УДК 004.75
DOI https://doi.org/10.32782/2663-5941/2025.6.2/48

Тіменко А.В.
Національний університет «Запорізька політехніка»;
Таврійський державний агротехнологічний університет імені Дмитра Моторного

Шкарупило В.В.
Національний університет біоресурсів і природокористування України;
Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова Національної академії наук України

Куликовська Н.А.
Національний університет «Запорізька політехніка»

Грушко С.С.
Національний університет «Запорізька політехніка»
Таврійський державний агротехнологічний університет імені Дмитра Моторного

ФОРМАЛЬНА МОДЕЛЬ ПЕРЕВІРКИ СУМІСНОСТІ КОМПОНЕНТІВ IOT-СИСТЕМИ

У статті запропоновано формальну модель перевірки сумісності компонентів IoT-систем на основі темпоральної логіки дій TLA та формалізму TLA+. При описі формальної моделі IoT-системи використовуються концепції «дії» та «динаміки». Модель базується на структурі Кріпке, де стани системи представлені декартовим добутком множин значень змінних вебсервісів, а переходи між станами формалізовані як темпоральні події. Розроблено механізм специфікації переходів на основі тернарного оператора, що дозволяє формувати ланцюжки викликів вебсервісів та описувати динаміки системи. Запропоновано два рівні постановки задачі перевірки сумісності: у вузькому сенсі для окремої динаміки та у широкому сенсі для всіх альтернативних сценаріїв функціонування системи. Модель дозволяє перевіряти як узгодженість протоколів взаємодії між компонентами, так і темпоральну узгодженість послідовностей викликів вебсервісів. Адекватність моделі підтверджена експериментально на прикладі системи обробки даних експериментів з двома альтернативними динаміками функціонування. Верифікація виконувалась за допомогою інструменту TLC (TLA+ model checker) для систем з 5, 10 та 15 вебсервісами на обчислювальній платформі Intel Core i5-4200M з 8 ГБ оперативної пам'яті. Виміряні часові витрати на верифікацію склали 1.2, 247 та 1843 секунди відповідно, що підтверджує експоненційне зростання обчислювальної складності при збільшенні кількості компонентів. Результати верифікації підтвердили коректність взаємодії компонентів для всіх розглянутих сценаріїв роботи системи. Запропонована модель забезпечує модульність опису компонентів, реконфігуровність специфікацій при зміні вимог без повного перепроєктування системи та можливість автоматизованої верифікації на етапі проєктування IoT-систем, що дозволяє виявляти потенційні проблеми сумісності на ранніх етапах розробки.

Ключові слова: *Інтернет речей, вебсервіс, формальна модель, специфікація, верифікація, композиція, сумісність, узгодженість, перевірка на моделі, великі дані.*

Постановка проблеми. Стрімке зростання кількості «розумних» пристроїв (смартфони, планшети, сенсори) актуалізує потребу у реалізації концепції Інтернету речей (Internet of Things, IoT) [1, с. 27; 2, с. 16], яка передбачає взаємодію мільярдів фізичних об'єктів через мережу Інтернет. Існує безліч сценаріїв реалізації IoT: «розумний дім» [3, с. 5], «розумне місто» [4, с. 22] тощо, де для ідентифікації пристроїв використовується технологія RFID [5, с. 25].

Різноманітність комунікаційних протоколів та взаємодіючих пристроїв обумовлює проблему інтероперабельності (сумісності). IoT-система розглядається на основі чотирирівневої архітектури [6, с. 2]: рівень сенсорів та з'єднань, мережевий рівень, рівень сервісів управління, рівень

«розумних» додатків. У даній роботі система розглядається на рівні сервісів як композиція взаємодіючих вебсервісів [7, с. 1], що забезпечує автоматизацію, повторне використання та гнучкість реконфігурування.

Компоненти системи взаємодіють шляхом обміну повідомленнями. Хоча проблема сумісності може бути вирішена стандартизацією [8, с. 23], практично використовуються різні альтернативні протоколи: MQTT, XMPP, CoAP тощо [9, с. 3].

Аналіз останніх досліджень і публікацій. Існують різні підходи до забезпечення сумісності IoT-систем. Один з підходів полягає у створенні механізму трансляції протоколів для індустріальних сценаріїв (IIoT) [10, с. 20]. Альтернативне рішення – використання інструментарію міжплатформної взаємодії [11, с. 7], де оперують поняттям вертикальної сумісності. Можливі рішення включають системи-шлюзи на мобільних пристроях [12, с. 28] та інструментарій HyperCat з IoT-концентратором [13, с. 7].

Для автоматизації взаємодії M2M (Machine-to-Machine) запропоновано M2M-шлюз для мобільних пристроїв [14, с. 13], що реалізує концепцію горизонтальної сумісності з забезпеченням QoS-характеристик [15, с. 14]. Компромісний підхід полягає у стандартизації з використанням формальних мов опису поведінки IoT-пристроїв, як у системі EUDroid [16, с. 5]. Задачу перевірки сумісності пропонується вирішувати на етапі проєктування, оперуючи моделями архітектури IoT-системи [17, с. 7].

Постановка завдання. Метою статті є підвищення ефективності процесу розробки IoT-систем шляхом розробки формальної моделі перевірки сумісності компонентів системи, призначеної слугувати базисом для проведення автоматизованої верифікації методом перевірки на моделі на етапі проєктування системи.

Наукова новизна роботи полягає у використанні концепції «дії» темпоральної логіки дій (TLA, Temporal Logic of Actions), що дозволило отримувати компактні реконфігуровані формальні специфікації, які відображають специфіку системи/

Виклад основного матеріалу. При описі формальної моделі IoT-системи використовуються концепції «дії» та «динаміки». Під «дією» розуміється виклик вебсервісу – компонента IoT-системи, під «динамікою» – послідовність таких викликів, що унікальним чином ідентифікують деякий обчислювальний процес – сценарій функціонування системи. Для формалізації озвучених концепцій використовується темпоральна логіка TLA та відповідний формалізм TLA+. Даний формалізм широко використовується при перевірці проєктних рішень для вебсервісів Amazon (AWS), при розробці надійних модулів програмної платформи системи управління залізничним рухом рівня SIL 4.

Формальна модель на основі структури Кріпке

Для формалізації «дії» використовується структура Кріпке [17, с. 45]. Структурою Кріпке M на множині атомарних висловлювань AP є структура наступного вигляду:

$$M = \langle S, \{s_0\}, R, L \rangle, \quad (1)$$

де S – скінченна множина станів, $s_0 \in S$ – початковий стан, $R \subseteq S^2$ – множина переходів між станами, $L : S \to 2^{AP}$ – функція розмітки станів.

Нехай $s \in S$ – деякий поточний стан, а $s' = R(s) \in S$ – наступний стан як результат переходу $(s,s') \in R$. «Дія» у даному ключі – це булева функція переходу, що приймає істинне значення при переході $(s,s') \in R$.

Для формування множини AP введемо дві множини:

$$V = \{v_i\}_{i=1}^{m \in N}, \quad (2)$$

$$D = \{d_1, d_2\}, \quad (3)$$

де V – множина змінних станів, D – множина значень змінних станів: $d_1 = 0$, $d_2 = 1$.. Кожна $v_i \in V$ представляє деякий вебсервіс у складі композиції, який або підлягає виклику, або вже був викликаний. Множину AP представимо декартовим добутком:

$$AP = V \times D. \quad (4)$$

Для акцентування уваги на діях, представимо множину AP у вигляді об'єднання:

$$AP = AP' \cup AP'', \quad (5)$$

де $AP' = \{ap_i'\}$ – множина передумов для дій: $ap_i' = (v_i, 0) \in AP' \subset AP$ – передумова для і-ї дії; $AP'' = \{ap_i''\}$ – множина постумов: $ap_i'' = (v_i, 1) \in AP'' \subset AP$ – постумова (результат) для і-ї дії.

Для представлення в формальній моделі сумісності двох сервісів за основу береться дія, що обумовлює перехід $(s,s') \in R$. Вводиться концепція «події» – e_i, яка представляється в формальній моделі як імплікація, модифікована темпоральним оператором Next (X) [28, с. 125]:

$$e_i \equiv (ap_i' \to X\, ap_i'') \equiv (\neg ap_i' \vee X\, ap_i''), \quad (6)$$

що означає: якщо умова $v_{pre,i}$ істинна, тоді відповідна умова $ap'_i \in AP'$ повинна бути істинною в наступний момент модельного часу. Аналогічним чином формалізується концепція «пустої події» – ee_i (empty event), яка означає, що деяка $v_i \in V$ не змінює свого значення:

$$ee_i \equiv (ap'_i \to X\, ap'_i) \equiv (\neg ap'_i \vee X\, ap'_i). \quad (7)$$

Запропоновану формалізацію подій (6) та (7) покладено в основу специфікації переходу $(s,s') \in R$. Сервіс, представлений змінною $v_i \in V$, розглядається як сумісний з сервісом $v_j \in V (i \neq j)$, за умови, що має місце відповідна подія e_i (6). Іншими словами, якщо сервіс $v_i \in V$ є ініціатором взаємодії з сервісом $v_j \in V$,то після виконання $v_i \in V$ має бути викликаний $v_j \in V$.

Для створення «подійного» каркасу специфікації переходу використовується тернарний оператор «?:» мови програмування С:

$$(e_i)?e_j : ee_j, \quad (8)$$

Специфікація на основі (8) слугує механізмом формування ланцюжків викликів: якщо умова e_i (6) істинна, перевірці підлягає умова e_j, інакше – ee_j (7). Алгоритм формування специфікацій переходів представлений на рис. 1.

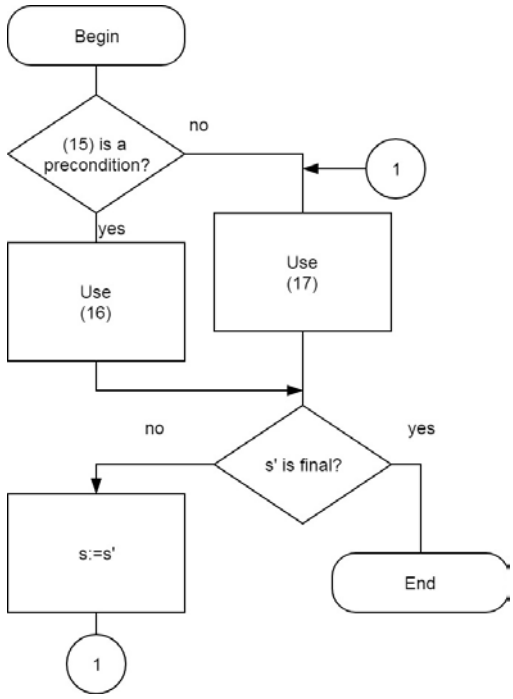

Рис. 1. Алгоритм формування специфікацій переходів

Формалізації ланцюжків викликів на основі (8) називаються «динаміками»:

$$b_k = e_1, e_2, ..., e_m, \quad (9)$$

Нижній індекс елементів ланцюжка подій вказує на відносний порядок виникнення подій: $e_1 \prec e_2 \prec ... \prec e_m$. Події, для яких виконується відношення $e_i \prec e_j$, називаються «сумісними» подіями. Темпоральну специфікацію на основі виразів (6) та (9) – шаблон специфікації динаміки – представимо кон'юнкцією:

$$f(b_k) \equiv e_1 \wedge e_2 \wedge ... \wedge e_m, \quad (10)$$

Оскільки система може функціонувати згідно з різними сценаріями, з кожною динамікою відотожнюється відповідна підмножина станів системи переходів:

$$S = \bigcup_{k=1}^{n \in N} S_k, \quad (11)$$

де $S_k \subseteq S$ – підмножина станів системи переходів, асоційована з і-ю динамікою, k – кількість альтернативних динамік (сценаріїв функціонування системи).

Поняття сумісності розглядається у вузькому та широкому сенсі. Якщо мається на увазі перевірка сумісності компонентів у контексті окремої динаміки, має місце постановка задачі у вузькому сенсі:

$$M, s^* \models f(b_k), \quad (12)$$

де $s^* \in S_k$ і $s^* \notin S \setminus S_k$. Вираз (12) означає, що для структури (1) темпоральна формула (10) повинна виконуватись для всіх станів $\forall s^* \in S_k$. Для формулювання задачі перевірки сумісності у широкому сенсі формується диз'юнкція темпоральних формул, що охоплює всі альтернативні динаміки:

$$\xi \equiv f(b_1) \vee f(b_2) \vee ... \vee f(b_k) \vee ... \vee f(b_n). \quad (13)$$

Постановка задачі перевірки сумісності у широкому сенсі:

$$M, s \models \xi. \quad (14)$$

Вираз (14) означає, що система повинна коректно функціонувати за будь-яким з можливих сценаріїв, забезпечуючи узгоджену взаємодію компонентів для всіх передбачених динамік. Це дозволяє перевірити сумісність компонентів IoT-системи на етапі проєктування до фактичного розгортання системи.

З метою перевірки адекватності запропонованої моделі було розглянуто практичний сценарій предметної галузі. Розглядався сценарій системи обробки даних експериментів, де користувач може виконувати наступні дії: підготовка та запуск експерименту, отримання даних (результату) проведення експерименту та візуалізація отриманих даних. Діаграма дій для розглянутого сценарію представлена на рис. 2.

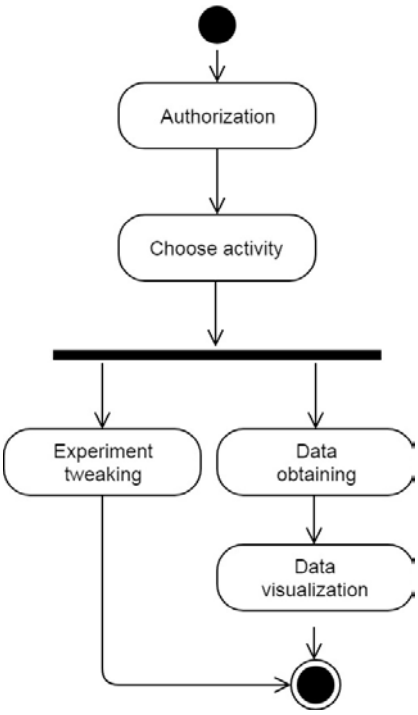

Рис. 2. Діаграма дій для розглянутого сценарію

На рис. 2 представлені дві альтернативні динаміки роботи з системою. Перша динаміка передбачає послідовність: підготовка експерименту запуск експерименту → отримання результатів. Друга динаміка включає додатковий крок візуалізації: підготовка експерименту → запуск експерименту → отримання результатів → візуалізація даних.

Для формальної специфікації системи було використано формалізм TLA+. Специфікація була синтезована згідно з запропонованою моделлю (формули 1–14). Специфікація включала: опис початкового стану системи (Init-предикат); опис переходів між станами (Next-предикат на основі формули 8); темпоральні умови узгодженості взаємодій компонентів (формули 6, 7); специфікації альтернативних динамік (формули 10, 13).

Верифікація синтезованої специфікації була виконана за допомогою інструменту TLC (TLA+ model checker). Перевірка проводилась на обчислювальній платформі з наступними характеристиками: процесор Intel Core i5-4200M, частота 2.5 ГГц, оперативна пам'ять 8 ГБ, операційна система Windows 10.

Були виміряні часові витрати, супутні автоматизованій верифікації специфікації. Результати вимірювань наведені в табл. 1.

Таблиця 1

Результати вимірювання часу верифікації

Кількість вебсервісів	Кількість динамік	Кількість станів	Час верифікації, с
5	2	32	1.2
10	5	1024	247
15	8	32768	1843

Як видно з табл. 1, час верифікації зростає експоненційно зі збільшенням кількості компонентів системи. Для системи з 5 вебсервісами час верифікації склав 1.2 секунди. Для системи з 10 вебсервісами час склав 247 секунд (близько 4 хвилин). Для системи з 15 вебсервісами час склав 1843 секунди (приблизно 30 хвилин). Результати верифікації підтвердили коректність взаємодії компонентів для всіх сценаріїв.

Висновки. У роботі запропоновано формальну модель перевірки сумісності компонентів IoT-системи на основі темпоральної логіки дій TLA та формалізму TLA+. Модель призначена для автоматизованої верифікації на етапі проєктування. Адекватність моделі підтверджена експериментально. Виміряні часові витрати (1.2-1843 секунд) є прийнятними. Практична значущість полягає у виявленні проблем сумісності на ранніх етапах проєктування, що зменшує ризики, знижує витрати та підвищує надійність IoT-систем. Перевагою запропонованої моделі є реконфігуровність – можливість адаптувати специфікацію до зміни вимог без повного перепроєктування. Компактність специфікацій полегшує їх розуміння та підтримку.

Подальші дослідження: розширення моделі для семантичної сумісності та QoS; оптимізація алгоритмів верифікації; інтеграція у інструментарії розробки; автоматична генерація специфікацій.

Список літератури:

1. Atzori L., Iera A., Morabito G. The Internet of Things: A survey. *Computer Networks*. 2010. Vol. 54, No. 15. P. 2787–2805.

2. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. 2013. Vol. 29, No. 7. P. 1645–1660.

3. Minerva R., Biru A., Rotondi D. Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiative*. 2015. P. 1–86.

4. Zanella A., Bui N., Castellani A., Vangelista L., Zorzi M. Internet of Things for Smart Cities. *IEEE Internet of Things Journal*. 2014. Vol. 1, No. 1. P. 22–32.

5. Want R. An introduction to RFID technology. *IEEE Pervasive Computing*. 2006. Vol. 5, No. 1. P. 25–33.
6. Sethi P., Sarangi S. R. Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*. 2017. Vol. 2017. Article ID 9324035.
7. Dustdar S., Schreiner W. A survey on web services composition. *International Journal of Web and Grid Services.* 2005. Vol. 1, No. 1. P. 1–30.
8. Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials.* 2015. Vol. 17, No. 4. P. 2347–2376.
9. Naik N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *Proceedings of IEEE International Systems Engineering Symposium (ISSE)*. 2017. P. 1–7.
10. Grangel-González I., Halilaj L., Coskun G., Auer S., Collarana D., Hoffmeister M. Towards a Semantic Administrative Shell for Industry 4.0 Components. *Proceedings of IEEE International Conference on Semantic Computing (ICSC)*. 2016. P. 230–237.
11. Noura M., Atiquzzaman M., Gaedke M. Interoperability in Internet of Things: Taxonomies and Open Challenges. *Mobile Networks and Applications*. 2019. Vol. 24. P. 796–809.
12. Bandyopadhyay S., Sengupta M., Maiti S., Dutta S. A Survey of Middleware for Internet of Things. *Recent Trends in Wireless and Mobile Networks*. 2011. Vol. 162. P. 288–296.
13. Blackstock M., Lea R. IoT Interoperability: A Hub-based Approach. *Proceedings of International Conference on the Internet of Things (IOT).* 2014. P. 79–84.
14. Desai P., Sheth A., Anantharam P. Semantic Gateway as a Service Architecture for IoT Interoperability. *Proceedings of IEEE International Conference on Mobile Services.* 2015. P. 313–319.
15. Perera C., Zaslavsky A., Christen P., Georgakopoulos D. Context Aware Computing for The Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials.* 2014. Vol. 16, No. 1. P. 414–454.
16. Ur Rehman M. H., Liew C. S., Abbas A., Jayaraman P. P., Wah T. Y., Khan S. U. Big Data Reduction Methods: A Survey. *Data Science and Engineering*. 2016. Vol. 1. P. 265–284.
17. Chen M., Mao S., Liu Y. Big Data: A Survey. *Mobile Networks and Applications*. 2014. Vol. 19. P. 171–209.

Timenko A.V., Shkarupylo V.V., Kulykovska N.A., Hrushko S.S. FORMAL MODEL FOR CHECKING THE INTEROPERABILITY BETWEEN THE COMPONENTS OF THE IOT SYSTEM

The article is devoted to the development of a formal model for verifying the compatibility of Internet of Things (IoT) system components based on Temporal Logic of Actions (TLA) and the TLA+ formalism. The model is based on Kripke structures, where system states are represented by Cartesian products of web service variable value sets, and transitions between states are formalized as temporal events. A transition specification mechanism based on a ternary operator has been developed, enabling the formation of web service invocation chains and the description of system dynamics. Two levels of compatibility verification problem formulation are proposed: in the narrow sense for a single dynamic and in the broad sense for all alternative system operation scenarios. The model allows verification of both communication protocol consistency between components and temporal consistency of web service invocation sequences. The adequacy of the model was confirmed experimentally using an experimental data processing system scenario with two alternative operation dynamics. Verification was performed using the TLC (TLA+ model checker) tool for systems with 5, 10, and 15 web services on an Intel Core i5-4200M platform with 8 GB RAM. The measured verification time costs were 1.2, 247, and 1843 seconds respectively, confirming exponential growth in computational complexity. The verification results confirmed the correctness of component interactions for all considered scenarios. The proposed model provides modularity in component description, specification reconfigurability when requirements change without complete system redesign, and the possibility of automated verification at the IoT system design stage, enabling identification of potential compatibility issues at early development stages..

Key words: *Internet of Things, web service, formal model, specification, verification, composition, compatibility, consistency, model checking, big data.*

Дата надходження статті: 26.11.2025
Дата прийняття статті: 16.12.2025
Опубліковано: 30.12.2025