І.А. КОТОВ, канд. техн. наук, доц.
Криворізький національний університет

АВТОМАТНА МОДЕЛЬ РЕАКТИВНОЇ ТРИГЕРНОЇ СППР ДЛЯ УПРАВЛІННЯ РЕЖИМОМ ЕНЕРГОСИСТЕМИ ГІРНИЧО-МЕТАЛУРГІЙНОГО КОМПЛЕКСУ

Метою статті є виклад результатів розробки автоматної моделі реактивної тригерної системи підтримки прийняття рішень (СППР) для управління режимом енергосистеми. В системі реалізована властивість реактивності за рахунок використання запропонованих апаратно-програмних тригерів. Модель реактивної тригерної системи підтримки рішень інтерактивно пов'язана як з діями користувача-оператора, так і зі станами компонентів енергосистеми. Теоретична розробка і впровадження програмного комплексу тригерної системи підтримки прийняття рішень в середу диспетчерського управління енергосистеми є актуальною науково-технічною проблемою.

Методи дослідження полягають в комплексуванні автоматної моделі станів та тригерної моделі функціонування програмної системи підтримки рішень. При цьому стан автоматної моделі пов'язується з виконанням пакета метаправил для управління логічним висновком. У роботі використані методи теорії множин, математичної логіки, теорії автоматів, електроенергетичних систем (ЕЕС), теорії графів, математичної статистики.

Наукова новизна полягає в новій моделі подієвої взаємодії системи підтримки рішень з енергооб'єктами електроенергетичної системи. Для схеми станів реалізована автоматна модель функціонування з конкретизацією семантики станів, транзакцій і тригерів. Розроблена модель візуалізації автоматного формалізму у вигляді керованих тригерами станів програмної системи. Показано, що структурна схема тригерних станів реалізує базовий набір структур алгоритму. Розроблена схема візуалізації виконання транзакцій, як одного стану програмної системи.

Практична значимість роботи полягає в удосконаленні автоматизації управління режимами енергосистеми шляхом впровадження в середу діючого оперативного інформаційно-управляючого комплексу (ОІУК) розробленої автоматної моделі реактивної тригерної системи підтримки прийняття рішень. Реалізація та впровадження програмного комплексу дозволить скоротити збитки від аварійних ситуацій в енергосистемі, підвищити якість диспетчерського управління режимами і технологічними процесами електричних мереж.

Результатами роботи є подієва модель станів системи підтримки рішень, модель апаратно-програмних тригерів для зв'язку з компонентами енергосистеми, автоматна тригерна модель станів, асоційованих з транзакціями метаправил для запуску та управління логічним висновком в базі знань. Дано визначення стану моделі і транзакції метаправил. Розроблені формальні моделі переходів, транзакцій і функцій виходів станів автоматної моделі. Наведено візуалізація тригерної моделі і принципи її функціонування.

Ключові слова: транзакція, енергосистема, автомат, онтологія, евристика, база знань, інкорпорація

doi: 10.31721/2306-5451-2019-1-49-36-42

Проблема і її зв'язок з науковими і практичними завданнями. Визначальними якостями функціонування СППР є складність і об'ємність об'єкту управління – енергосистеми гірничо-металургійного комплексу, зв'язок із засобами ОІУК автоматизованої системи диспетчерського управління (АСДУ), необхідність роботи як в режимі реального часу, так і інтерактивно – в якості порадника оперативно-диспетчерського персоналу (ОДП), робота з різними формами подання знань, поєднання обчислювальних завдань і логічного висновку та інші.

Перераховані якості формують ряд особливих вимог як до самої інтелектуальної системи, так і до методології її проектування. Приведемо базові з таких вимог: одночасна робота з декларативними і процедурними формами подання знань, однакова робота системи в режимі логічного висновку в базі знань (БЗ) і обробки бази даних (БД), забезпечення практичного апаратного інтерфейсу з компонентами енергооб'єктів і АСДУ, реалізація призначеного для користувача візуального інтерфейсу з ОДП, ефективна трансляція у внутрішні коди системи і інтерпретація управляючих метаправил, реалізація програмного механізму логічного висновку у вигляді транзакцій метаправил.

Сформульовані вимоги обумовлюють необхідність нового ексклюзивного підходу до подання і візуалізації структурних і функціональних моделей програмного комплексу СППР. Засоби візуалізації моделі функціонування СППР повинні забезпечувати алгоритмічну точність і простоту моделі, однозначну інтерпретацію графічних примітивів, відображення станів програми і бази знань, моделювання подій і реактивність інтелектуальної системи, використання мінімального формального символічного алфавіту. При цьому, модель повинна вирішувати наступні головні завдання: забезпечення однакового представлення алгоритмів функціонування СППР і алгоритмів трансляції і обробки бази знань на усіх рівнях, репрезентації дієвого вико-

нання програми залежно від параметрів (сигналів) станів зовнішнього оточення інтелектуальної системи – комплексу АСДУ, ОДП і енергооб'єктів ЕЕС.

Аналіз досліджень і публікацій. Проведемо короткий аналіз існуючих функціональних моделей візуалізації алгоритмів (сценаріїв, поведінки), які найбільш адекватно реалізують поставлені завдання [1, 2]. Кожен з формалізмів має сильні і слабкі сторони, а також переважну область застосування. Більш важливим є аналіз їх обмежень.

Спочатку необхідно відзначити текстуальні форми запису алгоритмів. Ця форма є найбільш природною і найменш формалізованою для людини. Вона може застосовуватися на самому початковому етапі концептуального опису завдання для обговорення в колективі розробників. Однак незначне ускладнення алгоритму або деталіровка завдання призводять до різкого розростання обсягу опису, заплутування викладу, зникнення наочності і появи суперечливості.

Різновидом текстуальної записи алгоритму є псевдокод. Цей підхід корисний, коли необхідно абстрагуватися від конкретного середовища і мови програмування, представляючи, однак, якусь структуру програми. Цей підхід корисний для ілюстрації невеликих і ізольованих фрагментів алгоритмів. В іншому він несе обмеження текстуальної форми запису.

У деяких випадках буває корисним безпосередній запис алгоритму в рамках деякої мови програмування. Цей підхід використовується, коли алгоритм вже відомий і потрібно відпрацювання деяких нюансів, приписок, модифікацій. Крім того, даний підхід можна застосувати, в основному, в колективах розробників, вже знайомих з попередніми варіантами вихідного коду. Інакше, вивчення чужого вихідного коду вимагатиме занадто багато часу.

Кращим і найбільш масовим є формалізм блок-схем, як найбільш поширений образ відображення алгоритмів, який досить підходить для безлічі добре формалізованих задач [1, 2]. Формальна мова схем алгоритмів стандартизована і добре випробувана на практиці. У більшості випадків, спираючись на талант розробника, вдається звести задачу до візуалізації блок-схемою алгоритму. Однак, у міру зростання складності та ієрархічності завдання, схема алгоритму втрачає логіку в кількості зв'язків і переходів. Можна навіть зазначити, що «покрокова блок-схема є прикритим анахронізмом, придатним тільки для новачків в алгоритмічному мисленні» [3]. У багатьох випадках схеми алгоритмів рисуються навіть після того, як програма закінчена [1].

Діаграми Дейкстри визначають деякий альтернативний підхід до візуального моделювання алгоритмів [4, 5]. Дейкстра ввів принципи структуризації блок-схем: принцип обмеження топології, принцип вертикальної орієнтації входів і виходів, принцип єдиної вертикалі, принцип нанизання типових блок-схем на єдину вертикаль [6]. Підхід Дейкстри випередив час, він мінімізував типізацію блоків схеми і способи їх з'єднання, але сьогодні, в основному, також несе недоліки сучасних блок-схем алгоритмів.

З метою «ущільнення» схем алгоритмів і ліквідації їх заплутаності запропонований метод візуалізації Нессі-Шнейдермана [7]. Він забезпечує абсолютне дотримання принципів структурного програмування, покрокову деталізацію завдання при відсутності зв'язуючих стрілок. До недоліків схем Нессі-Шнейдермана слід віднести великий обсяг загальної схеми, яка важко сприймається. При відсутності зв'язуючих стрілок збільшується кількість меж блоків. Зменшення розмірів схеми зменшує внутрішні відділи і знижує їх читабельність.

Розглянуті формальні системи подання алгоритмів застосовуються і розвиваються, проте, вони не забезпечують механізмів відображення взаємодії системи з зовнішнім середовищем. Базовим формальним підходом для подання реактивних систем є комплекс різних варіантів автоматних моделей. Головним достоїнством таких моделей є врахування безлічі станів системи і впливу операцій на логіку переходів в наступні стани. Графічним поданням автоматної моделі є добре відомий граф переходів. Формалізми автоматів виражаються моделями автоматів Мілі і Мура [2, 8]. Аналіз робіт дозволяє відзначити наступні обмеження автоматних моделей при формалізації алгоритмів – відсутність ієрархії станів, узагальнення переходів, засобів відображення відновлення нормальної роботи після переривання, подання семантики взаємодії кінцевих автоматів [2].

В якості деяких рішень зазначених проблем використовуються SWITCH-технологія проектування [9], а також діаграми станів Д. Харела [10, 11].

В основі SWITCH-технології лежить можливість подання програми як ієрархії взаємодіючих автоматів. Модель дозволяє реєструвати зовнішні події. До недоліків моделі можна віднес-

ти «необхідність підтримки великих переліків подій, вхідних і вихідних впливів, представлених в кінцевих програмних модулях у вигляді формальних ідентифікаторів (буква + номер), що зводяться, в кінцевому рахунку, до числових кодів (номерів)», «необхідність ретельного протоколювання глобального списку подій» [9].

Візуальний формалізм діаграми станів Д. Харела є кроком вперед по відношенню до автоматної моделі, відрізняється високою наочністю і додатковими можливостями – ієрархією (реалізація суперстанів), моделюванням тимчасових затримок, наявністю історії станів. Багато розробників сьогодні спирається на дану модель, вона дозволяє реалізовувати внутрішню реактивність системи по відношенню до умов і станів. Однак, відсутня можливість візуалізувати вплив подій зовнішнього середовища на умови запуску станів.

Р-схеми – це технологія візуалізації проектування систем, яка була розроблена і застосована в інституті кібернетики Академії наук України під керівництвом В.М. Глушкова [12, 13]. Про доцільність застосування Р-схем говорить прийняття стандарту [13]. Базова частина Р-схеми в цілому відповідає розширеній мережі переходів, заснованої на моделі автомата Мілі. Недоліком формалізму Р-моделі є концентрація на поданні потоку управління і відсутність відображення потоків даних. Також складно представляються кілька точок входу в модель.

Мова ДРАКОН розроблена як засіб структурно-імперативного проектування топології схем програм з метою їх кращої структуризації і формалізації [14]. Реалізовано дві моделі: алгоритмічного примітиву для порівняно простих завдань і «силуету» – для складно-структурованих алгоритмів. Мова має потужні логічні і виразні властивості, підхід підтверджений практичним застосуванням. Недоліками можна вважати велику кількість графічних примітивів (іконок – 25 штук, макроіконок – 20 штук), дві моделі подання завдань, відсутність явного подання подій і взаємодії із зовнішнім середовищем.

Оригінальним рішенням в області моделювання обчислювальних процесів і проектування програмних комплексів є формалізм L-мереж, запропонований М. Ф. Лекаревим [15, 16]. Позитивними особливостями моделі візуалізації є глибоке теоретичне обґрунтування, лаконічна формальна мова, однакове подання «малих» і «великих» функціональних структур, однозначна інтерпретація схеми, «незалежність обчислювальної платформи, яка визначається L-мережею, від мови програмування», підвищення ефективності праці програміста на 1 – 2 порядків [11].

Узагальнюючи розглянуті вище основні підходи до моделювання та візуалізації процесів проектування складних програмних систем і відзначаючи їх безсумнівну корисність в певних областях і умовах, слід сказати, що вони не цілком відповідають завданням проектування подієвої інтелектуальної системи – СППР, що використовує інкорпоровану базу знань, керовану метаправилами, що активізуються зовнішнім середовищем. Тому сформульовані вище вимоги до засобів візуалізації моделі структурно-інформаційної схеми програмного комплексу СППР обумовлюють актуальність і необхідність розробки нових синтетичних засобів подання структури і функціонування СППР, яка взаємодіє з комплексом АСДУ енергосистеми і оперативним персоналом.

Постановка завдання. З огляду на сильні якості і обмеження проаналізованих підходів до проектування програмних систем є актуальною задача розробки синтетичної тригерної моделі візуалізації алгоритму (сценарію) функціонування програмного комплексу СППР, керованої подіями зовнішнього середовища.

Сформулюємо основні вимоги до системи візуалізації проектування, що впливають із змісту і специфіки обраної професійної області. Формалізм візуалізації функціонального поведінки інтелектуальної системи, пов'язаної із зовнішнім середовищем АСДУ, повинен забезпечувати наступне:

- реалізацію як обчислювальних задач, так і задач розпізнавання, трансляції, інтерпретації, поведінки і управління;
- використання мінімальної кількості концептуальних примітивів;
- мінімізацію кількості сполучних інформаційних трактів;
- виконання умови: один блок – одна операція (один пакет);
- наявність подієвих блоків зв'язку із зовнішнім середовищем (тригерів);
- наявність мінімальної кількості функціональних блоків з однаковим поведінкою;
- реалізацію функціональним блоком запиту до бази даних, метаправила до бази знань, однієї

операції або пакета операцій – транзакції;

наявність довільних подієвих точок входу в процес обчислень;

явне позначення станів;

явне завдання семантики станів;

забезпечення вкладеності і ієрархічності обчислювальних блоків.

Викладення матеріалу та результати. Реалізація зазначених необхідних властивостей моделі має спиратися на систему її станів. ОІУК АСДУ повертає цифрові дані про поточний стан режиму енергооб'єктів ЕЕС в інформаційні тракти, до яких підключається СППР. Програмні тригери поточного стану ідентифікують режимні параметри на вході і генерують ситуаційні коди на виході. Якщо СППР знаходиться в стані опитування тригерів, то вона отримує ситуаційний код і запускає відповідний йому стан. З даним станом асоційована транзакція (пакет метаправил), виконання якої реалізує операції по відношенню до БЗ (БД). Залежно від результату транзакції СППР переходить в наступний стан.

Розроблена узагальнена діаграма станів СППР, яка заснована на транзакціях метаправил, наведена на рис. 1. Діаграма поєднана зі схемою інформаційної взаємодії СППР з зовнішнім середовищем – апаратно-програмними засобами АСДУ ЕЕС.

Рис. 1. Діаграма станів тригерної СППР на основі транзакцій метаправил

Взявши за основу подієву модель станів функціонування СППР, наведену на рис. 1, формалізуємо механізм подання візуалізації її роботи, що задовольняє сформульованим вище вимогам. В якості стартового етапу приймемо узагальнену схему циклу управління.

Очевидно, що в даному випадку поведінка СППР може бути описана моделлю кінцевого автомата (КА) з виходами [17, 18]

$$M=(V, W, Q, q_0, F, \delta, \mu),$$

де V – вхідний алфавіт автомата СППР; W – вихідний алфавіт автомата СППР; Q – кінцева множина станів СППР; q_0 – початковий стан автомата СППР; F – не порожня множина фінальних станів автомата СППР, $F \subseteq Q$; δ – функція переходів автомата СППР, $\delta: Q \times V \rightarrow Q$; μ – функція виходів автомата СППР, $\mu: Q \times V \rightarrow W$.

На рис. 2 наведено приклад діаграми станів автомата з виходами.

Рис. 2. Приклад діаграми станів автомата з виходами

Для вирішення поставленого завдання – моделювання роботи СППР на основі транзакцій метатегорій – модернізуємо взяті за основу математичну модель КА. Конкретизуємо семантику поняття стану, транзакції і тригера в рамках пропонованої моделі.

Запропонована формальна модель операційного автомата програмного комплексу СППР, яка об'єднує стани і реалізований алгоритм обробки БЗ, може бути проілюстрована графом, представленим на рис. 3.

Рис. 3. Граф станів автоматної тригерної операційної моделі програмного комплексу СППР

На рисунку також показано зовнішнє середовище по відношенню до програмної системи. Тригери розглядаються як частина системи, але їх входи пов'язані з зовнішнім середовищем і не залежать від характеру і результатів роботи алгоритму.

Під станом автоматної моделі СППР – q – будемо розуміти логічно виділений етап її функціонування, пов'язаний з реалізацією однієї конкретної транзакції.

Під транзакцією – p – будемо розуміти обчислювальний блок, який реалізує послідовний набір команд, виразів, операцій і розглянутий в рамках поточної транзакції як один обчислювальний пакет. Кожна транзакція жорстко пов'язана з конкретним станом – q . Транзакції можуть містити набори метаправил для управління БЗ і виробництва логічного висновку, пакети запитів для управління БД, оператори математичних обчислень, команди управління поведінкою СППР і т.д. В результаті виконання транзакції формується результат транзакції – r – і символи (коди) з вихідного алфавіту системи W .

Під тригером – t – будемо розуміти зовнішній апаратний або програмний об'єкт, що генерує коди, які розпізнаються СППР як вхідний алфавіт – V – і переводять її в один, однозначно пов'язаний з даними тригером, стан – q .

Висновки та напрямки подальших досліджень. Розроблена тригерна схема станів і подієвої взаємодії СППР з енергооб'єктами ЕЕС. Для схеми станів реалізована автоматна модель функціонування СППР з конкретизацією семантики станів, транзакцій і тригерів. Узагальнена запропонована модель по відношенню до класичної моделі автомата з виходом. Розроблена модель візуалізації автоматного формалізму у вигляді керованих тригерами станів програмної системи. Кожний стан реалізує одну транзакцію метаправил. Результат виконання транзакції дає можливість розгалуження процесу обробки БЗ.

Напрямок подальших досліджень полягає в формалізації механізму виконання транзакцій і, на цій основі, мінімізації мови моделювання. Передбачається моделювати активізацію станів моделі за допомогою маркування вузлів її графа станів [19]. Передбачається розробка мови програмування тригерних транзакцій метаправил для управління базою знань.

Список літератури

1. **Пышкин Е.В.** Структуры данных и алгоритмы: реализация на C/C++ / Е.В. Пышкин – СПб.: ФТК СПбГПУ – 2009 – 200 с.
2. **Карпов Ю.Г.** Теория автоматов. Учебник для вузов – СПб.: Питер, 2003.- 208 с.
3. **Brooks, Frederick P., Jr.** The Mythical Man-Month. Essays on Software Engineering. Anniversary Edition. Addison – Wesley, 1995.
4. **Dahl O.-J., Dijkstra E.W., Hoare C.A.R.** Structured Programming. Academic Press. London and New York. 1972.
5. **Dijkstra, Edsger W.** A Discipline of Programming. Prentice-Hall Inc. Englewood Cliffs, N.J., 1976.
6. **Дал У., Дейкстра Э., Хоор К.** Структурное программирование – М.: Мир – 1975 – С. 25–28
7. **Cornelia M.Y., Marilyn L.S.** Nassi-Shneiderman charts an alternative to flowcharts for design – ACM SIGSOFT/BIGMETRICS Software and Assurance Workshop, November 1978 – pp. 386 – 393
8. **Поликарпова Н.И., Шалыто А.А.** Автоматное программирование – СПб.: Изд-во СПбГУ – 2008 – 167 с.
9. **Шалыто А.А., Туккель Н.И.** SWITCH-технология – автоматный подход к созданию программного обеспечения «реактивных» систем // «Программирование», №5 – 2001– с. 45 – 62
10. **Пышкин Е.В.** Структурное проектирование: основание и развитие методов. С примерами на языке C++: Учеб.пособие. – СПб.: ФТК СПбГПУ – 2005 – 324 с.
11. **Harel D.** On Visual Formalisms // Communications of the ACM – Vol.31, No.5, May 1988 – pp.514 – 530
12. **Вельбицкий И.В.** Безбумажная Р-технология проектирования широкого применения// Выч. техника социальных стран, вып.18 – М.: Финансы и статистика, 1985.- с. 21-39
13. ГОСТ 19.005-85. Р-схемы алгоритмов и программ. Обозначения условные графические и правила выполнения. – Действует с 01.07.1986.
14. **Калиногорский Н.А.** Автоматизация процесса разработки алгоритмов управления в интегрированной среде Дракон 2007-2010 / Сост. Н.А. Калиногорский – Новокузнецк: Изд. центр СибГИУ – 2013 – 50 с.
15. **Lekarev M.F.** Das graphische Verfahren der Software-Entwicklung für logisch komplizierte Anwendungen // Technische Berichte der Fachhochschule Hamburg, №25 (Aug.1993), s.36-38.
16. **Лекарев М.Ф.** Визуальный формализм для разработки программного обеспечения. – СПб.: СПбГТУ, Санкт-Петербург – 1997 – 95 с.
17. **Лекарев М.Ф.** L-сеть в сверхбольшом программном проекте – СПб.: СПбГТУ – 2000 – 96 с.
18. **Белоусов А. И.** Дискретная математика: учебник для вузов / А. И. Белоусов, С. Б. Ткачев; под ред. В. С. Зарубина, А. П. Крищенко – 5-е изд. – Москва: Издательство МГТУ им. Н. Э. Баумана – 2015 – 743 с.
19. **Лескин А.А., Мальцев П.А., Спиридонов А.М.** Сети Петри в моделировании и управлении – Л.: Наука – 1989 – 133 с.

Рукопис подано до редакції 25.09.2019