МАТЕМАТИЧНЕ ТА КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ

MATHEMATICAL AND COMPUTER MODELING

МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

УДК 004.9

АНАЛІЗ ПІДХОДІВ ДО МОДЕЛЮВАННЯ ТА ВЕРИФІКАЦІЇ КІБЕРФІЗИЧНИХ СИСТЕМ

Коротунов С. Ю. – аспірант кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

Табунщик Г. В. – канд. техн. наук, доцент, професор кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

АНОТАЦІЯ

Актуальність. Сучасні тенденції в продуктивності та складності вимог до використання систем вимагають принципово нових підходів до проектування, в яких кібернетичні та фізичні компоненти інтегруються на різних етапах. Кіберфізичні системи оточують людину майже у всіх сферах існування, починаючи з помешкань та транспорту і закінчуючи медичними апаратами та міжрегіональними електромережами. Тому верифікація та перевірка роботи таких систем є актуальною задачею сьогодення. В таких системах програмне забезпечення та фізичні підсистеми працюють у різних часових та просторових вимірах, взаємодіють різними способами. Розглянуто основні підходи до верифікації кіберфізичних систем. Об'єктом досліджень є процес верифікації кіберфізичних систем, предметом – методи верифікації кіберфізичних систем, моделі та логіки що використовуються при формальній верифікації.

Мета. Мета роботи полягає в проведенні аналізу підходів до верифікації кіберфізичних систем, з деталізацією окремих етапів, таких як вибір моделей, інструментів верифікації, та, власно, методів верифікації.

Метод. Основними методами, що викладені в роботі, є методи формальної верифікації кіберфізичних систем, а саме – симуляція, доведення теорем, символічне виконання та перевірка моделі. Детально розглянуто методологію методу перевірки моделі – модель Кріпке та темпоральні логіки: логіка дерев обчислень та логіка лінійного часу. Також проведено моделювання з використанням скінчених автоматів.

Результати. Виконано моделювання кіберфізичної системи у вигляді створення моделі Кріпке, що дозволило описати всі стани системи, необхідні для виконання формальної верифікації.

Висновки. Проведено дослідження характеристик кіберфізичних систем, виконано аналіз методів верифікації таких систем. Зазначені недоліки стандартної методології, які більш за все стосуються етапу моделювання кіберфізичних систем. Доведено найбільшу перспективність методу перевірки моделі, для якого розглянуто основну методологію. Дано характеристику моделям Кріпке та темпоральним логікам як основним елементам методу перевірки моделі. Показано можливість використання скінченних автоматів, а саме моделей Кріпке, для моделювання елементів кіберфізичної системи. Наукова новизна роботи полягає в тому що було розроблено моделі кіберфізичних систем, які, на відміну від існуючих, засновані на моделях Кріпке, що дозволяє зробити детальний опис усіх станів системи, що, у свою чергу, є важливим кроком для виконання верифікації такої системи. Практичною цінністю роботи є розроблені моделі електронезалежної станції альтернативної енергетики, що дозволять автоматизувати процес зарядки електричних транспортних засобів. Були реалізовані цифрові двійники, які дозволяють моделювати процеси електронезалежної станції альтернативної енергетики. Розроблені двійники використовуються при вивчені дисциплін при підготовки бакалаврів та магістрів спеціальності 121 комп'ютерні науки.

КЛЮЧОВІ СЛОВА: верифікація, кіберфізична система, модель Кріпке, скінченний автомат, темпоральна логіка, перевірка моделі, симуляція.

АБРЕВІАТУРИ

NCSWT – networked control systems wind tunnel;
SPF – symbolic pathfinder;
STHA – spatio-temporal hybrid automata;
WCPS – wireless cyber-physical simulator;
ДСА – детермінований скінченний автомат;
КФС – кіберфізична система;
ЛДО – логіка дерев обчислень;
ЛЛЧ – логіка лінійного часу;
НСА – недетермінований скінченний автомат;
СА – скінченний автомат.

DOI 10.15588/1607-3274-2020-3-5

НОМЕНКЛАТУРА

δ – ряд перехідних функцій;

π – шлях в моделі Кріпке;

Σ – кінцевий непустий вхідний алфавіт;

φ – формула темпоральних логік;

ψ – формула темпоральних логік;

AP – набір елементарних тверджень;

a – швидкість розпаду (вказує на швидкість зміни температури);

F – функція зміни стану системи;

G – набір приймаючих станів;

L – функція маркування, яка позначає кожний стан структури набором атомних тверджень;

M – модель Кріпке;

p – довільне елементарне твердження з набору атомних тверджень;

Q – кінцевий набор станів;

q – довільне елементарне твердження з набору атомних тверджень

q_0 – початковий стан;

R – перехідне відношення;

S – сукупність станів;

S_0 – набір початкових станів;

S_1 – довільний стан моделі з набору станів;

S_2 – довільний стан моделі з набору станів;

S_3 – довільний стан моделі з набору станів;

T – температура в кімнаті;

T_{heater} – температура нагрівача (номінальна потужність);

T_{max} – максимально допустима температура;

T_{min} – мінімально допустима температура;

T_{out} – температура поза кімнатою;

t – час;

u– сигнал, який вказує чи ввімкнений нагрівач;

x – стан системи (скалярний або векторний).

ВСТУП

Термін КФС був впроваджений у 2006 році Національним науковим фондом США [1]. Він описує широкий спектр складних, багатопрофільних інженерних систем наступного покоління, які інтегрують вбудовані обчислювальні технології (кібернетичну частину) в фізичну систему. КФС набувають популярності та розповсюдження. Практично кожен пристрій сьогодні має контролер, який зчитує вхідні дані через датчики, обробляє їх, а потім виконує дії за допомогою фізичних приводів та механізмів. КФС надають змогу поєднувати фізичні можливості (такі як рух, робота, або інші) з кібернетичними (обчислення, зв'язок, тощо) для вирішення проблем, які неможливо вирішити використовуючи виключно один вид можливостей.

Літаки [2], кардіостимулятори [3], сучасні електромережі [4] є прикладами таких систем, тому що робота фізичних компонентів визначається алгоритмами програмних компонентів, тобто кібернетичною складовою. Популярність КФС зростає в останні роки як в промисловості, так і в наукових дослідах, сьогодні вони представлені багатьма різноманітними критично важливими додатками [5].

DOI 10.15588/1607-3274-2020-3-5

В КФС програмне забезпечення і фізичні підсистеми тісно пов'язані між собою, вони працюють в різних часових і просторових вимірах, демонструючи різноманітні поведінкові структури і взаємодіють різними способами [6]. Сучасні тенденції в продуктивності та складності вимог до використання систем вимагають принципово нових підходів до проектування, в яких кібернетичні та фізичні компоненти інтегруються на різних етапах.

До основних властивостей КФС належать наступні [7]:

– високий ступінь автоматизації,

– реорганізація / реконфігурація динаміки,

– кібернетичні можливості кожного фізичного компонента,

– здатність працювати на різних масштабах,

– інтеграція на різних часових і просторових вимірах.

Через високу складність сучасних комп'ютерних систем (мільйони рядків коду та мільярди транзисторів) неможливо повністю уникнути помилок. Це стосується не тільки прикладного програмного забезпечення, але й критичних компонентів, таких як апаратні засоби (пристрої вводу-виводу, мікропроцесори) і програмного забезпечення (компілятори, операційні системи). Очевидно, що системи, в розробці та реалізації яких зроблені помилки, можуть в тій чи іншій ситуації вести себе непередбачуваними способами.

Важливо розуміти, що помилки в комп'ютерних системах не є винятковими. Згідно зі статистикою, середня кількість помилок на тисячу рядків неналагодженого коду коливається в межах 15–50 [8]. Більше того, існує тенденція до деградації якості конструкції (мабуть, це є наслідком зростаючої складності систем і оптимізації витрат на їх створення) [9].

Поєднання комп'ютерних наук та інженерії для реалізації КФС викликає значні технологічні проблеми, які роблять верифікацію функціонування таких систем критично необхідною. КФС потребують координування між різнорідними підсистемами, які складаються з обчислювальних пристроїв, розподілених датчиків та виконавчих механізмів [10]. Датчики та виконавчі елементи повинні забезпечувати інтерфейс між фізичним та кібернетичним рівнями та адаптуються до часу, що змінюється в фізичному та кібернетичному контексті.

Об'єктом дослідження є процес верифікації КФС.

Предметом дослідження – методи верифікації КФС, моделі та логіки що використовуються при формальній верифікації.

Мета роботи – проведення аналізу підходів до верифікації КФС, деталізуючи окремі етапи, такі як вибір моделей, інструментів верифікації, та, власно, методів верифікації.

1 ПОСТАНОВКА ЗАДАЧІ

КФС – це складні мережі обчислювальних систем, кожна з яких взаємодіє зі своїм фізичним середовищем шляхом обміну інформацією та енергією [11].

Подібні складні системи демонструють специфічну поведінку, що спостерігається лише тоді, коли мережева система розглядається в цілому, і не може бути оцінена простим додатком поведінки окремих компонентів. Проявами такої поведінки можуть бути макроскопічні властивості, які впливають на дії контролера, споживання ресурсів та експлуатаційні характеристики [12].

Одними з ключових інструментів, що використовуються для проектування та розробки КФС, є формальне моделювання та аналіз. Формальні моделі – це математичні конструкції, які можна використовувати для конкретизації поведінки складної системи з огляду на дискретну або безперервну зміну її параметрів.

Використовуючи формальні моделі, КФС можна сконструювати, використовуючи три етапи [13]:

– формальне моделювання: поведінку КФС можна відобразити за допомогою декількох безперервних або дискретних параметрів системи;

– верифікація: моделі можуть бути змодельовані у часі чи просторі або теоретично проаналізовані, щоб визначити, чи відповідає модель вимогам;

– валідація: моделі можуть бути реалізовані та оцінені в реальному середовищі або у віртуальній емуляції, що імітує середовищі реального світу, щоб перевірити, чи відповідає реалізація вимогам.

Одним з найбільш актуальних розділів у вивченні КФС сьогодні є саме верифікація таких систем [14]. Адже при повсюдному розповсюдженні КФС, вкрай важливим стає можливість довести що такі системи будуть відмовостійкими та надійними навіть в критичні моменти та при непередбачених ситуаціях.

2 ОГЛЯД ЛІТЕРАТУРИ

Верифікація – це процес перевірки відповідності системи (її моделі) вимогам, що пред'являються до неї. Якщо вона відповідає вимогам, така система називається коректною. В іншому випадку – некоректною, а факт невідповідності системи вимогам – помилкою. Таким чином, верифікацію можна визначити як процес аналізу системи на наявність або відсутність помилок в ній. Невизначений вирок також можливий, коли помилки не знайдені, але їхня відсутність не доведена.

Методи перевірки можна розділити на три основні групи [15]:

– формальні методи, які використовують математично строгий аналіз моделі програми та моделі вимог;

– методи тестування, що перевіряють фактичну поведінку програми на певному наборі сценаріїв;

– експертиза, проведена людьми на основі їх знань і досвіду безпосередньо щодо результатів проектування (наприклад, інспекція коду).

Кожна із зазначених груп методів має свої переваги і недоліки, кожна має свою область застосування. Повна верифікація складних систем відповідального призначення неможлива без спільного використання різних підходів. Ця робота присвячена формальним методам верифікації програм.

Формальна верифікація базується на математичному (логічному) моделюванню та вимогах до нього [16]. Ідея така ж, як і при використанні моделей в інших областях знань:

– створена модель – ідеалізований опис досліджуваного об'єкта або явища;

– модель досліджується з використанням математичних методів;

– результати досліджень передаються реальному об'єкту або явищу.

Загальний алгоритм формальної верифікації [17] наступний:

– створюється формальна модель програми;

– створюється формальна модель вимог;

– формально верифікується відповідність моделі програми вимогам моделі;

– на підставі результатів тесту, робиться висновок, чи відповідає реальна програма реальним вимогам (іншими словами, що в програмі є або немає помилок).

Для представлення програмних моделей та моделей вимог використовуються мови формальної специфікації програм (мови моделювання) та мови формальної специфікації вимог відповідно.

Формальна верифікація КФС є нетривіальним завданням через їх стохастичну природу, нелінійність, тимчасовість. Наразі відомі декілька основних підходів до формальної верифікації КФС: симуляція, доведення теорем, символічне виконання та перевірка моделі.

Симуляція, або тестування КФС широко використовується при проектуванні бездротових мереж для верифікації взаємодії фізичного світу та мережевих компонентів. Так, представлений симулятор TrueTime [18] для оцінювання ефекту на продуктивність мережі безперервних систем управління. WCPS [19] симулює ефекти затримки мережі і втрати даних при роботі з управління цивільною інфраструктурою. PiccSIM [20] та NCSWT [21] інтегрують системи керування з бездротовими мережами для досягнення більш реалістичних моделей. Також багато робот присвячено симуляції постійного трафіку в КФС [22, 23, 24].

Метод доведення теорем використовує математичну аргументацію для верифікації коректної роботи КФС. Інструмент доведення теорем KeYmaera [25] використовується разом з символічними обчисленнями для верифікації гібридних систем. Для верифікації КФС використовується STHA [26], що враховує просторову та часову природу КФС. Крім цього достатньо велика кількість робіт присвячена розробці автоматичних або напівавтоматичних інструментів для виконання верифікації методом доведення теорем [27, 28].

Метод символічного виконання виконується у якості аналізу КФС з метою автоматичної генерації тест-кейсів, що можуть виявити помилки у роботі КФС. При цьому змінні системи представлені у якості символів, а етапи роботи КФС – виразів над цими

DOI 10.15588/1607-3274-2020-3-5

символами. Таким чином, метод символічного виконання може автоматично створювати тест-кейси для програмного забезпечення КФС, які після цього можна симулювати на фізичній складовій системи для аналізу роботи всієї КФС при різних вхідних даних [29, 30]. Ще одним прикладом використання даного методу є SPF [31], що використовується для верифікації коду КФС, написаному для JAVA компілятора.

Перевірка моделі це ще один метод формальної верифікації, що можна використовувати для КФС [32]. Цей метод виконує перевірку математичної моделі на відповідність вимогам до системи, вираженим зазвичай у темпоральних логіках [33, 34].

При виконанні аналізу існуючої літератури можна виявити певні переваги та недоліки у всіх існуючих методах формальної верифікації КФС.

Так, метод симуляції (тестування) це найбільш швидкий та простий спосіб виявити помилки у системі. Він легко піддається автоматизації та може використовуватися як для кібернетичної, так і для фізичної складової КФС. Однак, він не враховує у верифікації виникнення подій та їхній таймінг під час роботи КФС. Такі події можуть навіть змінювати параметри системи, тим самим впливаючи на результати верифікації. А неможливість уточнення таймінгу призводить до погіршення якості та виникненню похибок при перевірці. А найбільшим недоліком є те, що хоча тестування і є надійним інструментом для виявлення помилок, воно не здатне довести та показати повну їх відсутність [35].

Метод доведення теорем є потужним інструментом для верифікації складних систем. Він зазвичай спрямований на доказ повної функціональної коректності системи. Однак він, у свою чергу, не дуже ефективний при спробах автоматичної верифікації, оскільки отримані докази можуть бути дуже довгими та важкими для розуміння навіть експертів у предметній області [36]. Крім цього, більшість робіт присвячених цьому методу використовують інваріантні параметри системи, роблячи неможливим верифікації критичних подій [37].

Метод символічного виконання використовує алгоритми для верифікації. Він відповідає семантиці традиційних мов програмування та зазвичай основну увагу приділяє ефективності системи. Добре автоматизується, однак, як і метод доведення теорем має проблеми при верифікації критичних подій. Крім цього основний фокус займає код розробленої програми, а не модель всієї системи.

Метод перевірки моделі вирішує певні проблеми перелічених методів, адже описує всі стани системи, з урахуванням подій та критичних випадків. Крім того існує можливість автоматизувати даний метод, а отже прискорити та полегшити верифікацію без залучення експерта. Ще однією важливою перевагою є можливість верифікації паралельних систем. Серед недоліків методу перевірки моделі можна навести те, що верифікація складних систем, зокрема КФС, призводить до комбінаторного зростання кількості станів системи для перевірки, що, у свою чергу, призводить до зростання часу та необхідної обчислювальної пам'яті в порівнянні зі згаданими вище методами[38].

3 МАТЕРІАЛИ І МЕТОДИ

Перевірка моделі у загальному вигляді складається з наступних складових [39]:

– модель – необхідно використовувати таку модель, що зможе адекватно відобразити всі стани та переходи як фізичної, так і кіберфізичної частини КФС (а також комунікації між ними);

– специфікації – зазвичай використовуються формальні логіки, які дозволяють природньо описати вимоги до коректної поведінки КФС;

– алгоритми – у їх якості використовуються процедури прийняття рішень про коректність системи (чи відповідає модель вказаній формулі темпоральної логіки).

Зазвичай, перевірка моделей дозволяє не тільки оцінити коректність КФС, а й навести контр-приклади у разі висновку про некоректну поведінку системи.

Процес верифікації методом перевірки моделей починається зі створення моделі системи. У якості моделі станів та переходів найчастіше використовується модель Кріпке. Далі специфікації системи виражаються у формулах формальних логік, зазвичай використовуються темпоральні логіки. Після отримання φ, процедура прийняття рішення оцінює чи виконується вираз $M \models \varphi$, тобто чи є M моделлю формули φ.

Моделі Кріпке були введені у кінці 1950-х років Саулом Аароном Кріпке для логічного і філософського використання [40]. У загальному вигляді модель Кріпке є системою можливих світів і переходів між ними: кожен світ є статичним і інтерпретується традиційним чином. Її діаграма наведена на рис. 1. Модель Кріпке це кортеж наступного виду:

$$\langle S, S_0, R, L \rangle; \quad (1)$$

$$S_0 \subseteq S; \quad (2)$$

$$R \subseteq S \times S; \quad (3)$$

$$L: S \to 2^{AP}. \quad (4)$$

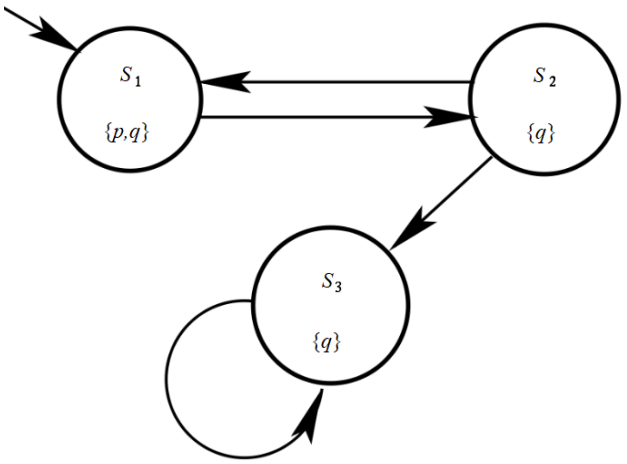

Рисунок 1 – Діаграма моделі Кріпке

DOI 10.15588/1607-3274-2020-3-5

Модель Кріпке – це анотований граф переходів з кінцевим станом. Динамічна поведінка системи у моделі Кріпке представлена шляхом через граф. Шлях $\pi = S_0, S_1, S_2, \ldots$ це кінцева або нескінченна послідовність станів системи, при умові що $(S_i, S_{i+1}) \in R$ для всіх $i \geq 0$. Таким чином, верифікація системи відбувається методом перевірки шляхів моделі Кріпке на відповідність специфікаціям системи.

Робота КФС описується в термінах послідовностей подій, розподілених у часі. Як зазначалося вище, найчастіше для специфікації вимог до КФС використовуються темпоральні логіки. Темпоральні логіки є формальними мовами, що дозволяють визначити взаємозв'язки подій у часі: причинно-наслідкові зв'язки, обмеження відносної послідовності, величини затримок між подіями тощо. Найбільш популярними для методу перевірки моделей є ЛЛЧ [41] та ЛДО [42].

ЛДО*[43] це множина, що об'єднує ЛЛЧ та ЛДО. Вона поєднує в собі квантори шляху та темпоральні оператори. Квантори шляху інтерпретуються по станам системи, в той час як темпоральні оператори – по шляхам в системі. ЛДО* містить два квантори шляху: перший, що позначає «для кожного нескінченного шляху з цього стану» та другий – «існує нескінченний шлях з цього стану».

Темпоральні оператори ЛДО* можна інтерпретувати наступним чином.

1. Формула φ є істиною в наступний момент часу – $X\varphi$.

2. Формула ψ є істинною зараз або обов'язково стане істинною в майбутньому, але до цього моменту (не включно) φ повинна бути істиною – $\varphi U \psi$.

3. Формула φ є істиною зараз або стане істиною у майбутньому – true$U\varphi$, що також може бути відображене як $F\varphi$.

Формула ¬φ є невірною зараз і ніколи не стане істиною у майбутньому (завжди, відтепер, φ є істиною) – $\neg F \neg \varphi$ яка також представлена як $G\varphi$.

ЛДО це синтаксичний фрагмент ЛДО*, в якому після кожного квантору шляху слідує темпоральний оператор. Формулу даної логіки можна позначити так:

$$\varphi := p|(\varphi \vee \varphi)|(\varphi \wedge \varphi)|\neg\varphi|AX\varphi|EX\varphi \\ |AF\varphi|EF\varphi|AG\varphi|EG\varphi|A\varphi U\varphi|E\varphi U\varphi \quad (5)$$

ЛЛЧ це другий фрагмент ЛДО*, який не містить кванторів шляху, крім передового A. Формула ЛЛЧ має наступний вигляд:

$$\varphi := p|(\varphi \vee \varphi)|(\varphi \wedge \varphi)|\neg\varphi|X\varphi|F\varphi|G\varphi|\varphi U\varphi . \quad (6)$$

В цілому, клас складності ЛДО кращий ніж у ЛЛЧ, однак це компенсується тим, що у ЛЛЧ набагато легше та швидше виводити контр-приклади, тоді як у ЛДО вони можуть досягати всіх станів верифікуємої моделі. Зазвичай на практиці розмір простору станів є більш критичним ніж довжина темпоральної специфікації. Тому ЛЛЧ зазвичай ефективніша.

Підсумовуючи, можна зробити висновок що перевірка моделі є найбільш перспективним методом верифікації КФС з наразі існуючих. Однак, і він має проблеми та перспективи для розвитку. Це пов'язано передусім з тим, що інструментарію анотованого графу переходів може не вистачати для сучасних надскладних КФС. Крім цього може виникати так звана «проблема вибуху станів» – комбінаторного збільшення кількості станів системи, що ускладнює, або навіть унеможливлює створення моделі Кріпке для всіх станів КФС. Тому пошук альтернатив та розробка нових підходів для моделювання КФС є надзвичайно актуальною задачею.

4 ЕКСПЕРИМЕНТИ

В якості експериментальної частини роботи було вирішено побудувати моделі Кріпке для системи керування електронезалежною електромобільною зарядною станцією [44]. Така станція являє собою КФС і складається з сонячної панелі, вітрогенератору, джерела живлення, навантаження, акумулятору та контролера заряду. Схема підключення компонентів зображена на рис. 2.

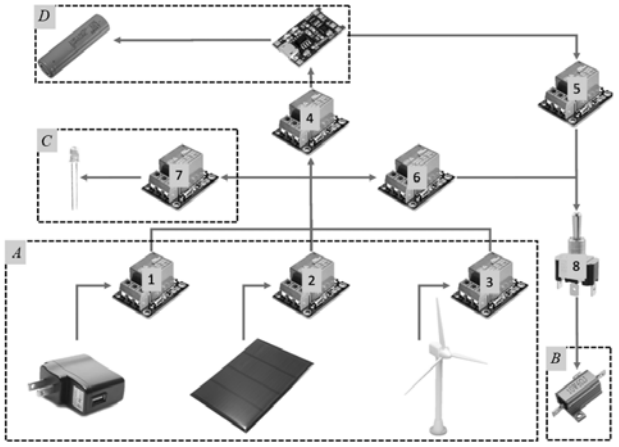

Рисунок 2 – Схема електронезалежної станції альтернативної енергетики

Модель електронезалежної станції альтернативної енергетики дозволяє обирати джерело живлення (будь-яке окремо або 2–3 джерела одночасно), контролює процес заряду акумулятора (захист від перезаряду або перезаряду), дозволяє «продавати» залишкову енергію та вмикати/вимикати навантаження.

Джерело живлення помічено літерою A, навантаження – B, продаж надлишку зображено у вигляді світлодіоду – C, акумулятор та контролер заряду – D.

Варіанти роботи модулю заряду акумулятора [45]:

– заряд меньше 20%: відкрити реле 5 та 7, закрити 1 та 6;

– заряд від 20% до 90%: відкрити реле 1, 6 та 7, закрити 5;

– заряд більше 90%: закрити реле 5 та 7, відкрити 4 та 6.

Алгоритм роботи [46] даної моделі наведений на рис. 3.

DOI 10.15588/1607-3274-2020-3-5

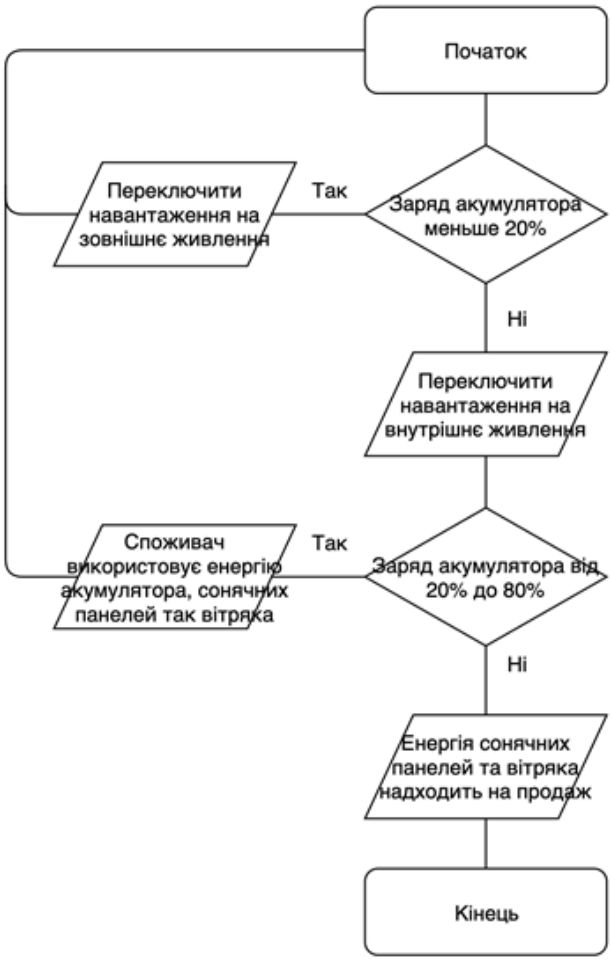

Рисунок 3 – Робота енергетичної системи

Готова модель цифрового двійника зображена на рис. 4.

Для побудови моделі Кріпке необхідно визначити СА системи [47]. Після цього, з урахуванням властивостей системи, можна побудувати модель Кріпке, базуючись на СА.

СА є обчислювальними моделями, які створюють регулярні мови та можуть бути використані для моделювання послідовної логіки. Існують два типи СА: ДСА і НСА. ДСА описується п'ятиелементним кортежем:

$$\langle Q,\Sigma,\delta,q_0,G\rangle. \quad (7)$$

Для кожного вхідного символу має бути одна функція переходу Σ з кожного стану.

Подібно до ДСА, НСА описується вищезгаданим п'ятиелементним кортежем. На відміну від ДСА, НСА не повинні мати функцій переходу для кожного символу в Σ, і можуть мати декілька функцій переходу в одному і тому ж стані для одного і того ж символу. Крім того, НСА можуть використовувати нульові переходи. Нульові переходи дозволяють переходити з одного стану в інший без необхідності зчитувати символ.

Рисунок 4 – Модель електронезалежної станції альтернативної енергетики

НСА є настільки ж потужними, як і ДСА. Проте ДСА потребуватиме набагато більше станів і переходів, ніж НСА для вирішення тієї ж самої проблеми. Перетворення з ДСА в НСА і навпаки можливе, що робить їх еквівалентними.

5 РЕЗУЛЬТАТИ

В результаті експерименту було побудовано дві моделі Кріпке для роботи електронезалежної станції альтернативної енергетики, які наведені на рис. 5 та рис. 6.

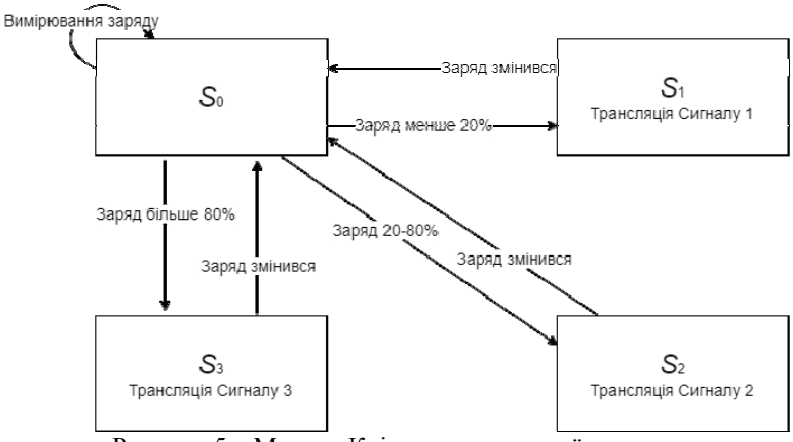

Рисунок 5 – Модель Кріпке енергетичної системи

DOI 10.15588/1607-3274-2020-3-5

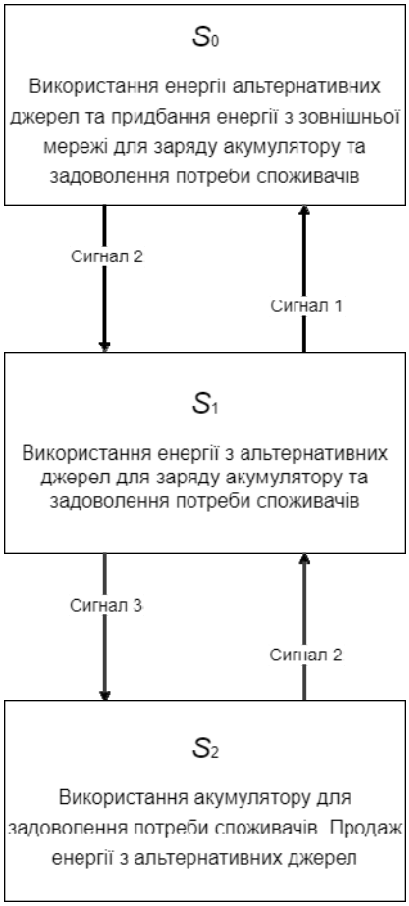

Рисунок 6 – Модель Кріпке електронезалежної станції альтернативної енергетики

У кожному зі станів моделі Кріпке, зображеної на рис. 6 кожне реле може бути у двох станах – відкрите (1) та закрите (0).

Результати моделювання електронезалежної станції альтернативної енергетики наведені у табл. 1. На ній неведені відомості про стани восьми реле, зображених на рис. 2.

Таблиця 1 – Результати моделювання електронезалежної станції альтернативної енергетики

Номер реле	Результати моделювання		
	У стані S_0	У стані S_1	У стані S_2
1	1	0	0
2	1	1	1
3	1	1	1
4	1	1	0
5	0	0	1
6	1	1	0
7	0	0	1
8	1	1	1

Як можна бачити з результатів, розроблена модель дозволяє керувати станами реле електронезалежної станції альтернативної енергетики для досягнення оптимальної ефективності системи.

Виконане моделювання КФС дозволяє описати всі стани системи, необхідні для виконання формальної верифікації.

6 ОБГОВОРЕННЯ

Був проведений аналіз підходів до верифікації КФС. Було розглянуто методологію найперспективнішого підходу та виявлено певні недоліки. В якості експериментальної частини дослідження були спроектовані моделі Кріпке електронезалежної станції альтернативної енергетики.

Перша модель реалізує алгоритм роботи моделі енергетичної системи. Так, залежно від рівня заряду акумулятору, модель може транслювати три типи сигналу. Друга модель, у свою чергу, використовує такі сигнали для переведення моделі електронезалежної станції альтернативної енергетики у відповідні стани. Переходи між станами відбуваються завдяки перехідним функціям, які також відображені на моделі. Крім цього, кожен стан моделі має атомарні пропозиції, які істинні в цьому стані.

Порівнюючи метод перевірки моделі з іншими методами формальної верифікації, можна зробити висновок що він має перевагу, адже описує всі стани системи, з урахуванням подій та критичних випадків. Окрім цього, метод перевірки моделі можна використовувати не тільки для повністю розробленої системи, а й на етапах проектування та розробки, тобто знаходити помилки раніше. Таким чином, очевидним є факт того, що даний метод верифікації більше підходить для роботи з паралельними системами, якими і є КФС, а отже верифікація таких систем повинна виконуватися саме методом перевірки моделі.

ВИСНОВКИ

В роботі дано характеристику КФС, проведено аналіз методів верифікації таких систем. За результатами досліджень зроблено висновок про найбільшу перспективність методу перевірки моделі, для якого розглянуто основну методологію. Моделі Кріпке та темпоральні логіки охарактеризовані як основні елементи методу перевірки моделі.

Проаналізовано недоліки стандартної методології, які більш за все стосуються етапу моделювання КФС. Доведено можливість використання СА, а саме моделей Кріпке, для моделювання розробленої КФС. На прикладі побудованих моделей Кріпке для станції альтернативної енергетики показано можливість використання даних підходів до моделювання фізичних та кібернетичних елементів реальної КФС.

Зазначено, що використання моделей Кріпке дозволить формалізувати модель цифрового двійника електронезалежної станції альтернативної енергетики. Що, у свою чергу, необхідно для виконання верифікації системи. Верифікація електронезалежної станції альтернативної енергетики формальними методами дозволить імплементувати такі підходи у реальному світі.

Наукова новизна роботи полягає в тому що було розроблено моделі КФС, які, на відміну від існуючих, засновані на моделях Кріпке, що дозволяє зробити детальний опис усіх станів системи, що, у свою чергу,

DOI 10.15588/1607-3274-2020-3-5

є важливим кроком для виконання верифікації такої системи.

Практичною цінністю роботи є розроблені моделі електронезалежної станції альтернативної енергетики, що дозволять автоматизувати процес зарядки електричних транспортних засобів.

Були реалізовані цифрові двійники, які дозволяють моделювати процеси електронезалежної станції альтернативної енергетики. Розроблені двійники використовуються при вивчені дисциплін при підготовки бакалаврів та магістрів спеціальності 121 комп'ютерні науки.

Наступним етапом роботи пропонується продовжувати дослідження альтернативних методів моделювання КФС, виконання формальної верифікації методом перевірки моделі та використання отриманих результатів для станції альтернативної енергетики з метою оптимізації споживачів та джерел енергії.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Workshop on cyber-physical systems [Electronic resource] – Access mode: http://varma.ece.cmu.edu/CPS
2. Johnson T. T. Parametrized verification of distributed cyber-physical systems: an aircraft landing protocol case study / T. T. Johnson, S. Mitra // Cyber-Physical Systems : Third international conference, Beijing, 17–19 April 2012 : proceedings. – Beijing : IEEE/ACM, 2012. – P. 161–170. DOI: 10.1109/iccps.2012.24
3. From verification to implementation: a model translation tool and a pacemaker case study / [M. Pajic, Z. Jiang, I. Lee et al.] // IEEE 18th Real Time and Embedded Technology and Applications Symposium, Beijing. – 2012. – P. 173–184. DOI:10.1109/rtas.2012.25
4. Cyber-physical security of a smart grid infrastructure / [Y. Mo, T. H. Kim, K. Brancik et al.] // Proceedings of the IEEE. – 2012. – Vol. 100, № 1. – P. 195–209. DOI: 10.1109/jproc.2011.2161428
5. Cyber-physical systems: the next computing revolution / [R. Rajkumar, I. Lee, L. Sha et al.] // Design Automation : 47th international conference, Anaheim, 13–18 June 2010 : proceedings. – Anaheim : IEEE, 2010. – P. 731–736. DOI:10.1145/1837274.1837461
6. Cyber-Physical Systems [Electronic resource] – Access mode: https://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm
7. Miclea L. About dependability in cyber-physical systems / L. Miclea, T. Sanislav // 9th East-West Design & Test Symposium (EWDTS), Sevastopol. – 2011. – P. 17–21. DOI:10.1109/EWDTS.2011.6116428
8. McConnell S. Code complete / S. McConnell. – Redmond : Microsoft Press, 2009. – 960 p.
9. Lloyd S. Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos / S. Lloyd. – New York : Knopf, 2006. – 240 p.
10. Korotunov S. Cyber-physical systems architectures and modeling methods analysis for smart grids / S. Korotunov, G. Tabunshchyk, C. Wolff // Computer Sciences and Information Technologies : 13th International Scientific and Technical Conference, Lviv, 11–14 Sept. 2018 : proceedings. – Lviv : IEEE, 2018. – P. 181–186. DOI:10.1109/STC-CSIT.2018.8526726
11. Gupta S. Safety / S. Gupta, T. Mukherjee, K. Venkatasubramanian // Body area networks: safety, security, and sustainability. – Cambridge : Cambridge University Press, 2013. – Section 4. – 36 p. DOI:10.1017/CBO9781139108126.006
12. Kernbach S. Generation of desired emergent behavior in swarm of micro-robots / S. Kernbach, O. Kornienko, P. Levi // Artificial Intelligence : 16th Eureopean Conference, Valencia, 22–27 August 2004 : proceedings. – Valencia : IOS Press, 2004. – Vol. 110. – P. 239–243.
13. Cyber-Physical Systems. Foundations, Principles and Applications / [H. Song, D. B. Rawat, S. Jeschke et al.]. – Cambridge : Academic Press, 2017. – 514 p.
14. Analysis of the verification approaches for the cyberphysical systems / [S. Korotunov, G. Tabunshchyk, K. Henke et al.] // Third International Workshop on Computer Modeling and Intelligent Systems (CMIS), Zaporizhzhia. – 2019. – P. 950–961.
15. A framework for the design and verification of software measurement methods / [N. Habra, A. Abran, M. Lopez et al.] // Journal of Systems and Software. – 2008. – Vol. 81, № 5. – P. 633–648. DOI:10.1016/j.jss.2007.07.038.
16. Субботин С. А. Метод синтеза диагностических моделей на основе радиально-базисных нейронных сетей с поддержкой обобщающих свойств / С. А. Субботин // Радіоелектроніка, інформатика, управління. – 2016. – № 2 (37). – С. 64–69. DOI:10.15588/1607-3274-2016-2-8
17. Tuch H. OS Verification – Now! [Electronic resource] / H. Tuch. – Access mode: https://www.usenix.org/legacy/event/hotos05/final_papers_backup/tuch/tuch_html/index.html
18. Cervin A. How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime / A. Cervin, D. Henriksson, B. Lincoln // IEEE Control Systems Magazine. – 2003. – Vol. 23, № 3. – P. 16–30. DOI:10.1109/MCS.2003.1200240
19. Realistic case studies of wireless structural control / [B. Li, Z. Sun, K. Mechitov et al.] // Cyber-Physical Systems : 4th international conference, Philadelphia, 8–11 April 2013 : proceedings. – Philadelphia : IEEE, 2013. – P. 179–188. DOI:10.1109/ICCPS.2013.6604012
20. PiccSIM Toolchain – design, simulation and automatic implementation of wireless networked control systems / [T. Kohtamaki, M. Pohjola, J. Brand et al.] // Networking, Sensing and Control : International conference, Okayama, 26–29 March 2009 : proceedings. – Okayama : IEEE, 2009. – P. 49–54. DOI:10.1109/ICNSC.2009.4919244
21. NCSWT: An integrated modeling and simulation tool for networked control systems / [E. Eyisi, J. Bai, D. Riley et al.] // Simulation Modelling Practice and Theory. – 2012. – Vol. 27. – P. 90–111. DOI:10.1016/j.simpat.2012.05.004
22. Hybrid packet/fluid flow network simulation / [C. Kiddle, R. Simmonds, C. Williamson et al.] // Seventeenth Workshop on Parallel and Distributed Simulation (PADS), San Diego. – 2003. – P. 143–152. DOI:10.1109/PADS.2003.1207430
23. Liu J. Parallel simulation of hybrid network traffic models / J. Liu // 21st International Workshop on Principles of Advanced and Distributed Simulation. – 2007. – P. 141–151. DOI:10.1109/PADS.2007.26
24. Melamed B. HNS: A streamlined hybrid network simulator / B. Melamed, S. Pan, Y. Wardi // ACM Transactions on Modeling and Computer Simulation. – 2004. – Vol. 14, № 3. – P. 251–277. DOI:10.1145/1010621.1010623
25. Platzer A. KeYmaera: a hybrid theorem prover for hybrid systems (system description) / A. Platzer, J. D. Quesel // Automated Reasoning : 4th International Joint Conference, Berlin Heidelberg, 12–15 August 2008 : proceedings. –

DOI 10.15588/1607-3274-2020-3-5

Berlin : Springer-Verlag, 2008. – P. 171–178. DOI:10.1007/978-3-540-71070-7_15
26. Banerjee A. Spatio-temporal hybrid automata for safe cyber-physical systems: A medical case study / A. Banerjee, K. S. Gupta // Cyber-Physical Systems : 4th international conference, Philadelphia, 8–11 April 2013 : proceedings. – Philadelphia : ACM/IEEE, 2013. – P. 71–80. DOI:10.1145/2502524.2502535
27. Formal software verification: model checking and theorem proving : Embedded Systems Laboratory Technical Report : ESL-TIK-00214 / Massachusetts Institute of Technology ; M. Ouimet. – Cambridge, 2007. – 12 p.
28. Bagade P. Safety assurance of medical cyber-physical systems using hybrid automata: a case study on analgesic infusion pump / P. Bagade, A. Banerjee, S. K. Gupta // Medical CPS Workshop. – 2013. – P. 111–118. DOI:10.1.1.294.6604
29. Scalable symbolic execution of distributed systems / [R. Sasnauskas, O. S. Dustmann, B. L. Kaminski et al.] // Distributed Computing Systems : 31st International Conference, Minneapolis, 20–24 June 2011: proceedings. – Minneapolis : IEEE, 2011. – P. 333–342. DOI:10.1109/ICDCS.2011.28
30. CLSE: closed-loop symbolic execution / [R. Majumdar, I. Saha, K. C. Shashidhar et al.] // NASA Formal Methods Symposium. – 2012. – P. 356–370. DOI:10.1007/978-3-642-28891-3_33
31. Pasareanu C. S. Symbolic PathFinder: symbolic execution of Java bytecode / C. S. Pasareanu, N. Rungta // Automated Software Engineering : 25th international conference, Antwerp, 20–24 September 2010 : proceedings.– Antwerp : IEEE/ACM, 2010. – P. 179–180. DOI:10.1145/1858996.1859035
32. Clarke E. M. Statistical model checking for cyber-physical systems / E. M. Clarke, P. Zuliani // Automated Technology for Verification and Analysis, 9th International Symposium, Taipei. – 2011. – Vol. 6996. – P. 1–12. DOI:10.1007/978-3-642-24372-1_1
33. Feedback control for statistical model checking of cyber-physical systems / [K. Kalajdzic, C. Jegourel, A. Lukina et al.] // International Symposium on Leveraging Applications of Formal Methods. – 2016. – Vol. 9952. – P. 46–61. DOI:10.1007/978-3-319-47166-2_4
34. Clarke E. M. Automatic verification of finite state concurrent systems using temporal logic specifications: a practical approach / E. M. Clarke, E. A. Emerson, A. P. Sistla // Proceedings of the 10th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. – 1983. – P. 117–126. DOI:10.1145/567067.567080
35. Dijkstra E. W. The humble programmer / E. W. Dijkstra // Communications of the ACM. – 1972. – Vol. 15, № 10. – P. 859–866. DOI:10.1145/355604.361591
36. Субботин С. А. Синтез нейро-нечетких сетей с ранжированием и специфическим кодированием признаков для диагностики и автоматической классификации по прецедентам / С. А. Субботин // Радіоелектроніка, інформатика, управління. – 2016. – № 1 (36). – С. 50–57. DOI:10.15588/1607-3274-2016-1-6
37. Abraham-Mumm E. Verification of hybrid systems: formalization and proof rules in PVS / E. Abraham-Mumm, U. Hannemann, M. Steffen // Engineering of Complex Computer Systems : 7th international conference, Skovde, 11–13 June 2001 : proceedings. Skovde : IEEE, 2001. – P. 48–57. DOI:10.1109/ICECCS.2001.930163
38. Handbook of Model Checking / [E. M. Clarke, T. A. Henzinger, H. Veith et al.]. – Cham : Springer International Publishing, 2018. – 1210 p.
39. Clarke E. M. Model Checking / E. M. Clarke, O. Grumberg, B. Peled. – Cambridge : MIT Press, 2001. – 314 p.
40. Gabbay D. Semantical Considerations for Modal Logics by Saul A. Kripke / D. Gabbay // The Journal of Symbolic Logic. – 1969. – Vol. 34, № 3. – P. 501. DOI:10.2307/2270922
41. Pnueli A. The temporal logic of programs / A. Pnueli // 18th Annual Symposium on Foundations of Computer Science (SFCS), Providence. – 1977. – P. 46–57. DOI:10.1109/SFCS.1977.32
42. Clarke E. M. Design and synthesis of synchronization skeletons using branching time temporal logic / E. M. Clarke, E. A. Emerson // Workshop on Logic of Programs. – 1982. – P. 52–71.
43. Modeling and simulation of the services for vehicle charging infrastructure interaction / [P. Arras, G. Tabunshchyk, V. Okhmak et al.] // Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications : 10th international conference, Metz, 18–21 September 2019 : proceedings. – Metz : IEEE, 2019. – P. 330–333. DOI:10.1109/IDAACS.2019.8924449
44. Cost optimization simulation for electric vehicle charging infrastructure / [G. Tabunshchyk, P. Arras, S. Korotunov et al.] // IEEE European Technology & Engineering Management Summit (ETEMS), Dortmund. – 2020. – P. 76–88.
45. McNaughton R. Elementary computability, formal languages, and automata / R. McNaughton. – Lawrence : Z B Publishing Industries, 1993. – 400 p.
46. Arras P. Type or paste your text here to design optimization techniques in mechanical design and education of engineers convert case / P. Arras, G. Tabunshchyk // Advances in Design, Simulation and Manufacturing. – 2020. – P. 13–22. DOI:10.1007/978-3-030-22365-6_2
47. Korotunov S. Genetic algorithms as an optimization approach for managing electric vehicles charging in the Smart Grid / S. Korotunov, G. Tabunshchyk, V. Okhmak // Computer Modeling and Intelligent Systems : 3rd international workshop, Zaporizhzhia, 27–01 May 2020 : proceedings. – Zaporizhzhia : CEUR WS, 2020. – P. 184–198.

Received 23.06.2020.
Accepted 28.08.2020.

УДК 004.9

АНАЛИЗ ПОДХОДОВ К МОДЕЛИРОВАНИЮ И ВЕРИФИКАЦИИ КИБЕРФИЗИЧНИХ СИСТЕМ

Коротунов С. Ю. – аспирант кафедры программных средств Национального университета «Запорожская политехника», Запорожье, Украина.

Табунщик Г. В. – канд. техн. наук, доцент, профессор кафедры программных средств Национального университета «Запорожская политехника», Запорожье, Украина.

DOI 10.15588/1607-3274-2020-3-5

АННОТАЦИЯ

Актуальность. Современные тенденции в производительности и сложности требований к использованию систем требуют принципиально новых подходов к проектированию, в которых кибернетические и физические компоненты интегрируются на разных этапах. Киберфизические системы окружают человека почти во всех сферах существования, начиная с домов и транспорта и заканчивая медицинскими аппаратами и межрегиональными электросетями. Поэтому верификация и проверка работы таких систем является сегодня актуальной задачей. В таких системах программное обеспечение и физические подсистемы работают в разных временных и пространственных измерениях, взаимодействуют разными способами. Рассмотрены основные подходы к верификации киберфизических систем. Объектом исследований является процесс верификации киберфизических систем, предметом – методы верификации киберфизических систем, модели и логики, которые используются при формальной верификации.

Цель. Цель работы заключается в проведении анализа подходов к верификации киберфизических систем, с детализацией отдельных этапов, таких как выбор моделей, инструментов верификации, и, собственно, методов верификации.

Метод. Основными методами, которые изложены в работе, являются методы формальной верификации киберфизических систем, а именно – симуляция, доказательства теорем, символическое выполнение и проверка модели. Подробно рассмотрена методология проверки модели – модель Крипке и темпоральные логики: логика деревьев вычислений и логика линейного времени. Также проведено моделирование с использованием конечных автоматов.

Результаты. Выполнено моделирование киберфизической системы в виде создания модели Крипке, что позволило описать все состояния системы, необходимые для выполнения формальной верификации.

Выводы. Проведено исследование характеристик киберфизических систем, выполнен анализ методов верификации таких систем. Указаны недостатки стандартной методологии, которые более всего касаются этапа моделирования киберфизических систем. Доказано наибольшая перспективность метода проверки модели, для которого рассмотрена основная методология. Дана характеристика моделям Крипке и темпоральным логикам как основным элементам метода проверки модели. Показана возможность использования конечных автоматов, а именно моделей Крипке, для моделирования элементов киберфизической системы. Научная новизна работы заключается в том, что были разработаны модели киберфизических систем, которые, в отличие от существующих, основанные на моделях Крипке, что позволяет сделать детальное описание всех состояний системы, что, в свою очередь, является важным шагом для выполнения верификации такой системы. Практической ценностью работы являются разработанные модели електронезависимой станции альтернативной энергетики, которые позволят автоматизировать процесс зарядки электрических транспортных средств. Были реализованы цифровые двойники, которые позволяют моделировать процессы електронезависимой станции альтернативной энергетики. Разработанные двойники используются при изучении дисциплин при подготовке бакалавров и магистров специальности 121 компьютерные науки.

КЛЮЧЕВЫЕ СЛОВА: верификация, киберфизическая система, модель Крипке, конечный автомат, темпоральная логика, проверка модели, симуляция.

UDC 004.9

ANALYSIS OF APPROACHES TO THE SIMULATION AND VERIFICATION OF CYBER-PHYSICAL SYSTEMS

Korotunov S. U. – Postgraduate student at the Department of Software of National University “Zaporizhia Polytechnic”, Zaporizhzhia, Ukraine.

Tabunshchyk G. V. – PhD, Professor of the Department of Software of National University “Zaporizhia Polytechnic”, Zaporizhzhia, Ukraine.

ABSTRACT

Context. Current trends in the performance and complexity of system requirements require fundamentally new approaches to design, in which cybernetic and physical components are integrated at different stages. Cyber-physical systems are systems that provide close interaction between physical and cybernetic components, integration of computing, physical processes and networks. In such systems, software and physical subsystems operate in different temporal and spatial dimensions, interacting in different ways. Cyber-physical systems surround humans in almost every area of existence, from housing and transportation to medical devices and interregional power grids. Therefore, verification and validation of such systems is an urgent task today. Approaches to verification of cyber-physical systems are considered. The object of research is the process of verification of cyber-physical systems, the subject is the methods of verification of cyber-physical systems, models and logic used in formal verification.

Objective. The purpose of the work is to analyze approaches to the verification of cyber-physical systems, detailing the individual steps, such as the selection of models, verification tools, and, in fact, verification methods.

Method. The main methods outlined in the paper are methods of formal verification of cyber-physical systems, namely simulation, theorem proving, symbolic execution, and model checking. In addition, the methodology of the model checking method – the Kripke structure and temporal logics: logic of computational trees and linear time logic is discussed in detail. Modeling using finite state machines is also performed.

Results. The paper deals with modeling of the cyber-physical system in the form of creation of the Kripke structure that allowed to describe all states of the system necessary for executing of formal verification.

Conclusions. The paper describes the characteristics of cyber-physical systems, analyzes the methods of verification of such systems. After analysis the conclusion is made about the most promising method of model verification, for which the basic methodology is considered. Characteristics of Kripke structure and temporal logics are described as the main elements of the model checking method. Following the review, the shortcomings of the standard methodology most relevant to the modeling stage of cyber-physical systems are concluded. The possibility of using finite state machines, namely Kripke structures, for modeling elements of a cyber-

DOI 10.15588/1607-3274-2020-3-5

physical system is shown. The scientific novelty of the work is that models of cyber-physical systems have been developed, which, unlike existing ones, are based on Kripke structures, which allow to make a detailed description of all states of the system, which, in turn, is an important step to verify such a system. The practical value of the work is the developed models of the independent power station of alternative energy, which will automate the process of charging electric vehicles. Digital duplicates have been implemented, which allow modeling the processes of an independent energy station of alternative energy. The developed duplicates are used in the study of disciplines in the preparation of bachelors and masters in 121 computer science.

KEYWORDS: verification, cyber-physical system, Kripke structure, finite state machine, temporal logic, model verification, simulation.

REFERENCES

1. Workshop on cyber-physical systems [Electronic resource] – Access mode: http://varma.ece.cmu.edu/CPS
2. Johnson T. T., Mitra S. Parametrized verification of distributed cyber-physical systems: an aircraft landing protocol case study, *Cyber-Physical Systems, Third international conference, Beijing, 17–19 April 2012, proceedings*. Beijing, IEEE/ACM, 2012, P. 161–170. DOI: 10.1109/iccps.2012.24
3. Pajic M., Jiang Z., Lee I. et al. From verification to implementation: a model translation tool and a pacemaker case study, *IEEE 18th Real Time and Embedded Technology and Applications Symposium.* Beijing, 2012, pp. 173–184. DOI: 10.1109/rtas.2012.25
4. Mo Y., Kim T. H., Brancik K. et al. Cyber-physical security of a smart grid infrastructure, *Proceedings of the IEEE*, 2012, Vol. 100, № 1, pp. 195–209. DOI: 10.1109/jproc.2011.2161428
5. Rajkumar R., Lee I., Sha L. et al. Cyber-physical systems: the next computing revolution, *Design Automation, 47th international conference, Anaheim, 13–18 June 2010, proceedings.* Anaheim, IEEE, 2010, pp. 731–736. DOI: 10.1145/1837274.1837461
6. Cyber-Physical Systems [Electronic resource]. Access mode: https://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm
7. Miclea L., Sanislav T. About dependability in cyber-physical systems, *9th East-West Design & Test Symposium (EWDTS).* Sevastopol, 2011, pp. 17–21. DOI: 10.1109/EWDTS.2011.6116428
8. McConnell S. Code complete. Redmond, Microsoft Press, 2009, 960 p.
9. Lloyd S. Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos. New York, Knopf, 2006, 240 p.
10. Korotunov S., Tabunshchyk G., Wolff C. Cyber-physical systems architectures and modeling methods analysis for smart grids, *Computer Sciences and Information Technologies , 13th International Scientific and Technical Conference.* Lviv, 11–14 Sept. 2018, proceedings. Lviv, IEEE, 2018, pp. 181–186. DOI: 10.1109/STC-CSIT.2018.8526726
11. Gupta S., Mukherjee T., Venkatasubramanian K. Safety Body area networks: safety, security, and sustainability. Cambridge, Cambridge University Press, 2013, Section 4, 36 p. DOI: 10.1017/CBO9781139108126.006
12. Kernbach S., Kornienko O., Levi P. Generation of desired emergent behavior in swarm of micro-robots, *Artificial Intelligence, 16th Eureopean Conference, Valencia, 22–27 August 2004, proceedings*. Valencia, IOS Press, 2004, Vol. 110, P. 239–243.
13. Song H., Rawat D. B., Jeschke S. et al. Cyber-Physical Systems. Foundations, Principles and Applications. Cambridge, Academic Press, 2017, 514 p.
14. Korotunov S., Tabunshchyk G., Henke K. et al. Analysis of the verification approaches for the cyberphysical systems, *Third International Workshop on Computer Modeling and Intelligent Systems (CMIS)*. Zaporizhzhia, 2019, pp. 950–961.
15. Habra N., Abran A., Lopez M. et al. A framework for the design and verification of software measurement methods, *Journal of Systems and Software,* 2008, Vol. 81, № 5, pp. 633–648. DOI: 10.1016/j.jss.2007.07.038.
16. Subbotin S. A. Diagnostic models synthesis method based on radial basis neural networks with support for shorthand properties, *Radío Electronics, Computer Science, Control,* 2016, № 2 (37), pp. 64–69. DOI: 10.15588/1607-3274-2016-2-8
17. Tuch H. OS Verification – Now! [Electronic resource]. Access mode: https://www.usenix.org/legacy/event/hotos05/final_papers_backup/tuch/tuch_html/index.html
18. Cervin A., Henriksson D., Lincoln B. How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime, *IEEE Control Systems Magazine,* 2003, Vol. 23, № 3, pp. 16–30. DOI: 10.1109/MCS.2003.1200240
19. Li B., Sun Z., Mechitov K. et al. Realistic case studies of wireless structural control, *Cyber-Physical Systems, 4th international conference, Philadelphia, 8–11 April 2013, proceedings*. Philadelphia, IEEE, 2013, pp. 179–188. DOI: 10.1109/ICCPS.2013.6604012
20. Kohtamaki T., Pohjola M., Brand J. et al. PiccSIM Toolchain – design, simulation and automatic implementation of wireless networked control systems, *Networking, Sensing and Contro, International conference, Okayama, 26–29 March 2009, proceedings.* Okayama, IEEE, 2009, pp. 49–54. DOI:10.1109/ICNSC.2009.4919244
21. Eyisi E., Bai J., Riley D. et al. NCSWT: An integrated modeling and simulation tool for networked control systems, *Simulation Modelling Practice and Theory,* 2012, Vol. 27, pp. 90–111. DOI: 10.1016/j.simpat.2012.05.004
22. Kiddle C., Simmonds R., Williamson C. et al. Hybrid packet/fluid flow network simulation, *Seventeenth Workshop on Parallel and Distributed Simulation (PADS).* San Diego, 2003, pp. 143–152. DOI: 10.1109/PADS.2003.1207430
23. Liu J. Parallel simulation of hybrid network traffic models, *21st International Workshop on Principles of Advanced and Distributed Simulation*, 2007, pp. 141–151. DOI: 10.1109/PADS.2007.26
24. Melamed B., Pan S., Wardi Y. HNS: A streamlined hybrid network simulator, *ACM Transactions on Modeling and Computer Simulation,* 2004, Vol. 14, № 3, pp. 251–277. DOI: 10.1145/1010621.1010623
25. Platzer A., Quesel J. D. KeYmaera: a hybrid theorem prover for hybrid systems (system description), *Automated Reasoning, 4th International Joint Conference, Berlin Heidelberg, 12–15 August 2008, proceedings.* Berlin, Springer-Verlag, 2008, pp. 171–178. DOI: 10.1007/978-3-540-71070-7_15
26. Banerjee A., Gupta K. S. Spatio-temporal hybrid automata for safe cyber-physical systems: A medical case study, *Cyber-Physical Systems, 4th international conference,*

DOI 10.15588/1607-3274-2020-3-5

Philadelphia, 8–11 April 2013, proceedings. Philadelphia, ACM/IEEE, 2013, pp. 71–80. DOI: 10.1145/2502524.2502535
27. Formal software verification: model checking and theorem proving : Embedded Systems Laboratory Technical Report : ESL-TIK-00214 / Massachusetts Institute of Technology ; M. Ouimet. Cambridge, 2007, 12 p.
28. Bagade P., Banerjee A., Gupta S. K. Safety assurance of medical cyber-physical systems using hybrid automata: a case study on analgesic infusion pump, *Medical CPS Workshop*, 2013, pp. 111–118. DOI: 10.1.1.294.6604
29. Sasnauskas R., Dustmann O. S., Kaminski B. L. et al. Scalable symbolic execution of distributed systems, *Distributed Computing Systems, 31st International Conference, Minneapolis, 20–24 June 2011, proceedings*. Minneapolis, IEEE, 2011, pp. 333–342. DOI: 10.1109/ICDCS.2011.28
30. Majumdar R., Saha I., Shashidhar K. C. et al. CLSE: closed-loop symbolic execution, *NASA Formal Methods Symposium*, 2012, pp. 356–370. DOI: 10.1007/978-3-642-28891-3_33
31. Pasareanu C. S., Rungta N. Symbolic PathFinder: symbolic execution of Java bytecode, *Automated Software Engineering, 25th international conference, Antwerp, 20–24 September 2010, proceedings*. Antwerp, IEEE/ACM, 2010, pp. 179–180. DOI: 10.1145/1858996.1859035
32. Clarke E. M., Zuliani P. Statistical model checking for cyber-physical systems, *Automated Technology for Verification and Analysis, 9th International Symposium,* Taipei, 2011, Vol. 6996, pp. 1–12. DOI: 10.1007/978-3-642-24372-1_1
33. Kalajdzic K., Jegourel C., Lukina A. et al. Feedback control for statistical model checking of cyber-physical systems, *International Symposium on Leveraging Applications of Formal Methods,* 2016, Vol. 9952, pp. 46–61. DOI: 10.1007/978-3-319-47166-2_4
34. Clarke E. M., Emerson E. A., Sistla A. P. Automatic verification of finite state concurrent systems using temporal logic specifications: a practical approach, *Proceedings of the 10th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, 1983, pp. 117–126. DOI: 10.1145/567067.567080
35. Dijkstra E. W. The humble programmer, *Communications of the ACM*, 1972, Vol. 15, № 10, pp. 859–866. DOI: 10.1145/355604.361591
36. Subbotin S. A. Synthesis of neuro-fuzzy networks ranking and specific encoding attributes for diagnostics and automatic classification of the precedents, *Radío Electronics, Computer Science, Control,* 2016, № 1 (36), pp. 50–57. DOI: 10.15588/1607-3274-2016-1-6
37. Abraham-Mumm E., Hannemann U., Steffen M. Verification of hybrid systems: formalization and proof rules in PVS, *Engineering of Complex Computer Systems, 7th international conference*, *Skovde, 11–13 June 2001*, proceedings. Skovde, IEEE, 2001, pp. 48–57. DOI:10.1109/ICECCS.2001.930163
38. Clarke E. M., Henzinger T. A., Veith H. et al. Handbook of Model Checking. Cham, Springer International Publishing, 2018, 1210 p.
39. Clarke E. M. Model Checking / E. M. Clarke, O. Grumberg, B. Peled. Cambridge : MIT Press, 2001, 314 p.
40. Gabbay D. Semantical Considerations for Modal Logics by Saul A. Kripke, *The Journal of Symbolic Logic*, 1969, Vol. 34, № 3, P. 501. DOI: 10.2307/2270922
41. Pnueli A. The temporal logic of programs, 18th Annual Symposium on Foundations of Computer Science (SFCS), Providence, 1977, pp. 46–57. DOI: 10.1109/SFCS.1977.32
42. Clarke E. M., Emerson E. A. Design and synthesis of synchronization skeletons using branching time temporal logic, *Workshop on Logic of Programs*, 1982, pp. 52–71.
43. Arras P., Tabunshchyk G., Okhmak V. et al. Modeling and simulation of the services for vehicle charging infrastructure interaction, *Intelligent Data Acquisition and Advanced Computing Systems, Technology and Applications, 10th international conference, Metz, 18–21 September 2019: proceedings*. Metz, IEEE, 2019, pp. 330–333. DOI: 10.1109/IDAACS.2019.8924449
44. Tabunshchyk G., Arras P., Korotunov S. et al. Cost optimization simulation for electric vehicle charging infrastructure, *IEEE European Technology & Engineering Management Summit (ETEMS).* Dortmund, 2020, pp. 76–88.
45. McNaughton R. Elementary computability, formal languages, and automata. Lawrence : Z B Publishing Industries, 1993, 400 p.
46. Arras P., Tabunshchyk G. Type or paste your text here to design optimization techniques in mechanical design and education of engineers convert case, *Advances in Design, Simulation and Manufacturing,* 2020, pp. 13–22. DOI: 10.1007/978-3-030-22365-6_2
47. Korotunov S., Tabunshchyk G., Okhmak V. Genetic algorithms as an optimization approach for managing electric vehicles charging in the Smart Grid, *Computer Modeling and Intelligent Systems, 3rd international workshop, Zaporizhzhia, 27–01 May 2020, proceedings*. Zaporizhzhia, CEUR WS, 2020, pp. 184–198.

DOI 10.15588/1607-3274-2020-3-5