завдання аналізу й формальної верифікації системи необхідно використовувати досить потужний формальний апарат. Розробка абсолютно нового формалізму сполучена зі значними труднощами й не є необхідною, оскільки існує достатня кількість відомих формальних засобів для рішення близьких завдань [4, 6,10,12]. Тому ефективним шляхом є вибір деякого існуючого базисного формалізму і його наступна модифікація з обліком сформульованих раніше вимог для рішення поставлених завдань із метою розробки перспективного математичного й програмного забезпечення для моделювання й аналізу складних ДС, а також включення цих засобів до складу ДІС типу ІСППР РЧ.

На підставі проведеного аналізу можна виділити ряд бажаних характеристик, якими повинен мати математичний формалізм, використовуваний для НІМ процесів у складних ДС:

- комбінація функціонального й структурного підходів;
- візуальна виразність;
- підтримка роботи з темпоральними залежностями обох типів – кількісними і якісними;
- можливість використання засобів темпоральної логіки;
- можливість об'єднання моделей фрагментів системи для комплексного аналізу.

Одним з найбільш важливих вимог є можливість структурного представлення моделі, тому пропонується орієнтуватися на формальний апарат на основі графів – графоорієнтованого засобу моделювання, що підтримують також

візуальну виразність. У якості базового формалізму розглянемо кінцеві автомати, часові автомати й мережі Петрі.

У якості формальних моделей для складних систем або їх складових часто використовують автоматні моделі з Діскретним часом [59]. Такий автомат перетворить інформацію із кроків у Діскретні моменти часу й формує результуючу інформацію відповідно до заданого алгоритму. Перетворення можливі за допомогою технічних і (або) програмних засобів.

Автомат можна інтерпретувати як деяке обладнання (чорний ящик), на яке подаються вхідні сигнали, знімаються вихідні сигнали і яке може мати певні внутрішні стани.

Кінцевий автомат (КА) має один вхід, один вихід і являє собою об'єкт, що функціонує в Діскретні моменти часу. У кожний момент часу t автомат перебуває в одному з можливих станів W , число яких передбачається кінцевим.

Робота КА полягає в тому, що він реалізує деяке відображення множини слів вхідного алфавіту X у множина слів вихідного алфавіту Y , інакше кажучи, здійснює перетворення вхідних слів у вихідні.

За характером відліку часу КА поділяються на синхронні й асинхронні. Автомат вважається синхронним, коли моменти вступу вхідних сигналів, зміни станів і видачі вихідних сигналів визначаються примусово синхронізуючими сигналами (заздалегідь визначеними). Реакція КА на кожне значення вхідного сигналу закінчується за один такт синхронізації. Асинхронні автомати не мають «твердої» тактності, вони змінюють свої стани при вступі вхідних сигналів, які можуть з'являтися в довільні моменти часу із заданого інтервалу.

КА як засіб для формального опису детермінованих об'єктів з дискретним часом знаходять широке застосування при аналізі алгоритмів і в програмуванні. Поняття класичного КА пов'язане з поняттями алгоритму й послідовної алгоритмічної системи, для яких характерний послідовний спосіб функціонування: система (автомат) послідовно переходить зі стану в стан відповідно до заданої функції переходу й здійснює черговий (послідовний) крок алгоритму. Звичайний (стаціонарний) автомат має функції переходів і виходів, які

не залежать явно від часу. Увага до таких автоматів зложилося історично як до моделей реальної апаратури, що працює в стаціонарних умовах.

Інший тип детермінованих систем з Діскретним часом, що узагальнює поняття КА, – нестаціонарні КА. При моделюванні ДС і ДіС переважніше динамічна модель, для якої функції переходів і виходів можуть явно залежати від часу. Це пов'язане з мінливістю умов функціонування (зміна факторів зовнішнього середовища, витрата ресурсів т. буд.) системи (об'єкта), тобто з невиконанням властивості стаціонарності. За допомогою нестаціонарних КА моделюються, наприклад, системи часової й просторової комутації в техніку обміну інформацією. Однак широта застосування КА не означає їхньої універсальності.

Апарат КА призначений, насамперед, для моделювання окремих процесів. При моделюванні ж складних ДС необхідно враховувати не тільки самі процеси, але і їх взаємодія. Частково цю проблему дозволяють розв'язати мережі кінцевих автоматів, тобто схеми, які виходять у результаті приєднання виходів одних автоматів до входів інших [19]. Розрізняють три типи з'єднання КА: послідовне з'єднання, паралельне та з'єднання зі зворотним зв'язком. При цьому необхідно дотримувати двох умов коректності схеми:

- у будь-який момент часу на кожний вузол мережі (як зовнішній, так і внутрішній) надходить який-небудь елементарний сигнал;
- неоднозначність сигналів у якому-небудь вузлі мережі хоча б в один момент часу є неприпустимою.

Передбачається, що всі вхідні в композицію автомати мають той самий структурний алфавіт і працюють у тому самому автоматному часі. Під структурним вхідним алфавітом автомата розуміється набір можливих значень сигналів, що подаються на один зовнішній вузол автомата. У цей час найпоширеніший двійковий алфавіт.

Такий підхід теоретично дозволяє представити процес функціонування окремого компонента складної ДС як КА, а потім об'єднати такі КА в єдину мережу, відбивши деякі особливості взаємодії процесів. Однак центральним

завданням структурної теорії автоматів, що розглядає подібні мережі, є завдання структурного синтезу, тобто пошук композиції елементарних автоматів, яка еквівалентна вихідному КА. При цьому аналіз і верифікація окремих КА і їх взаємодії звичайно не розглядається.

Правила побудови коректних схем і необхідність використання єдиного алфавіту накладають істотні обмеження на процес розробки моделей при використанні мереж КА в якості моделей складних ДС. Даний апарат також погано придатний, зокрема, для моделювання процесів прийняття рішень у ДС при наявності стохастических елементів. Класичний апарат КА й мереж КА не є ефективним засобом для розробки ІМ складних ДС, для моніторингу й керування якими застосовуються ДІС типу ІСППР РЧ.

2.3 Моделі на Мережах Петрі

У міру ускладнення систем усе більша увага залучають «неалгоритмічні» паралельні системи з недетермінованою поведінкою, у яких окремі компоненти функціонують, в основному, незалежно, взаємодіючи один з одним час від часу.

Серед багатьох існуючих методів опису й аналізу паралельних систем існує підхід, який заснований на використанні мережевих моделей, що сходять до мереж спеціального виду, запропонованим К. Петрі для моделювання асинхронних інформаційних потоків у системах перетворення даних [17].

Мережі Петрі (МП) давно зарекомендували себе як зручний, наочний, і в те ж час математично строгий формалізм для моделювання й аналізу складних систем. МП дозволяють із достатнім ступенем деталізації й візуалізації моделювати обчислювальні процеси, протоколи взаємодії, а також процеси керування. Зручність даного формалізму продемонстроване на великій множині практичних додатків. МП дозволяють природно описувати синхронізацію,

паралелізм, конфлікт і причинну залежність, а також наочно представляти структуру й функціонування складних систем, включаючи ДС.

Основи теорії звичайних МП викладені, зокрема, у роботі [18]. МП задається як кортеж:

$$PN \equiv \langle P, T, F, w, m_0 \rangle,$$

Графічним представленням МП служить двочастковий орієнтований граф с двома типами вершин: вершини-місця позначаються колами вершини-переходи – формально за допомогою множини послідовностей спрацьовування й множини досяжних у мережі маркувань. Ці поняття визначаються через правила спрацьовування переходів мережі.

МП цікаві, насамперед, тим, що вони дозволяють представляти й вивчати в динаміці поведінку системи паралельних процесів у будь-якому дискретному обладнанні або в програмі. Існує можливість застосування даного формалізму для рішення досить складних теоретичних завдань із області ШІ. Однієї з перших робіт, присвячених цьому питанню, є стаття [7], у якій обґрунтовується раціональність використання МП для паралельного дедуктивного висновку. Впливає, однак, відзначити, що рішення подібних «інтелектуальних» завдань за допомогою класичного формалізму МП найчастіше вимагає побудови вкрай громіздких моделей. Тому активно розвивається напрямок, пов'язане з розробкою різних модифікацій МП, що базуються на класичних МП, що й володіють додатковими властивостями, необхідними для ефективного рішення певних класів завдань. Такі МП високого рівня [18] характеризуються звичайно наступними особливостями:

- маркування мережі задається за допомогою індивідуальних, тобто помітних між собою фішок;
- місця можуть містити комплекти індивідуальних фішок; розмір комплекту визначається арністю місця;
- переходи можуть спрацьовувати в різних режимах, видаляючи фішки з одних позицій і додаючи їх в інші, при цьому єдиним апіорним обмеженням є

вимога локальності, тобто в будь-якому режимі перехід може видаляти фішки тільки зі своїх вхідних місць і додавати тільки у вихідні.

Відомо досить велика кількість робіт, присвячених подібним модифікацій МП, серед яких можна виділити, зокрема, інгібіторні МП [62], рекурсивні МП [15], ієрархічні МП [18], розширення МП за рахунок об'єктно-орієнтованих конструкцій (абстракція, інкапсуляція, спадкування й ін.) з метою одержання структурованих моделей, що явно відбивають ієрархічну й мультиагентну структуру системи.

Відмінною рисою вкладених МП є те, що фішки, що позначають позиції, розглядаються як об'єкти, що мають самостійну поведінку, яка, у свою чергу, описується також деякими МП. Назва «вкладені МП» указує на те, що елементи мереж у них самі є МП, аналогічно тому, як у системі вкладених множин елементами деякої множини можуть бути множини. Вкладені МП є зручним та потужним засобом для моделювання й аналізу ієрархічних мультиагентних систем [19]. Вони мають природній механізм модульності. Як і більшість інших модифікацій, цей тип МП орієнтований на аналіз статичних властивостей модельованих систем, що, безумовно, є важливим завданням. Однак відсутність представлення часу підтримки роботи з темпоральними залежностями накладає істотні обмеження на використання подібних формалізмів у якості базових для імітаційного моделювання складних ДС.

Інший формалізм на основі класичних МП – розфарбовані МП (РМП) (Coloured Petri Nets, CPN), апарат яких розробляється робочою групою під керівництвом К. Йенсена [20]. РМП – це графоорієнтована мова для проектування, опису, імітації й контролю розподілених і паралельних систем. Графічними примітивами показується плин процесу, а конструкціями спеціальної мови імітується необхідна обробка даних. Приведемо деякі формальні визначення згідно [3].

Нехай задана мова L типізованих виражень, які побудовані зі змінних і констант із використанням тільки операції додавання комплектів, деяка кінцева модель цієї мови U , елементами якої є індивідуальні (помітні) фішки.

Формально РМП можна визначити як кортеж:

$$CPN \equiv \langle \Sigma, P, T, F, \xi, \dot{\gamma}, \varepsilon, m_0 \rangle,$$

$\varepsilon: \dot{F} \rightarrow \dot{L}$ – функція виражень дуг, що зіставляє дугам вираження мови \dot{L} , такі, що тип вираження дуги повинен збігатися з типом того місця, з яким зв'язана ця

У РМП кратність дуг переноситься із графового представлення на рівень виражень ε мови \dot{L} . Для кожної РМП можна побудувати звичайну МП, і навпаки. Крім того, існує можливість трансляції вкладених МП у РМП [11]. РМП являє собою більш компактну й зручну мову моделювання, ніж звичайні МП (аналогічно тому, що мови програмування високого рівня більш зручні для практичного програмування, ніж мови рівня асемблерів).

Для практичного використання РМП розроблена система CPN Tools [20], що забезпечує потужну інструментальну підтримку імітаційного моделювання й аналізу поведінкових властивостей. Інструментарій CPN Tools складається із двох компонентів: графічного редактора, написаного мовою ВЕТА [21], і базового симулятора, розробленого за допомогою модульної функціональної мови Standart ML [22]. Ця система підтримує розробку й аналіз моделей на основі класичних МП і РМП, причому реалізований механізм часових затримок фішок, що дозволяє моделювати кількісні темпоральні залежності. Відсутність підтримки роботи з якісними затримками й темпоральними залежностями ускладнює використання цього формалізму для моделювання складних ДС. Однак високий рівень візуальної виразності дозволяє розглядати апарат РМП як основу для подальших досліджень у плані використання в ДС типу ІСППР РЧ.

3 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

3.1 Аналіз методів моделювання процесів у складних динамічних системах

Для створення ефективних ІМ процесів функціонування ДС необхідно визначити формалізм, на основі якого буде проводитися моделювання. За результатами досліджень, наведеним у попередній главі, прийняте рішення про використання в якості базового формалізму апарата РМП, що має гарні засоби візуалізації. У даному розділі пропонується модифікація цього апарата, що дозволяє ефективно працювати з темпоральною інформацією при моделюванні ДС, а також методи аналізу й верифікації моделей, розроблених на основі пропонованого формалізму (рис. 3.1)..

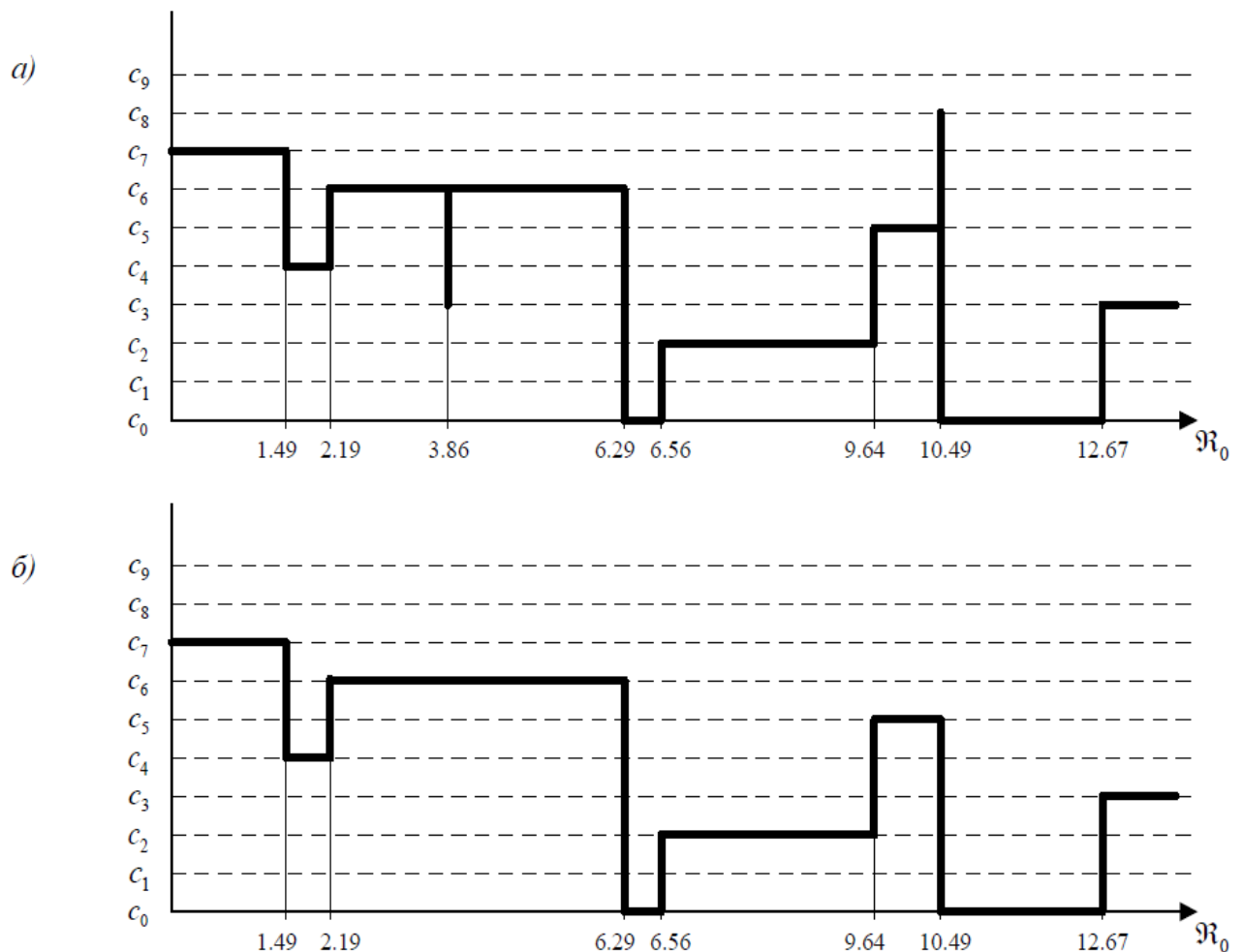


Рисунок 3.1 – Приклад реалізації процесу

У цілому систему складних ДС, для моніторингу й керування якими використовуються ІСППР РЧ можна назвати агрегативною [4], тобто зображуваною у вигляді сукупності компонентів-агрегатів, кожний з яких у будь-який момент часу (множина моментів часу $T = t_0$) перебуває в одному з можливих станів. Для опису розглянутих далі систем будемо використовувати термін «А-Система».

Реалізацією А-Процесу в А-Системі \dot{S} є функція $\dot{S} \rightarrow \dot{R}$. Графічно реалізацію процесу в компоненті А-Системи можна представити у вигляді двійкової часової діаграми

а) у компоненті А-Системи:

$$r = \langle \langle c_7, 0.0 \rangle, \langle c_5, 1.49 \rangle, \langle c_4, 1.49 \rangle, \langle c_6, 2.19 \rangle, \langle c_3, 3.84 \rangle, \langle c_4, 3.86 \rangle, \langle c_6, 3.86 \rangle, \langle c_0, 6.29 \rangle, \langle c_2, 6.56 \rangle, \langle c_5, 9.64 \rangle, \langle c_1, 10.49 \rangle, \langle c_0, 10.49 \rangle, \langle c_8, 10.49 \rangle, \langle c_6, 10.49 \rangle, \langle c_0, 10.49 \rangle, \langle c_3, 12.6767 \rangle \rangle.$$

б) у компоненті реальної А-Системи:

$$r = \langle \langle c_7, 0.0 \rangle, \langle c_4, 1.49 \rangle, \langle c_6, 2.19 \rangle, \langle c_0, 6.29 \rangle, \langle c_2, 6.56 \rangle, \langle c_5, 9.64 \rangle, \langle c_0, 10.49 \rangle, \langle c_3, 12.67 \rangle \rangle.$$

3.2 Модифікації мереж Петрі для моделювання процесів

Як відзначалося раніше, основним недоліком класичних МП є відсутність можливості урахування фактора часу й темпоральних залежностей. Це не дозволяє ефективно моделювати ті процеси, у яких від поточного часу залежить стан аналізованої системи – тобто більшість процесів, для аналізу, керування й моніторингу яких призначені ДІС типу ІСППР, а також процесів у самих ДІС. Тому значна частина досліджень в області розвитку й модифікації апарата МП присвячена розробці різних темпоральних розширень. Серед таких модифікацій можна виділити наступні основні групи:

- часові МП, у яких певним елементам мережі (місцям, переходам, фішкам) ставляться у відповідність деякі темпоральні характеристики (часові мітки, часові затримки), які визначають можливість спрацьовування переходів;
- стохастичні МП, засновані на концепції стохастичних часових затримок;
- функціональні МП, для яких часові затримки визначаються як функції дуг і переходів мереж.

Часові й стохастичні МП є окремими випадками функціональних МП, оскільки функції часових затримок дозволяють задати як детерміновані й недетерміновані затримки для першого підкласу, так і стохастичні затримки для другого.

Формально апарат РМП не орієнтований на роботу з темпоральною інформацією, однак інструментарій CPN Tools [21] підтримує простий клас часових МП, що допускає використання часових затримок на фішках. Кожна фішка має свою часову мітку t . Для моментів модельного часу, менших t , фішка не обробляється системою моделювання, а перебуває в так званім недійсному стані й не бере участь у спрацьовуванні переходів. Після моменту часу t фішка «прокидається» і бере участь у спрацьовуванні переходів. Такий спосіб представлення часу дозволяє миттєве спрацьовування переходів. Подібний підхід ефективний, якщо при моделюванні передбачається відповідність кожної фішки деякому ресурсу з певними темпоральними обмеженнями. Однак при створенні моделей процесів у складних ДС раціонально використовувати фішки як індикатори стану певного об'єкта системи. Крім того, проста підтримка часових міток робить у край складним завдання врахування темпоральних залежностей між об'єктами моделі. Тому підхід до обробки темпоральної інформації, використовуваний у РМП, неефективний при рішенні завдання моделювання складних ДС.

РМП реального часу (РМП РЧ) є темпоральним підкласом РМП, орієнтованим на моделювання й аналіз систем реального часу. На відміну від РМП, у РМП РЧ використовується інша модель часу й пріоритети переходів, на

які накладені деякі структурні обмеження. Ці особливості РМП РЧ дозволяють здійснювати пряме моделювання елементів, типових для паралельної обробки (паралельного програмування): пріоритети завдань, таймаути і т.д. Визначення РМП РЧ базується на визначенні РМП із наступними відмінностями:

- кожному переходу ставиться у відповідність пріоритет, що дозволяє прямо моделювати детермінований вибір;

- кожна дуга має два параметри: вагове й часове вираження. Вагове вираження аналогічне вираженням дуг класичних РМП. При кожному обчисленні часового вираження дуги на виході повинне бути невід'ємне дійсне число;

- кожному місцю ставиться у відповідність часова мітка. Додатне значення часової мітки описує, як довго фішка у відповіднім місці буде недоступною для переходу. Фішка є доступною для переходу, якщо часова мітка має недодатне значення, наприклад, рівність мітки мінус трьом означає, що «вік» фішки три часові одиниці. Даний підхід дозволяє визначити «вік» фішки, необхідний для участі фішки в переході.

Можна вважати, використовуючи термінологію ВА, що кожному місцю РМП РЧ ставиться у відповідність локальний годинник, який ішов синхронно один з одним і с реальним часом, причому, якщо значення реального часу постійно збільшується, то значення локальних годин (часові мітки) постійно зменшуються.

Нехай задані: \dot{L} – мова типізованих виражень, які побудовані зі змінних і констант із використанням тільки операції додавання комплектів; U – кінцева модель цієї мови, елементами якої є індивідуальні (помітні) фішки.

Тип визначається як кінцева підмножина U , типом суми двох комплектів є об'єднання їх типів.

Можна визначити РМП РЧ як набір:

$$RTCPN \equiv \langle \Sigma, P, T, \dot{F}, \dot{\xi}, \dot{\gamma}, \pi, \varepsilon_{\Sigma}, \varepsilon_T, m_0 \rangle,$$

$\varepsilon_{\Sigma}: \dot{F} \rightarrow \dot{L}$ – функція вагових виражень дуг, що ставить у відповідність кожній дузі вираження мови \dot{L} , такі, що тип вагового виразу дуги має збігатися з типом того місця, з яким зв'язана ця дуга:

$$(\forall p \in P)(\forall t \in T)(\langle t, p \rangle \in \dot{F} \supset \xi(\varepsilon_\Sigma(\langle t, p \rangle)) \subseteq \dot{\xi}(p)) \wedge (\langle p, t \rangle \in F \supset (\varepsilon_\Sigma(\langle p, t \rangle) \subseteq \dot{\xi}(p)));$$

$\varepsilon_T: \dot{F} \rightarrow f_0$ – функція часових виражень дуг, що ставить у відповідність кожній дузі деяке невід’ємне раціональне число;

Модель на основі РМП РЧ (рис. 3.2) дозволяє розглянути систему як А-Систему, що складається з компонентів, яким відповідають шість місць мережі: *ContrSyst* – елемент, що контролює систему; *Console* – консоль для відображення сигналів; *Brake* – механізм реагування; *Driver* – особа, що приймає рішення; таймери *Timer1* і *Timer2*. Діаграми прикладів реалізацій процесу в *ContrSyst* наведені на рис. 3.2.

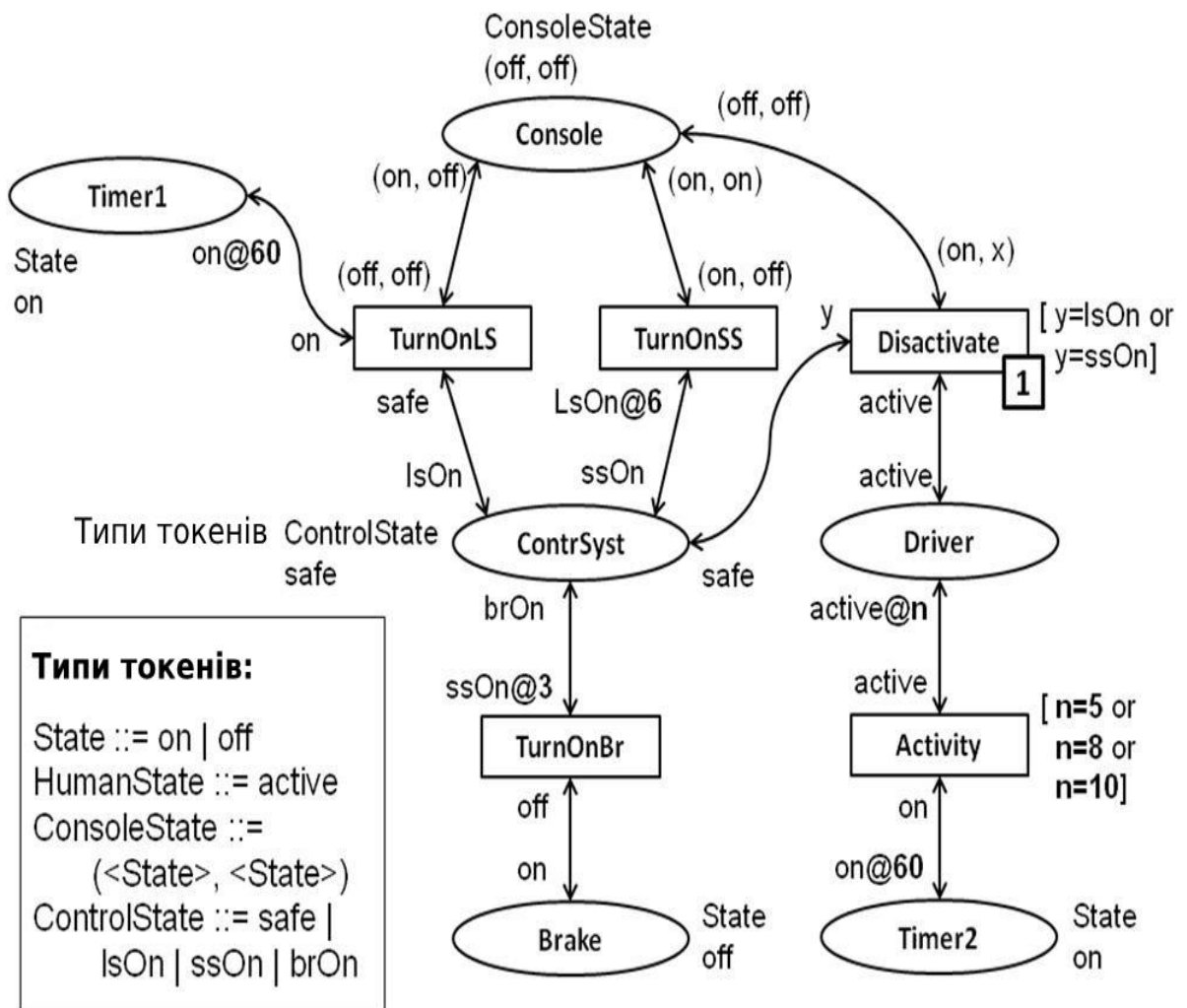


Рисунок 3.2 – Модель системи на основі РМП РЧ

Зміна станів компонентів моделюється за допомогою п'яти переходів: *TurnOnLs* – включення сигналу; *TurnOnSs* – включення сигналу 2; *TurnOnBrake* – запуск механізму реагування; *Disactivate* – дезактивація сигналів; *Activity* – моделювання дій особи.

Задане початкове маркування, початкові значення часових міток дорівнюють нулю й опущені. Перехід *Disactivate* має пріоритет 1, інші переходи – 0 (немає на схемі). Вагові й часові вираження дуг розділені символом @. Якщо часове вираження дорівнює 0, то воно опущене. Кожна дуга із двома стрілками замінює для наочності пари дуг.

Правила оперування темпоральною інформацією в РМП РЧ дозволяють моделювати на їхній основі процеси функціонування складних ДС і оперувати кількісними темпоральними залежностями. Однак очевидно, що можливості моделювання зростуть при наявності засобів, що дозволяють виражати як кількісні, так і якісні темпоральні залежності.

3.3 Модифікація РМП РЧ із підтримкою якісних темпоральних залежностей

Для розширення можливостей і більш ефективного використання РМП РЧ як основи для моделювання процесів у складних ДС і включення відповідних моделей до складу ДІС типу ІСППР РЧ необхідно розв'язати проблему представлення й оперування якісними темпоральними залежностями. Пропонується розширити апарат РМП РЧ [17-19], додавши можливість використовувати асоційовані з місцями мережі темпоральні інтервали й засоби висновку на основі ТЛА.

Існує дві причини зміни маркування РМП РЧ:

- спрацьовування переходу t ;
- перебіг часу – поступове зменшення значення кожної часової мітки на фіксовану величину, поки не з'явиться перехід, який може спрацювати.

Для переходу t позначимо V_t множину змінних мови \dot{L} , які зустрічаються у вираженнях функцій ε_{Σ} і ε_T вхідних і вихідних дуг переходу й у вираженні захисної функції $\dot{\gamma}(t)$.

Перехід t є дозволеним (допустимий) (може спрацювати) у стані в m підстановці β тоді и тільки тоді, коли одночасно виконуються наступні умови:

$$Q(t, \beta, m) \cong (\forall p \in P)(\langle p, t \rangle \in \dot{F} \supset (\varepsilon_{\Sigma}(\langle p, t \rangle)_{\beta} \subseteq \mu(p) \wedge \varepsilon_{\Gamma}(\langle p, t \rangle)_{\beta} \leq -\dot{\tau}(p)) \wedge \langle t, p \rangle \in \dot{F} \supset \dot{\tau}(p) \leq 0 \wedge (\dot{\gamma}(t)_{\beta} = true).$$

В одному стані перехід може спрацювати по-різному, залежно від підстановки. Число можливих підстановок звичайне, оскільки звичайні як множина змінних V_t , так і множина фішок U . Для фіксованого стану мережі m позначимо множину усіх підстановок переходу t , в яких він допустимий, як-то:

$$Bind_t^m \cong \{\beta \mid \beta \in Bind_t, Q(t, \beta, m) = true\}.$$

Задамо множину переходів, допустимих хоча б в одній підстановці:

$$EnT_m \cong \{t \mid t \in T, Bind_t^m \neq \emptyset\}.$$

Слід називати таку підмножину множиною неконфліктних переходів. Переходи однієї такої множини можуть спрацювати паралельно, оскільки не впливають на умови спрацювання інших переходів множини.

Щоб побудувати всі множини неконфліктних переходів, необхідно розглянути комбінації елементів множини припустимих переходів EnT_m . Складність такого комбінаторного завдання порівнянна з кількістю комбінацій дорівнює $2^{|EnT_m|}$ (без врахування пошуку загальних підстановок для пар переходів).

При використанні функції пріоритетів переходів π необхідність повного перебору комбінацій припустимих переходів відсутня. Загальний принцип полягає в побудові множини з єдиного елемента – припустимого переходу t з множини EnT_m з максимальним пріоритетом $\pi(t)$ – і послідовним розширенням цієї множини іншими припустимими переходами з EnT_m за зменшенням пріоритету. Алгоритм пошуку множини неконфліктних переходів наведено в главі 3, складність такого алгоритму поліноміальна. При цьому основна

відповідальність лягає на розробника моделі, який повинен коректно вказати пріоритети переходів.

Таким чином, кожна часова мітка є таймером, що здійснює зворотний відлік. Даний підхід зручний при моделюванні циклічних процесів, що повторюються, оскільки допускає періодичне обнуління таймерів (при спрацьовуванні переходів).

Слід зазначити, що безумовний пріоритет при зміні станів мережі має подія спрацьовування переходу. Перебіг часу дозволяє тільки чекати моменту, коли може спрацювати черговий перехід. Тому після кожної зміни маркування здійснюються наступні дії.

Знайти всі допустимі переходи: побудувати множини $Bind_t^m$ і EnT_m за правилами. Якщо EnT_m не є не пустою множиною, то сформувані множини неконфліктних переходів і виконати спрацьовування кожного, інакше зменшити для кожного місця значення часової мітки.

Деяка довільна послідовність маркувань при цьому виглядає так:

$$m_i \xrightarrow{\langle t_i, \beta_i \rangle} m_{i+1} \xrightarrow{\langle t_{i+1}, \beta_{i+1} \rangle} m_{i+2} \xrightarrow{\tau'} m_{i+3} \xrightarrow{\tau'} m_{i+4} \xrightarrow{\langle t_{i+4}, \beta_{i+4} \rangle} m_{i+5}.$$

Уведено функцію $\ddot{t}: \dot{M} \times \dot{M} \rightarrow m_0$, що визначає час, що прошедше між появою маркування m і її зміною на m' . При цьому:

$$\ddot{t}(m, m') \equiv 0;$$

$$\text{якщо } m \xrightarrow{\langle t, \beta \rangle} m', \text{ то } \ddot{t}(m, m') = 0;$$

$$\text{якщо } m \xrightarrow{\tau'} m', \text{ то } \ddot{t}(m, m') = \tau;$$

$$\text{якщо } m \xrightarrow{\tau} m'' \xrightarrow{\langle t, \beta \rangle} m', \text{ то } \ddot{t}(m, m') = \ddot{t}(m, m'') + \ddot{t}(m'', m') = \tau;$$

$$\text{якщо } m \xrightarrow{\tau_1} m'' \xrightarrow{\tau_2} m', \text{ то } \ddot{t}(m, m') = \ddot{t}(m, m'') + \ddot{t}(m'', m') = \tau_1 + \tau_2.$$

При аналізі будемо використовувати наступну форму запису зміни станів мережі:

$$m \xrightarrow{\langle t, \beta, \ddot{t}(m, m') \rangle} m' .$$

Цей запис читається як «система перебувала в стані m $\ddot{t}(m, m')$ одиниць часу, а потім стан m змінився на m' у результаті спрацьовування переходу t у підстановці β ». Послідовність (2.10) у такій формі виглядатиме так:

$$m_i \xrightarrow{\langle t_i, \beta_i, 0 \rangle} m_{i+1} \xrightarrow{\langle t_{i+1}, \beta_{i+1}, 0 \rangle} m_{i+2} \xrightarrow{\langle t_{i+4}, \beta_{i+4}, 2\tau' \rangle} m_{i+5}.$$

можна визначити наступну захисну функцію: Brake(on){b}ContrSyst(safe). Така функція повертає значення true, якщо фішка on перебувала в місці Brake до того, як фішка safe перебувала в місці ContrSyst, і значення false, якщо ні.

Мережу РМП РЧ, що підтримує роботу з подібними конструкціями, зватимемо РМП РЧ із підтримкою темпоральної логіки Аллена (РМП РЧ ТЛА).

Модель системи екстреного гальмування потягу, побудована за допомогою РМП РЧ ТЛА, представлена на рис. 3.4.

Перевагою застосування РМП РЧ ТЛА в цьому випадку є можливість задавати не конкретний час реакції, як у випадку РМП РЧ, а інтервали, на яких ОПР може дезактивувати систему й кожний з яких визначає подальшу поведінку моделі. Таким чином, включення в модель засобів інтервальної темпоральної логіки дозволило адекватно відбити невизначеність, властиву завданню, що є одним з основних принципів при проектуванні перспективних ІСППР РЧ і ДІС у цілому.

Методи аналізу властивостей МП засновані на використанні графів досяжних (покриваючих) маркувань, вирішенні рівняння станів мережі й обчисленні лінійних інваріантів позицій і переходів. Застосовуються також допоміжні методи редукції, що дозволяють зменшити розмір МП зі збереженням її властивостей, і декомпозиції, що розділяють вихідну МП на підмережі.

Відомі наступні основні групи методів аналізу МП: засновані на побудові графів зміни станів; матричні методи, що використовують рівняння мережі й інваріанти; методи редукції. При роботі з порівняно складними модифікаціями МП (МП високого рівня) – такими як РМП, РМП РЧ – останні дві групи методів використовуються рідко через високу (у порівнянні із класичними МП) складність формальних побудов. Частіше в якості основного інструмента аналізу розглядаються графи досяжності й покриття. Такий підхід пропонується застосувати й для РМП РЧ ТЛА.

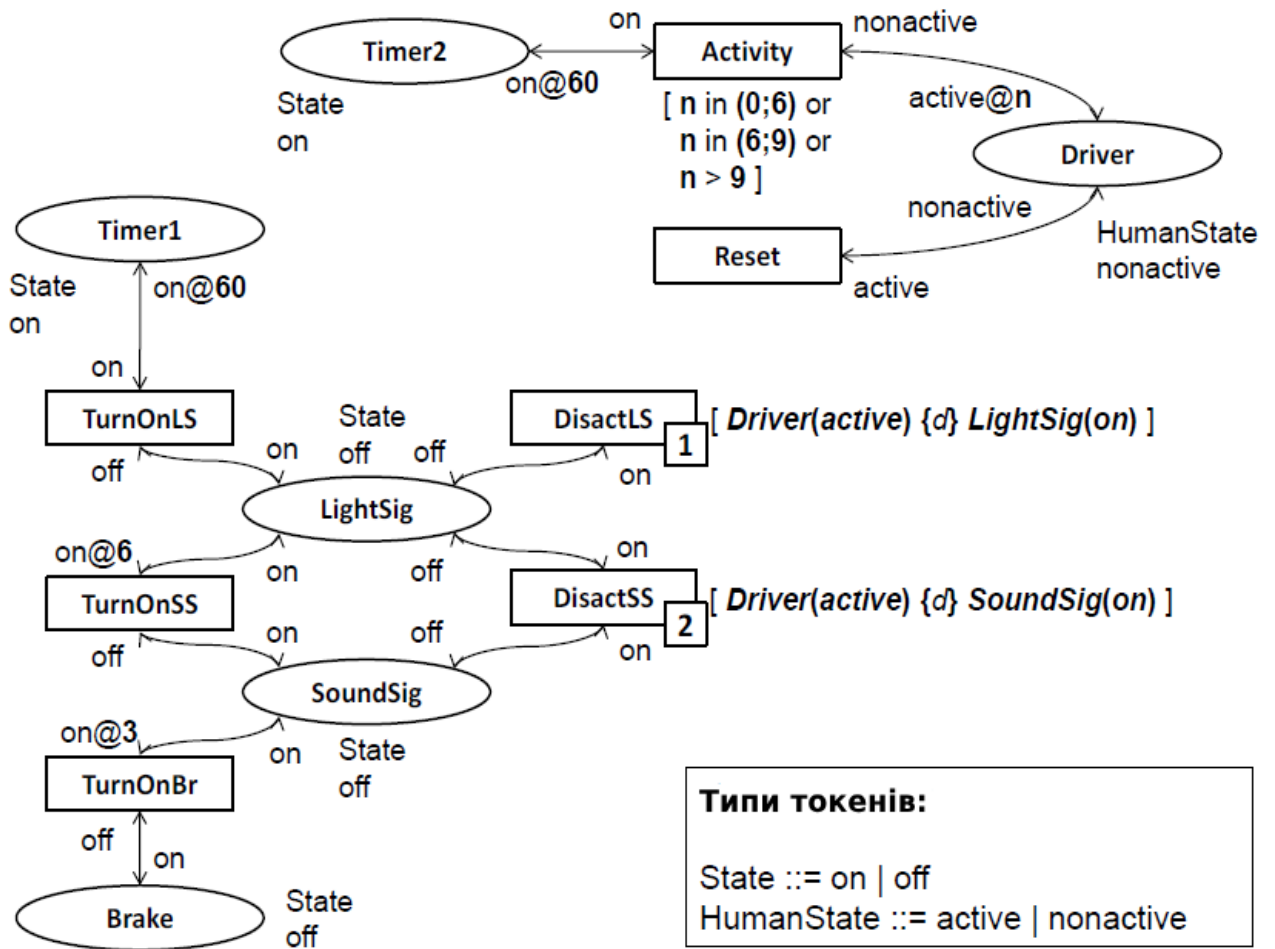


Рисунок 3.4 – Модель системи на основі РМП РЧ ТЛА

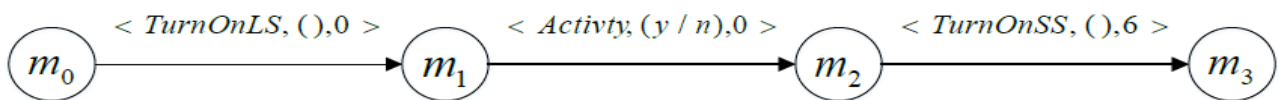


Рисунок 3.5 – Фрагмент графа досяжності

Формально ГД для мережі можна представити як набір:

$$RG \equiv \langle \dot{P}, \ddot{R}(m_0), \ddot{\xi}, \ddot{\epsilon} \rangle,$$

де \dot{P} – непорожня множина вузлів графа, $|\dot{P}| = |\ddot{R}(m_0)|$;

$\ddot{R}(m_0)$ – непорожня множина досяжних маркувань мережі;

$\ddot{\xi}: \dot{P} \leftrightarrow \ddot{R}(m_0)$ – взаємо-однозначне відображення між множиною вузлів графа й множиною досяжних маркувань мережі $(p_1, p_2) = \langle t, \beta, \bar{t} \rangle$.

Аналіз властивостей мережі може здійснюватися за допомогою маркування вузлів і міток дуг ГД. Кожна мітка дуги являє собою трійку, що полягає з переходу, його підстановки й значення часового проміжку перед його спрацьовуванням. Останній параметр, таким чином, дозволяє визначити час, витрачений на перехід з одного стану в інший. Використовуючи стандартні алгоритми пошуку найкоротшого або найбільшого шляху між двома вузлами ГД, можна знайти мінімальний та максимальний час переходу з одного стану в інший.

Якщо продовжити будувати ГД для аналізованої мережі, можна переконатися, що через безперервне зменшення значень часових міток він виявляється нескінченним. Подібна ситуація виникає при аналізі практично будь-якої РМП РЧ, у тому числі РМП РЧ ТЛА, причому ступінь зв'язаності мережі не впливає на кінцівку ГД.

3.4 Верифікація моделей процесів, створених на основі мереж Петрі

Проблема забезпечення коректності програмних і апаратних компонентів складних ДС здобуває сьогодні першорядне значення. Надійність і передбачуваність поведінки таких систем найчастіше є більш важливими властивостями, ніж продуктивність, модифіковність і т.п. Добре відомо, що навіть у тих випадках, коли функціонування кожного з паралельних взаємодіючих компонентів системи абсолютно зрозуміло, людині важко зрозуміти роботу всієї паралельної системи в цілому, процеси в якій взаємозалежні. Системи, які працюють коректно «майже завжди», роками можуть зберігати помилки, що проявляються у виняткових ситуаціях. Їхніми безпосередніми причинами є й некоректні специфікації, й неправильне розуміння специфікації розробниками, неузгодженість паралельних галузей процесів і багато чого іншого. Найбільш очевидним і широко розповсюдженим методом перевірки правильності програмних систем є тестування – перевірка роботи побудованої системи в різних

ситуаціях, при різних вхідних даних. Однак у випадку з паралельними системами зазвичай складно (або неможливо) заздалегідь визначити всі можливі траєкторії функціонування. Тому в якості основного методу підвищення якості розробки застосовується верифікація – формальна перевірка того, що система (модель) задовольняє сформульованим заздалегідь вимогам [29]. Методи верифікації різняться залежно від того, який формальний апарат лежить в основі опису (моделі), що перевіряється системою.

Для верифікації ДС властивості їх поведінки повинні бути виражені формально-логічними твердженнями, істинність яких залежить від часу. Звичайна логіка висловлень є погано придатною для формулювання тверджень про поведінку складних ДС при зміні їх станів у часі. Формалізація навіть найпростішого прикладу «Будь-який посланий запит коли-небудь пізніше буде обслужений» за допомогою, наприклад, логіки предикатів першого порядку призводить

Тому при верифікації темпоральних конструкцій необхідно використовувати засоби темпоральних логік. Якщо немає необхідності докладно описувати закономірності поведінки системи й взаємодія її об'єктів, доцільніше застосовувати не досить складні інтервальні логіки, а темпоральні розширення звичайної логіки висловлення. Традиційно при верифікації використовуються темпоральні логіки лінійного часу (Linear Temporal Logic, LTL) і розгалуженого часу (Computational Tree Logic, CTL).

Верифікація моделі на основі РМП РЧ ТЛА за допомогою методу МС є природним розширенням початкового аналізу мереж за допомогою графів станів.

Відомо, що як тестування, так і верифікація окремо не можуть гарантувати достатнього рівня коректності розроблених систем. Існує велика кількість прикладів, коли в ретельно перевірених й протестованих реалізаціях згодом за допомогою верифікації виявлялися тонкі помилки.

З другого боку, не можна сподіватися лише на верифікацію. Часто причина помилок криється в тому, що й при розробці, й при доведенні алгоритмів неявно висуваються неправильні припущення (специфікації) про характер роботи й,

внаслідок цього, неадекватні формальні моделі. Таким чином, як тестування, так і верифікація мають свої переваги й недоліки, тому ці підходи є взаємодоповнюючими і повинні застосовуватися спільно.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ БАЗОВИХ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ

4.1 Розробка прототипу інструментарію

Комп'ютерне моделювання складних ДС актуальне як для теоретичних досліджень, так і для практичного застосування. Сучасні інструментальні засоби конструювання ІС, зокрема, інструментальний комплекс конструювання систем реального часу G2 (Gensym Corp, США) дозволяють реалізовувати подібні проекти.

В середовищі G2 розроблено прототип інструментарію для моделювання процесів у складних ДС на основі РМП РЧ ТЛА, а також опис запропонованих до роботи інструментальних засобів, розроблених у середовищі Microsoft Visual Studio мовою C# і розширювальних можливостей базових засобів G2 у плані моделювання складних ДС у складі ДІС типу ІСППР РЧ, що також дозволяють автономне використання. Застосування прототипу продемонструвало можливість комп'ютерного моделювання процесів у складних ДС на основі формалізму РМП РЧ ТЛА, однак висока вартість комплексу G2 (як і інших подібних систем) стала приводом для розробки окремого додатка.

G2 – це об'єктно-орієнтоване інтегроване середовище для розробки й супроводу додатків – інтелектуальних систем реального часу, що використовують БЗ [15, 45]. На відміну від систем, орієнтованих на якусь одну методологію або на конкретну предметну область, G2 інтегрує в собі множину взаємодоповнюючих методів штучного інтелекту, що спрощує й прискорює процес розробки додатків і дозволяє робити їх досить універсальними. Програмні продукти, розроблені за допомогою G2, є незалежними від того апаратного забезпечення, на якому вони працюють, тобто вони є переносними. Можливість простого маніпулювання графічним представленням об'єктів в G2 і складання схем, що є відображенням технологічних ланцюжків або алгоритмів обробки даних, забезпечує базові засоби для побудови проблемно-орієнтованих мов візуального програмування. У цей час існують версії G2 для всіх типів робочих станцій і персональних ЕОМ, а також

забезпечується можливість інтеграції G2 з більшим числом програмних і апаратних засобів різних фірм.

В системі G2 був розроблений прототип інструментарію для моделювання процесів у складних ДС на основі РМП РЧ ТЛА. При цьому був зроблений ряд допущень:

- створення тільки безпечних мереж (кількість фішок у кожному місці не перевищує 1), причому ваговими вираженнями дуг теж є тільки одиничні фішки;
- спрощений алгоритм розв'язання конфлікту переходів (не розглядаються множини припустимих неконфліктних переходів);
- у моделі відсутні змінні, що призводить до того, що будь-яка підстановка переходу b є суть тривіальної: $b=()$;
- захисні функції на основі ТЛА необхідно реалізовувати в окремих процедурах;
- часовий крок зафіксований і дорівнює 1 секунді.

Середовище G2 підтримує реальний час. Тому додаток G2 завжди перебуває в одному із двох станів – статичному або динамічному. У статичному стані в середовищі не відбувається ніяких автоматичних змін, залишається тільки можливість ручних налаштувань і змін значень елементів користувачем. У динамічному режимі (зі зміною часу) виконується код програми, що описує поведінку системи, автоматично відбувається зміна параметрів середовища, середовище реагує на ці зміни й на зміни, зроблені користувачем. Автоматичне планування завдань підвищує ефективність і оптимізує використання обчислювальних ресурсів. Крім того, система G2 має вбудовані функції, що дозволяють працювати з поточним часом середовища й адекватно реагувати на його зміни.

Одним з основних компонентів G2 є машина висновку, що виконує міркування на основі знань, що втримуються в БЗ, і даних, що надходять від підсистеми імітаційного моделювання або від зовнішніх джерел (контрольно-вимірювальної апаратури, СКБД і т.п.). Правила продукційного типу активуються машиною висновку, при цьому перевіряється істинність умови, що перебуває в

посилці правила. Якщо умова дійсна, то машина висновку виконує дії, зазначені у висновку правила. У число виконуваних дій входять: присвоєння значення простому атрибуту, параметру або змінній; посилка керуючої інформації зовнішньому об'єкту; запуск процедури; створення екземпляра об'єкта й інші.

Особливістю машини висновку G2 є багатий набір способів активації правил, що включає крім активації в ході прямого або зворотного висновку такі механізми, як періодично виконувані правила й правила-реакції. Правила-реакції активуються в першу чергу й мають найбільший пріоритет, що гарантує своєчасну реакцію системи на зміни в навколишньому середовищі. Правила цього типу не використовуються за замовчуванням ні в прямому, ні у зворотному висновку, вони є метаправилами й реагують на події (переміщення об'єкта, установлення/усунення відносини, одержання/неотримання значення). Ці особливості дозволили розробити в данім середовищі прототип інструментарію для моделювання складних ДС на основі РМП РЧ ТЛА й для включення цього інструментарію до складу ДІС типу ІСППР РЧ.

Робочою областю в G2 називається область для візуальної організації елементів створюваного додатка. В одного додатка, як правило, існує кілька робочих областей, між якими встановлена стругаюча ієрархія залежно від рівня деталізації додатка. Розрізняють головну робочу область – найбільш високий рівень організації керування, являє собою кореневий контейнер для інших елементів додатка. Також виділяють підобласті – вони відповідають одному елементу робочої області попереднього рівня. З елементами підобласті можна працювати, перейшовши до неї за посиланням з елемента, що визначає підобласть.

При розробці прототипу інструментарію в середовищі G2 минулого створені наступні робочі області: ГОЛОВНЕ_ВІКНО, МОДЕЛЮВАННЯ, ПРОЦЕДУРИ, ПРАВИЛА, КЛАСИ, області методів класів.

Ієрархія робочих областей представлена на рис. 4.1. Головною робочою областю є ГОЛОВНЕ_ВІКНО, навігація між областями здійснюється за допомогою кнопок навігації, описаних у підключеному модулі *uilroot.kb*.

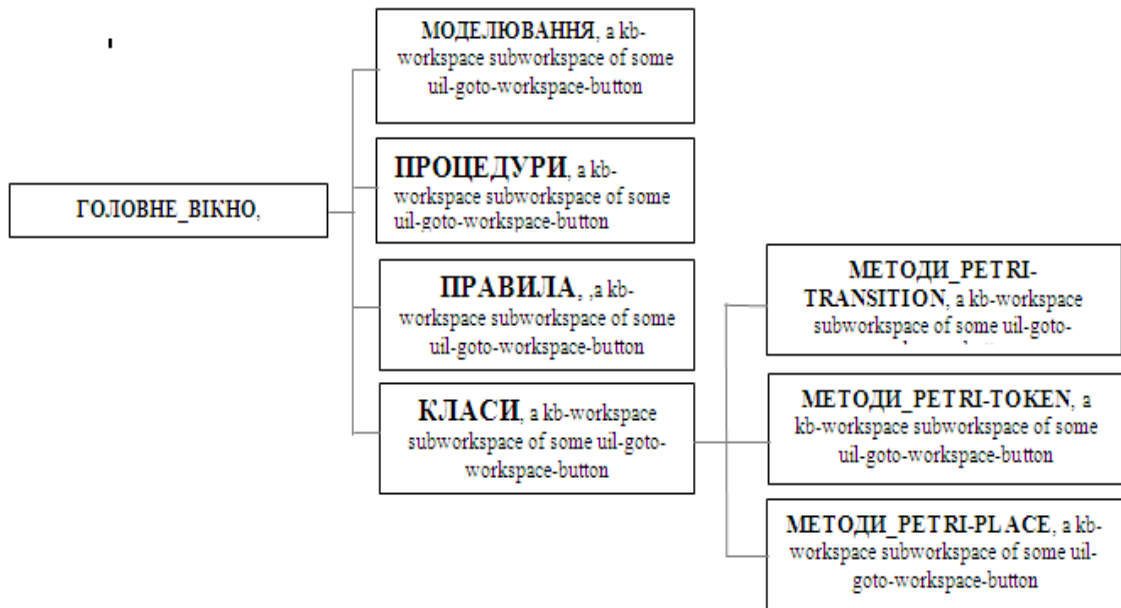


Рисунок 4.1 – Ієрархія робочих областей додатка

На робочій області ГОЛОВНЕ_ВІКНО розташовані кнопки перехід до підобластей і кнопки перемикання режимів роботи системи G2: доступні два режими – адміністратор і розроблювач (рис. 4.2а). Класи, параметри й відносини, створені для моделювання, можна бачити на підобласті КЛАСИ (рис. 4.2в). Там же перебувають кнопки перехід до підобластей методів розроблених класів. Класи елементів мережі розташовані в лівому стовпці, перехід до їхніх методів – у центральному. У лівій частині розміщені класи з'єднань і відносини між елементами. У нижній частині розташовані параметри – об'єкти, які можуть зберігати змінні дані певного типу – integer, float, text, truth-value або symbol. Вони, як і класи, є глобальними об'єктами в середовищі G2, і тому розташовані на цій підобласті. Процедури й функція додатка розташовані на робочій області ПРОЦЕДУРИ (рис. 4.2 г). Робочі області КЛАСИ, ПРОЦЕДУРИ й ПРАВИЛА необхідні для опису апарата роботи системи й при створенні моделі не потрібні. Усі дії, необхідні для створення моделі, можна виконати в робочій області МОДЕЛЮВАННЯ. Як видно з рис. 4.2б, у лівій частині області розташовані елементи керування створюваної системи: кнопка «Пуск-Стоп», що запускає й зупиняє моделювання; поточний час моделювання й кнопка «Обнулити час»; кнопка «Обнулити мітки», що дозволяє обнулити часові мітки всіх місць моделі;

блок створення об'єктів. Права частина області МОДЕЛЮВАННЯ повністю призначена для РМП РЧ ТЛА.

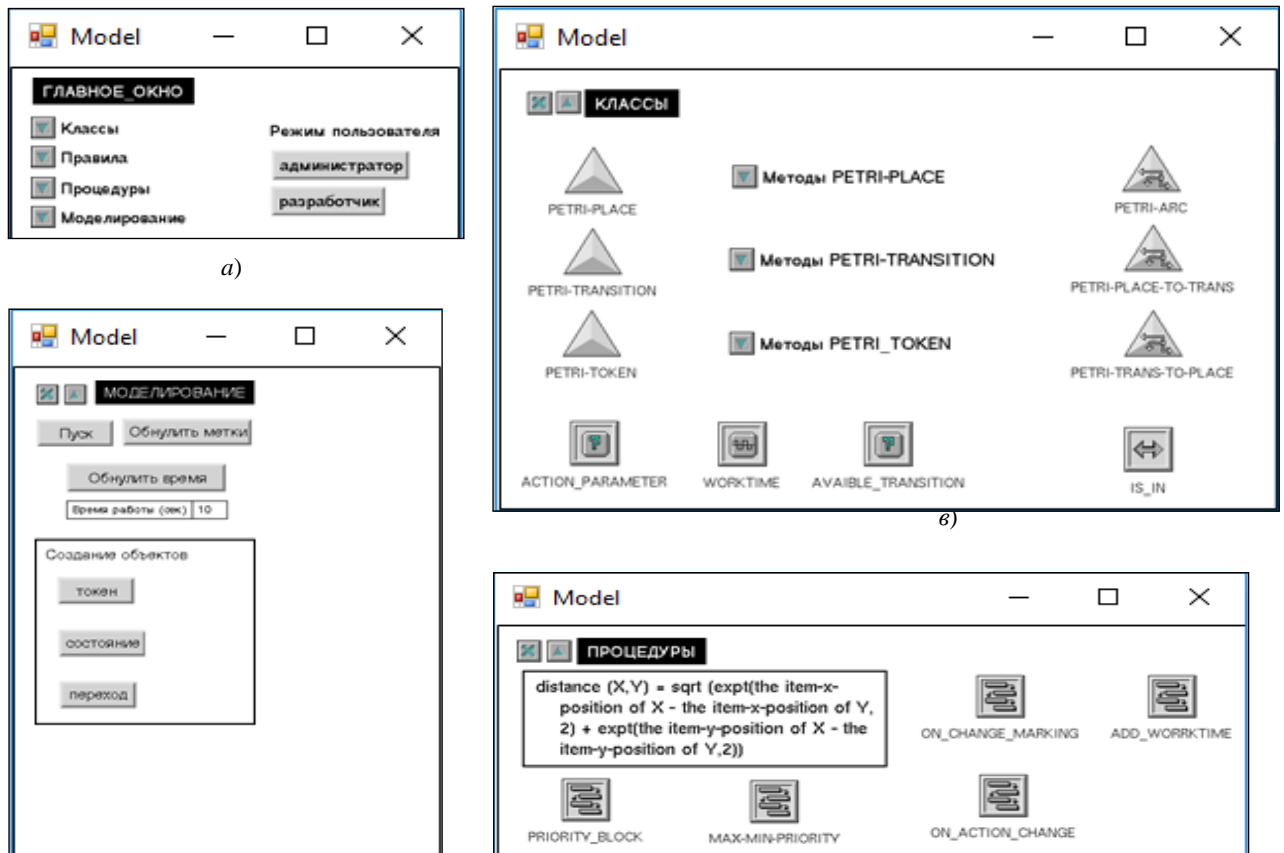


Рисунок 4.2 – Робочі області: а) ГОЛОВНЕ_ВІКНО, б) МОДЕЛЮВАННЯ, в) КЛАСИ, г) ПРОЦЕДУРИ

На робочій області ПРАВИЛА, частина якої зображена на рис. 4.3, розміщені правила, які забезпечують коректну роботу системи (прототипу).

Для моделювання ДС на основі РМП РЧ ТЛА в середовищі G2 минулого розроблені класи, що описують елементи, які відповідають об'єктам апарата МП. Це дозволяє коректно й досить просто моделювати певні процеси за допомогою РМП РЧ ТЛА.

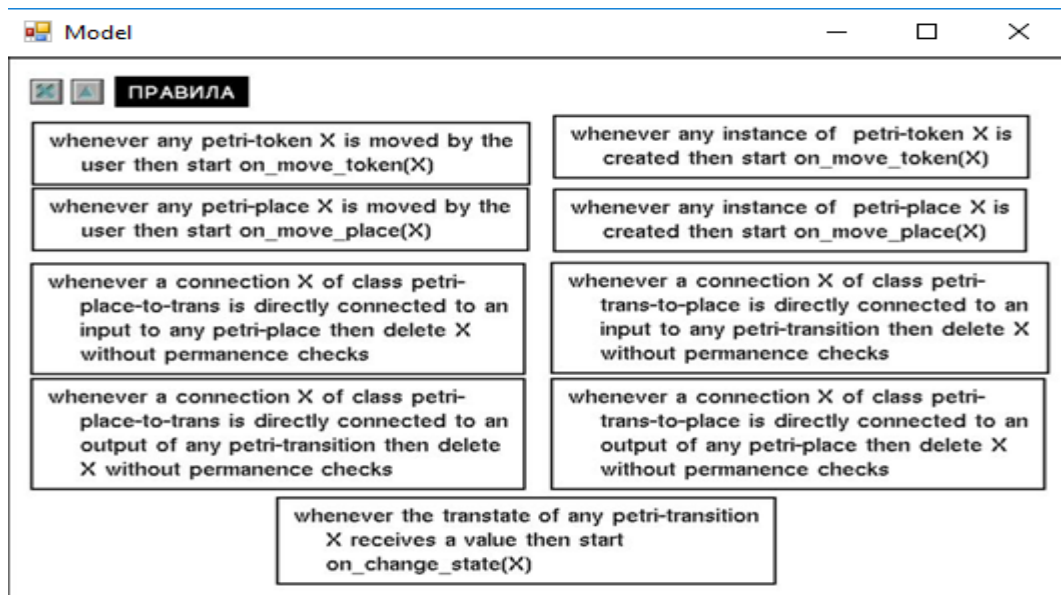


Рисунок 4.3 – Робоча область ПРАВИЛА (фрагмент)

Клас PETRI-PLACE описує елементи множини місць P мережі. Його визначають наступні атрибути: color is a symbol, initially is state; time_stamp is an integer, initially is 0; mark_stamp is a symbol, initially is no_token; already_seen is a truth-value, initially is false.

Атрибут color визначає тип (колір) кожного місця, атрибути time_stamp і mark_stamp визначають часову мітку й маркування відповідно, already_seen використовується під час аналізу доступності переходів при зміні маркування мережі, алгоритм якого описаний нижче.

Клас PETRI-TRANSITION описує елементи множини переходів T. Він задається наступними атрибутами: transtate is a symbol, has values active or nonactive, initially is nonactive; in_avaible is a truth-value, initially is false; out_avaible is a truth-value, initially is false; priority_conflict is a truth-value, initially is false; priority is an integer, initially is 0; guard is a truth-value, initially is true.

Атрибут transtate визначає поточний стан переходу в мережі й задається типом, що перелічується: може мати значення active – активний і nonactive – не активний. При зміні значення атрибута змінюється зображення об'єкта класу, цього дозволяє досягти правило, що запускає метод класу on_change_state:

whenever the transtate of any petri-transition X receives a value then start on_change_state(X)

Атрибут `priority` задає пріоритет переходу й ураховується при розв'язанні конфлікту ресурсів. Атрибут `guard` відповідає значенню захисної функції G , що обчислюється щораз окремо. Інші атрибути – `in_avaible`, `out_avaible` і `priority_conflict` – використовуються при визначенні припустимих переходів і розв'язанні конфліктів між ними.

Ще один клас, `PETRI-TOKEN`, описує фішки, які в сукупності з місцями, у яких вони розташовані, задають $\mu(p)$. Оскільки в РМП РЧ ТЛА розглядаються кольорові фішки, усі фішки розробленої моделі різняться типами `color`, значеннями `token_color` і місцями знаходження в мережі: `color is a symbol, initially is state; token_color is a symbol, has values on or off, initially is off.`

Для зв'язку між фішками й місцями `PETRI-PLACE` було створене відношення `IS_IN` типу `many-to-one`, описуване в такий спосіб:

```
petri-token may be is_in at most one
petri-place; more than one petri-token may
be is_in a petri-place.
```

4.2 Організація функціонування моделі

Функціонування ІМ засноване на роботі правила запуску й процедур перебігу часу, зміни маркування й розв'язання конфліктів. При запуску ІМ кнопкою «Пуск» на робочій області `МОДЕЛЮВАННЯ` відбувається виклик процедури `add_worktime`, яка відповідає за роботу моделювання й перебіг часу.

Для зміни маркування використовується процедура `on_change_marking`, мета роботи якої – аналіз ситуації, що склалася знову, визначення припустимих переходів і моделювання їх спрацьовування. У конфліктних ситуаціях (коли в маркуванні припустимі, що трохи ділять фішки переходів) необхідно зробити вибір, який перехід повинен спрацювати. Розроблена система моделювання підтримує правило вибору по пріоритету π : $T \rightarrow \tau_0$: в першу чергу спрацьовують переходи з максимальним значенням атрибута `priority`.

Процедури аналізу й розв'язання конфліктів досить складні. Однак загальна схема організації роботи ІМ порівняно проста.

При зміні маркування здійснюється обчислення інтервальних захисних функцій переходів, які описуються в окремому блоці процедур. Результати оцінки допустимості записуються в атрибути `guard` відповідних переходів, після чого робота ведеться тільки з тими з них, у яких даний атрибут має значення `true`.

Потім паралельно проводиться оцінка допустимості кожного переходу: перевіряється, чи задовольняє поточна ситуація умовам (2.3), відповідно до чого атрибутам переходу `in_avaible` і `out_avaible` привласнюються значення `true`, якщо умови виконуються, і значення `false` – якщо не виконується.

Після цього відбувається виклик процедури `priority_block`, завдання якої – визначити, які з поточних припустимих переходів є конфліктними. Атрибуту `priority_conflict` усіх переходів надається значення `false`. Потім для кожного місця, яке пов'язане із припустимим у поточнім маркуванні переходом, будується послідовність усіх припустимих переходів, яким необхідні фішки, що перебувають у цьому місці. Ця послідовність упорядковується за убутанням пріоритетів переходів. У процесі проходження по впорядкованій послідовності всі елементи, крім першого, одержують значення атрибута `priority_conflict true`. Якщо максимальний пріоритет послідовності дозволяє зробити перехід, значення його атрибута `priority_conflict` зберігається. Якщо дана ситуація не відповідає вимогам, усім конфліктним переходам виставляється значення атрибута `priority_conflict true`, що приводить до блокування всіх припустимих переходів на даному етапі. Тому при розробці ІМ особливо важливо грамотно розставляти пріоритети переходів: некоректне розміщення й відсутність розуміння загальної структури мережі можуть привести до множинних блокувань, відстежити які досить проблематично.

По завершенні процедури `priority_block` паралельно для всіх переходів, атрибути допустимості яких `in_avaibe` і `out_avaible` мають значення `true`, а `priority_conflict` – `false`, виконуються наступні дії:

- переклад переходу в активний стан;

- здійснення переходу.

Спрацьовування переходу здійснюється згідно із правилом (2.8). Зміна станів мережі відбувається шляхом видалення необхідних фішок із вхідних місць переходу, приміщенням фішок у вихідні місця й коректній роботі з часовими мітками. Оскільки допустимість переходу перевіряється при кожній зміні станів, аномалій такий механізм не викликає.

Процедура `on_change_marking` викликається в циклі без зміни часових міток, поки відбувається Спрацьовування хоча б одного переходу, за що відповідає логічний системний параметр `AVAILABLE_TRANSITION`. Після цього в циклі здійснюється одночасне зменшення значень часових міток у всіх місцях мережі.

4.3 Розробка інструментарію в середовищі Microsoft Visual Studio

Розробка прототипу в середовищі G2 і наступне створення з його допомогою моделей процесів дозволили підтвердити раціональність застосування модифікованих МП – РМП РЧ ТЛА – при створенні ІМ складних ДС. Однак використання середовища G2, незважаючи на безсумнівну зручність розробки, суттєво звужує область застосовності розроблювального програмного продукту. Тому було прийняте рішення розробляти повноцінну версію інструментарію на мовах високого рівня в середовищі Microsoft Visual Studio (MVS).

Завдання розробки програми-інструментарію можна розбити на дві підзадачі:

- розробка функціональної логіки роботи додатка, так званої «моделі» наукомістку частину, що містить усю, завдання;
- розробка графічного інтерфейсу користувача (graphical user interface, GUI) – «представлення», за допомогою якого користувач (розроблювач моделей процесів ДС) створює й надбудовує «модель».

Для розробки GUI було прийняте рішення використовувати найбільш сучасну графічну .NET систему Windows Presentation Foundation (WPF) для побудови клієнтських додатків Windows з візуально привабливими можливостями взаємодії з користувачем, що використовує мову XAML (eXtensible Application Markup Language). В основі WPF лежить векторна система візуалізації, що не залежить від роздільної здатності пристрою виводу й створена з урахуванням можливостей сучасного графічного пристрою. Графічною технологією, що лежить в основі WPF, є Directx, на відміну від Windows Forms, де використовується GDI/GDI+. Продуктивність WPF вище, чим в GDI+ за рахунок використання апаратного прискорення графіки через Directx.

Застосування WPF спричинило рішення використовувати шаблон Model-view-viewmodel, (MVVM) – «модель»-«представлення»-«модель представлення». «Модель представлення», з одного боку, є абстракцією «представлення», а з іншого – надає дані з «моделі». Загальна схема архітектури MVVM, що використовується в розроблювальному додатку.

Класичний шаблон «модель»-«представлення»-«контролер» (Model-View-Controller), при використанні якого зміни в користувацькому інтерфейсі не впливають безпосередньо на модель, а попередньо йдуть через контролер, у цьому випадку незручний, оскільки в технології WPF широко використовується концепція «зв'язування даних», що дозволяє зв'язувати дані з візуальними елементами інтерфейсу в обидва боки.



Рисунок 4.4 – Загальна схема архітектури додатка за шаблоном MVVM

Uml-Діаграма класів простору імен Rtcnptoolkit.Model наведена на мал. 3.8. Більшість елементів мережі являють собою екземпляри класів-спадкоємців абстрактного класу Idname, який містить два атрибути: ідентифікатор Id, що

представляє статистично унікальний 128-бітний ідентифікатор GUID (Globally Unique Identifier); назва елемента Name типу string. Використання ідентифікатора дозволяє однозначно визначати кожний елемент моделі. Унікальність кожного окремого GUID не гарантується, однак загальна кількість унікальних ключів настільки велика (2128), що ймовірність того, що будуть незалежно згенеровані два співпадаючі ключі, украй мала. Тому конфліктами, викликаними збігом ідентифікаторів, можна зневажити.

Класи Color, Variable і Colorset є безпосередніми спадкоємцями Idname і дозволяють організувати роботу з різними множинами типів фішок мережі. Сама множина кольорів визначається екземплярами класу Colorset, який включає колекції значень типів Col, змінних Var і констант Val. Значення типів є екземпляри класу Color, а змінні й константи задаються класом Variable, атрибут якого Color_Id зберігає ідентифікатор поточного кольору Color. Колекція всіх множин кольорів моделі Colset, відповідна до множин Σ формального визначення РМП РЧ ТЛА, зберігається в екземплярі класу Modeldeclarations, там же перебувають методи обробки.

Клас Petrinettransition дозволяє створювати моделі переходів. Атрибут Priority визначає пріоритет переходу ($\pi: T \rightarrow \mathfrak{R}_0$), а атрибут Guard задає захисну функцію переходу $\gamma: T \rightarrow \text{Bool}$. Захисна функція при цьому визначається як двовимірний «зубчастий» (jagged) списковий масив, де кожний елемент є елементарним логічним вираженням – екземпляром одного із класів Markexpr, Timenodeexpr, що є спадкоємцями абстрактного класу expr. Кожний рядок масиву Guard інтерпретується як кон'юнкція логічних виражень, а сукупність рядків – як диз'юнкція. Докладний опис класів логічних виражень наведений в наступному підпункті.

Дуги мережевої моделі визначаються екземплярами класу Petrinetarc, атрибути якого Nodefrom і Nodeto зберігають ідентифікатори вузлів, що з'єднуються дугою. Функції вагових і часових виражень дуг ε_Σ і ε_T задаються атрибутами Arcexpr і Timeexpr відповідно. При цьому Arcexpr є колекцією екземплярів класу Petrinettoken, однак у цьому випадку, на відміну від атрибутів

маркувань класу мість Petrinetplace, можливе використання не тільки ідентифікаторів кольорів Col, але й ідентифікаторів змінних Var і констант Val. Це істотне ускладнення логіки функціонування в порівнянні з логікою, яка використовувалася в прототипі інструментарію. Використання змінних у вагових функціях дуг і захисних функціях переходів приводить до необхідності організації роботи з нетривіальними підстановками, для чого використовується структура bind, що зберігає ідентифікатор змінної var_id і її значення var_id.

Функціонування ІМ здійснюється в просторі імен Rtcpntoolkit.Model цілком автономно. При зміні стану мережі для кожного місця, що бере участь в цьому процесі, створюється подія Update_Place делегата Update_Place_Delegate. Дана подія відслідковується екземпляром класу Windowviewmodel простору імен Rtcpntoolkit.Viewmodel відображенням, що управляє, в GUI.

4.4 Зберігання результатів

Завдання збереження даних моделей, розроблених за допомогою створеного інструментарію, було вирішено з використанням технології XML. Мова XML зручна для створення й обробки документів програмами й одночасно зручний для читання й створення документів людиною. Для зберігання моделей РМП РЧ ТЛА була розроблена спеціальна структура файлу. Кореневий елемент <net> містить два елементи верхнього рівня: <declaration>, де зберігаються множини кольорів, змінні й константи класу Modeldeclarations, і <topology>, що визначає топологію (структуру) моделі РМП РЧ ТЛА – місця, переходи й дуги. Елемент <declaration> містить кілька елементів <colset> кольорів, що зберігають інформацію з множин, Colorset. Кожний елемент <colset> містить у собі три групи: <colors>, <variables>, <values>, яким відповідають колекції значень типів Col, змінних Var і констант Val. Елемент <topology> містить дві групи елементів: <nodes>, де міститься інформація про місця й переходи мережі, і <arcs>, де

зберігається інформація про дуги. Індекси GUID усіх об'єктів моделі зберігаються як атрибути відповідних елементів XML-документа.

Для вузлів мережі зберігаються координати їх образів на робочій панелі, що дозволяє зберігати не тільки значущу інформацію моделі, але й параметри відображення (останнє надто важливо при роботі з більшими мережами). Методи `Save_Xmldatafile` і `Open_Xmldatafile` класу `Windowviewmodel` дозволяють зберігати інформацію про модель РМП РЧ ТЛА й витягати її з текстових файлів власного формату `.rtcrn`.

Використання XML-розмітки уможливорює також і початковий аналіз моделі по збереженому файлу. Таким чином, застосування технології XML дозволило реалізувати зручне зберігання файлів розроблювальних моделей РМП РЧ ТЛА, що є вкрай важливим етапом розробки інструментарію.

```
<?xml version="1.0" encoding="utf-8"?>
  <net>
    <declaration>
      <colset ID="a43eeffd-0956-42d9-aa7f-cb9368291f16" name="set1"> <colors>
        <color ID="3325fd9e-45e2-4751-bfb1-2f8e6d002c10" name="color1" /> </colors>
      <variables>
        <variable ID="29e07fbd-9cb6-4998-854f-32ebd28e3586" name="variable1" /> </variables>
      <values></values>
    </colset>
  </declaration>
  <topology>
    <nodes>
      <node ID="3b5f52fa-549c-44a9-9948-614658b21355" type="Place"
        name="Place1" colsetid="a43eeffd-0956-42d9-aa7f-cb9368291f16" timestamp="0">
        <tokens>
          <token colorid="3325fd9e-45e2-4751-bfb1-2f8e6d002c10" count="1" /> </tokens>
        <canvas position_x="79" position_y="75" /> </node>
      <node ID="e01ef563-e57c-40f1-9f7a-eb7fd08e368e" type="Transition" name="Transition" priority="0">
        <Guard>
          <conjunct>
            <expression type="Mark">
              <term_1 ID="29e07fbd-9cb6-4998-854f-32ebd28e3586" /> <term_2 ID="3325fd9e-
                45e2-4751-bfb1-2f8e6d002c10" />
            </expression>
          </conjunct>
        </Guard>
      </node>
    </nodes>
  </topology>
</net>
```

Рисунок 4.5 – Приклад Xml-Документа, що зберігає дані моделі РМП РЧ ТЛА

Форми для обчислень – перша містить кнопки додавання нових місць і переходів, друга – елементи керування моделюванням:

- кнопка Start, що запускає процес моделювання;
- кнопка Stop, що зупиняє процес моделювання;
- поле Timestep, що дозволяє визначити крок моделювання в секундах;
- кнопка Step by step, що дозволяє зробити один крок моделювання;
- кнопка повернення у вихідний стан Return to initial.

Створення моделі РМП РЧ ТЛА здійснюється в області моделювання. Елементарна мережа, що розташована в області моделювання на рис. 4.6, побудована за наведеним раніше прикладом XML-документа.

Рядок стану служить для виведення повідомлень системи. У правому куті рядка перебуває годинник.

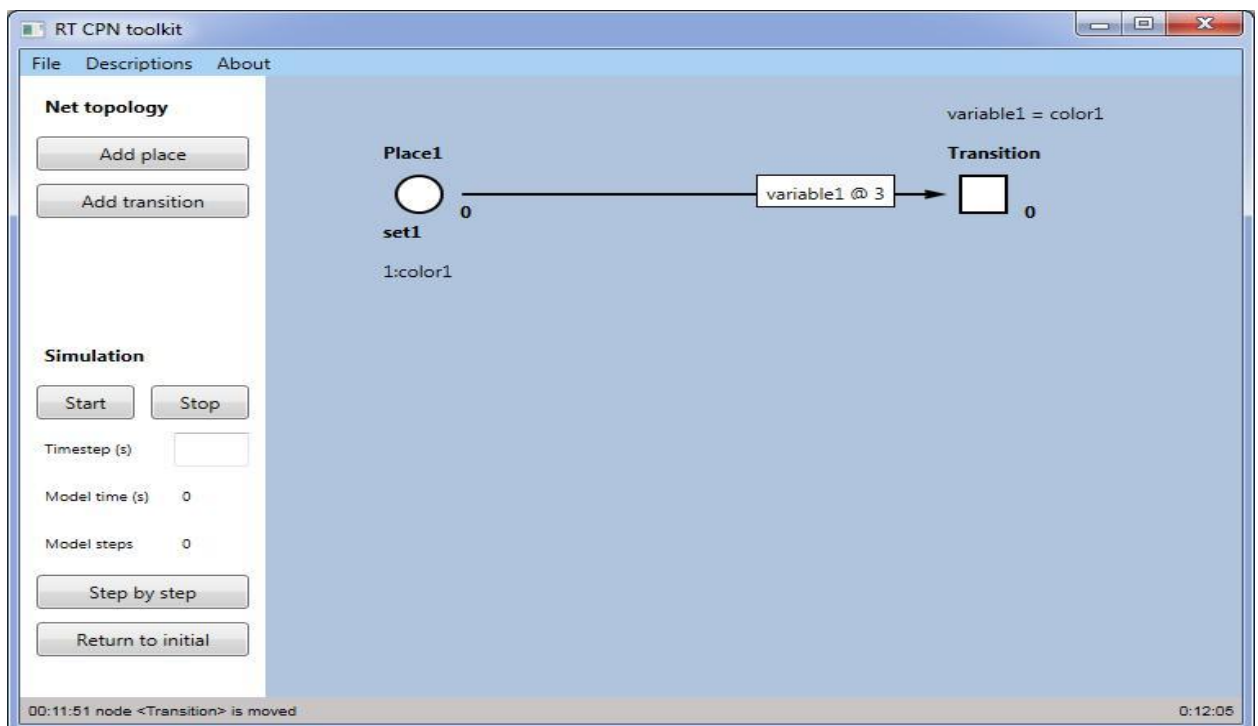


Рисунок 4.6 – Головне вікно інструментарію

Створення ІМ починається з визначення множини кольорів, змінних і констант. Для цього використовується вікно налаштувань типів і змінних DeclarationSettingsWindow (рис. 4.7). У новій моделі за замовчуванням створюється мінімальний можливий набір параметрів: одна множина кольорів і

один колір. Використовуючи вікно DeclarationSettingsWindow, користувач може створювати, змінювати й видаляти множини кольорів, самі кольори, змінні й константи. Існує можливість додавання опису редагуємих елементів.

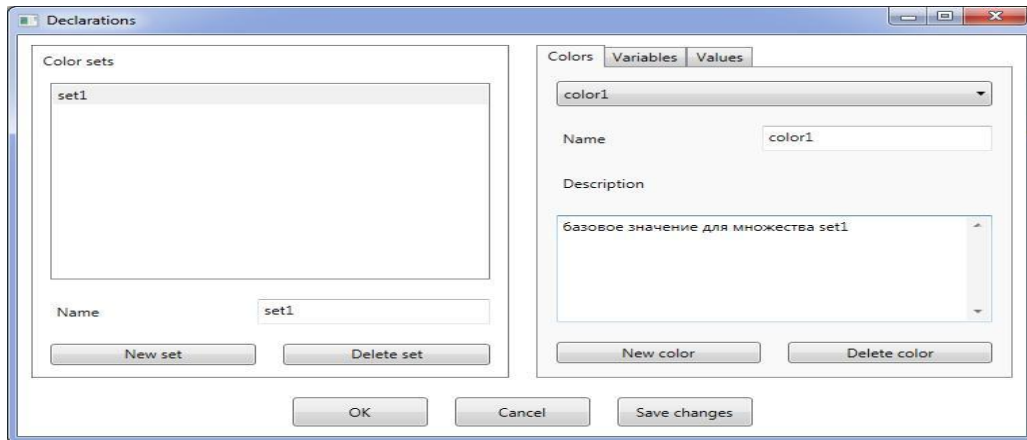


Рисунок 4.7 – Вікно налаштувань типів і змінних

Наступним кроком після визначення типів і змінних є створення топології мережі: додавання місць, переходів і дуг між ними. Місця й переходи можна створювати, використовуючи відповідні кнопки підобласті Net topology.

При створенні нового місця буде викликане вікно налаштувань елемента мережі ElementSettingsWindow у режимі налаштувань місця (рис. 4.8а), де можна задати:

- назву місця (поле Place name);
- множину кольорів місця (поле Place color set);
- початкове маркування (поле Place Initial marking);
- початкове значення часової мітки (поле Place initial timestamp).

Нетривіальна (непуста) маркування місця задається наступним синтаксисом:

`<Marking> ::= <Col_term> | <Marking> + <Col_term>`

`<Col_term> ::= <nat>:<color> | <color>`

`<nat>` – натуральне число, `<color>` – назва кольору.

RT CPN place settings

Place name: Place1

Place color set: set1

Place initial marking: 2:color1+3:color2

Place initial time stamp: 10

OK Cancel Save changes

a)

RT CPN place settings

Transition name: Transition

Guard function: variable1 = color1 | Place1(color1) {m} Place2(color4)

Transition priority: 1

OK Cancel Save changes

б)

RT CPN place settings

Arc starts at Place named: Place1

Arc ends at Transition named: Transition

Arc expression function: variable1+2:color2

Time expression function: 3

OK Cancel Save changes

в)

Рисунок – 4.8 – Вікна налаштувань елементів мережі:

а) у режимі налаштувань місця; б) у режимі налаштувань переходу;

в) у режимі налаштувань дуги

Правильність уведеного рядка визначається методом `Parse_InitialMarking` класу `ElementSettingsWindow`.

При створенні нового переходу буде викликане вікно налаштувань елемента мережі `ElementSettingsWindow` у режимі налаштувань переходу (рис. 4.8б), де можна задати:

- назва переходу (поле `Transition name`);
- захисну функцію (поле `Guard function`);
- пріоритет переходу (поле `Transition priority`).

Синтаксис захисної функції, з метою спрощення завдання значень були введені наступні зміни:

Правильність уведеного рядка визначається методами `Parse_Guard_function` і `Parse_TimenodeInterval` класу `ElementSettingsWindow`. Створення дуги здійснюється за допомогою пункту `Add ark` контекстного меню, що виникає при натисканні правої кнопки миші на існуючому вузлі в робочій області. Після цього необхідно розмістити курсор над вузлом, у який повинна привести дуга, і зробити натиснути ліву кнопку миші. Якщо з'єднуються вузли одного класу (місця або переходи), або вже існує дуга вузли, що з'єднують ці, будуть виведені відповідні повідомлення. А якщо ні, то буде додана нова дуга, при цьому буде викликане вікно налаштувань елемента мережі `ElementSettingsWindow` у режимі налаштувань дуги (рис. 4.8в), де можна задати:

- ваговий вираз дуги (поле `Arc expression function`);
- часовий вираз дуги (поле `Time expression function`).

Синтаксис вагового виразу дуги збігається із синтаксисом маркування місця з тою різницею, що в якості елементів `<color>` можна використовувати не тільки кольори, але й змінні або константи множини кольорів того місця, яке дуга з'єднує з переходом. Правильність уведеного рядка визначається методом `Parse_Arcexpression` класу `ElementSettingsWindow`.

Вікно налаштувань елементів мережі в процесі редагування мережі можна викликати, використовуючи контекстне меню. Таким чином, реалізований досить простий і зручний графічний інтерфейс користувача, що дозволяє виконувати всі необхідні дії для роботи з моделлю РМП РЧ ТЛА й орієнтований як на автономне використання, так і на застосування в ДІС типу ІСППР РЧ.

Наведений опис прототипу інструментарію моделювання ДС на основі РМП РЧ ТЛА, розробленого в середовищі G2 – об'єктно-орієнтованого інтегрованого середовища для розробки й супроводу додатків реального часу, що використовують бази знань. Показано, що середовище G2 є зручним інструментом для імітаційного моделювання ДС на основі РМП РЧ ТЛА. Робота програмного додатка проілюстрована прикладом. Показано, що за допомогою G2 можна відносно просто створювати різні варіанти МП при наявності темпоральних обмежень. Відзначені недоліки комплексу G2 – висока вартість комплексу й істотне звуження області застосовності розроблювального програмного продукту (додатка).

Розроблена повна й досить універсальна в плані як застосування у складі ДС типу ІСППР РЧ, так і автономного використання версія програмного продукту в середовищі Microsoft Visual Studio мовою C#. Даний опис архітектури розробленого інструментарію на основі шаблону Model-View-View-Mmodel, і структури даних моделі. Наведені оригінальні алгоритми обчислень логічних функцій на основі виражень ТЛА, алгоритми визначення припустимих переходів, розв'язання конфліктів переходів, спрацьовування переходів.

5 ОПИС МОЖЛИВОСТІ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Перевірка ефективності використання запропонованого підходу до моделювання й розробленого на його основі математичного й програмного забезпечення (інструментарію) проводилася шляхом створення на їх основі як відповідних додатків, використовуваних на етапі тестування, так і окремих компонентів для моделювання ДС.

Тестування розроблених інструментальних засобів було проведено з використанням комплексу заздалегідь підготовлених прикладів. Більша частина цих прикладів являє собою штучні конструкції, що дозволяють перевірити коректність запропонованих алгоритмів функціонування ІМ на основі РМП РЧ ТЛА, зокрема, алгоритмів визначення допустимості переходу й дозволу переходів, описаних у попередній главі. Проведене функціональне тестування дозволило вдосконалити ряд кроків алгоритмів, підвищити продуктивність програмного продукту.

Інша група прикладів була спрямована на дослідження залежності середнього часу виконання кроку (відповідних локальних алгоритмів) алгоритму моделювання від середньої кількості вільних змінних у захисних функціях переходів мережі. Графіки, що побудовані за результатами дослідження, наведені на рис. 5.1. Час виконання алгоритму моделювання сильно зростає при збільшенні числа вільних змінних у захисних функціях, оскільки складність алгоритму перевірки захисної функції переходу експоненційно залежить від вільних змінних.

Варто відзначити, що, незважаючи на значні кількості переходів у тестуємих прикладах, час обробки одного кроку не перевищило 0,2 секунди. При моделюванні реальних систем (процесів) у складних ДС кількість переходів звичайно суттєво менше, що сприяє зменшенню часу роботи програми.

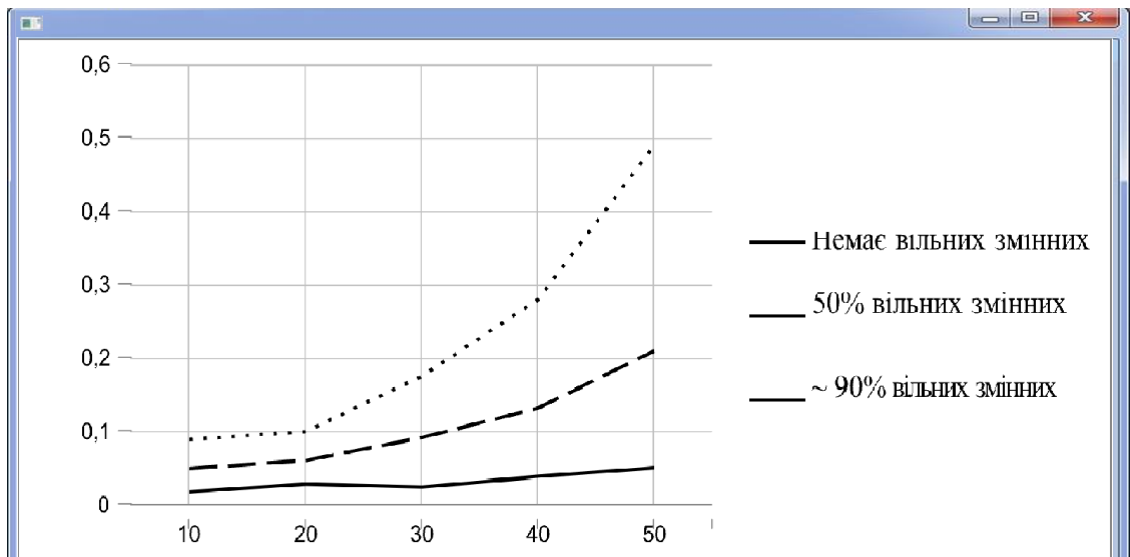


Рисунок 5.1 – Залежність часу виконання кроку моделювання від кількості неконфліктних переходів у мережі

За допомогою розроблених інструментальних програмних засобів були реалізовані ІМ на прикладі автоматичної системи на основі РМП РЧ і на основі РМП РЧ ТЛА.

ВИСНОВКИ

Досліджені підходи до моделювання інформаційних систем як складних ДС і темпоральних залежностей у таких системах і їх застосовність для проектування й аналізу моделей складних ДС. Обґрунтований підхід на основі імітаційного моделювання. Для реалізації ІМ запропоновано використовувати формалізм модифікованих МП – РМП РЧ.

Досліджений формалізм РМП РЧ ТЛА для моделювання складних ДС із можливістю виведення й оперування темпоральною інформацією, що базується на модифікації РМП, що й дозволяє використовувати в якості захисних функцій переходів мережі вираження ТЛА, що визначають темпоральні зв'язки між місцями мережі. Розроблені алгоритми на основі РМП РЧ ТЛА – визначення допустимості переходу, розв'язання конфлікту переходу, спрацьовування переходу, що мають поліноміальну оцінку складності, що дозволяє їх застосування в ДС типу ІСППР РЧ.

Запропоновані методи аналізу й верифікації моделей ДС, створених на основі РМП РЧ ТЛА. Результати аналізу існуючих підходів до верифікації показали, що найбільш перспективним є алгоритм Model Checking, основною перевагою якого є можливість повної автоматизації доказу поведінкових властивостей системи. Обґрунтоване застосування підходу Model Checking для верифікації моделей процесів у ДС і запропонований алгоритм верифікації для ІМ на основі РМП РЧ ТЛА, орієнтований на використання в ІСППР РЧ.

Виконана програмна реалізація розроблених методів і алгоритмів ІМ на основі РМП РЧ ТЛА: у середовищі G2 розроблений прототип інструментарію ІМ; у середовищі Microsoft Visual Studio розроблена повна версія програмного продукту, що дозволяє як автономно застосувати систему ІМ складних ДС, так і в складі ДС типу ІСППР РЧ. Наведений опис архітектури розробленого інструментарію, реалізовані оригінальні алгоритми обчислень логічних функцій на основі виразів ТЛА. Результати тестування підтвердили доцільність

застосування розробленого математичного й програмного забезпечення для моделювання процесів у складних ДС.

Проведена експериментальна апробація розроблених методів і інструментальних програмних засобів: у рамках робіт з модернізації електродинамічних моделей, а саме: для розробки моделей об'єктів ЕЕС; показано, що ІМ на основі РМП РЧ ТЛА зручно використовувати при проведенні наукових досліджень в області ЕЕС, розроблена модульна архітектура й процес розробки додатків для моделювання об'єктів. Розроблені засоби дозволяють формалізувати й спрощувати процес моделювання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Антипов С. Г., Фомина М. В. Метод формирования обобщенных понятий с использованием темпоральных деревьев решений // Искусственный интеллект и принятие решений. – 2015. – Т. 2. – С. 64–76.
2. Antipov S.G., Fomina, M.V. A method for compiling general concepts with the use of temporal decision trees // Scientific and Technical Information Processing. – 2011. – Vol. 38, no. 6. – Pp. 409–419. – URL: <http://dx.doi.org/10.3103/S0147688211060025>.
3. С. Г. Антипов, М. В. Фомина. Проблема обнаружения аномалий в наборах временных рядов // Программные продукты и системы. – 2012. – Т. 2. – С. 78–82.
4. Vagin, V.N., Fomina, M.V., Antipov, S.G. Modeling of algorithms of inductive concept formation in “noisy” databases // Automatic Documentation and Mathematical Linguistics. – 2013. – Vol. 47, no. 4. – Pp. 151–161. – URL: <http://dx.doi.org/10.3103/S0005105513040055>.
5. Вагин В.Н., Фомина М.В., Антипов С.Г. Моделирование алгоритмов индуктивного формирования понятий в «зашумленных» базах данных // Научно-техническая информация. Информационные процессы и системы. – 2013. – Т. 7. – С. 20–32.
6. Антипов С. Г. Концептуальная схема базы данных для исследования алгоритмов индуктивного формирования понятий при наличии шума в данных // Радиоэлектроника, электротехника и энергетика. – Т. 1. – Москва: Издательский дом МЭИ, 2009. – С. 275–276.
7. Антипов С. Г., Фомина М. В. Проблема формирования обобщенных понятий при наличии шума в решающих атрибутах // Четырнадцатая конференция по искусственному интеллекту международным участием КИИ 2014 (24-27 сентября 2014 г., г. Казань, Россия): Труды конференции. – Т. 2. – Казань: Изд-во РИЦ «Школа», 2014. – С. 204–212.

8. C. Stefano, C. Sansone, M. Vento. To reject or not to reject: that is the question - an answer in case of neural classifiers // *IEEE Transactions on Systems, Management and Cybernetics*. – 2000. – Vol. 1. – Pp. 84–94.
9. Гаазе-Рапопорт М. Г., Поспелов Д. А. Толковый словарь по искусственному интеллекту. – М.: Радио и связь, 1992. – Р. 256.
10. Арский Ю.М., Финн В. К. Принципы конструирования интеллектуальных систем // *Информационные технологии и вычислительные системы*. – 2008. – № 4. – С. 4–37.
11. Башмаков А. И., Интеллектуальные информационные технологии: Учеб. пособие. – Москва: Изд. МГТУ им. Н. Э. Баумана, 2005. – 304 с.
12. Еремеев А.П., Троицкий В.В. Модели представления временных зависимостей в интеллектуальных системах поддержки принятия решений // *Известия РАН. Теория и системы управления*. – 2003. – № 5. – С. 75–88.
13. Thomas G. Dietterich, Ryszard S. Michalski. Inductive Learning of Structural Descriptions: Evaluation Criteria and Comparative Review of Selected Methods // *Artif. Intell.* – 1981. – Vol. 16, no. 3. – Pp. 257–294.
14. Гарратано Дж., Райли Г. Экспертные системы: принципы разработки и программирование. – 4 изд. – Москва: ООО «И.Д. Вильямс», 2007. – 1152 с.
15. Поспелов Д. А.. Моделирование рассуждений. Опыт анализа мыслительных актов. – Москва: Радио и связь, 1989. – 184 с.
16. Newell A., Simon H. A. Computer science as empirical inquiry: symbols and search // *Commun. ACM*. – 2016. – mar. – Vol. 19, no. 3. – Pp. 113–126. – URL: <http://doi.acm.org/10.1145/360018.360022>.
17. Collins A.M., Quillian M.R. Retrieval time from semantic memory // *Journal of Verbal Learning and Verbal Behavior*. – 2009. – Vol. 8. – Pp. 240–248.
18. Вагин В. Н. Дедукция и обобщение в системах принятия решений. – Москва: Наука, 1988. – 384 pp.
19. Дудар З.В. Порівняння методів прогнозування часових рядів / З.В. Дудар, М.С. Широкопетлева, О.А. Пономаренко // *Бионика интеллекта*. – Харьков: ХНУРЭ, 2018. – Вип.2 (91). – С.41-47

20. Sowa F. J. *Encyclopedia of Artificial Intelligence*. – 2nd edition. – New York, NY, USA: John Wiley & Sons, Inc., 2012.
21. Minsky M. *A Framework for Representing Knowledge: Tech. Rep.* : Cambridge, MA, USA, 1974.
22. Axelrod R. M. *Structure of decision: the cognitive maps of political elites* / edited by Robert Axelrod; written under the auspices of the Institute of International Studies, University of California (Berkeley) and the Institute of Public Policy Studies, the University Michigan. – Princeton University Press, Princeton, N.J., 2016. – P. 404.
23. Tolman E. C. *Cognitive maps in rats and men*. // *Psychological review*. – 2018. – jul. – Vol. 55, no. 4. – Pp. 189–208.
24. Люгер Дж. Ф. *Искусственный интеллект: стратегии и методы решения сложных проблем*. – 4-е издание изд. – Москва: Издательский дом «Вильямс», 2013. – 864 с.
25. McCulloch W. S., Pitts W. *A logical calculus of the ideas immanent in nervous activity* // *Bulletin of Mathematical Biology*. – 2017. – dec. – Vol. 5, no. 4. – Pp. 115–133. – URL: <http://dx.doi.org/10.1007/BF02478259>.
26. *Outlier Detection Using Replicator Neural Networks* / Simon Hawkins, Hongx-ing He, Graham Williams, Rohan Baxter // *In Proc. of the Fifth Int. Conf. and Data Warehousing and Knowledge Discovery (DaWaK02)*. – 2012. – Pp. 170–180.
27. *Constructing Boosting Algorithms from SVMs: An Application to One-Class Classification* / Ratsch, Gunnar, Mika, Sebastian, Scholkopf, Bernhard, Muller, Klaus-Robert // *IEEE Trans. Pattern Anal. Mach. Intell.* – 2019. – sep. – Vol. 24, no. 9. – Pp. 1184–1199. – URL: <http://dx.doi.org/10.1109/TPAMI.2002.1033211>.
28. *Using artificial anomalies to detect unknown and known network intrusions* / Fan, W., Miller, M., Stolfo, S. et al. // *Knowl. Inf. Syst.* – 2018. – sep. – Vol. 6, no. 5. – Pp. 507–527. – URL: <http://dx.doi.org/10.1007/s10115-003-0132-7>.