Received: 4 March 2018 | Revised: 8 June 2018 | Accepted: 31 July 2018

DOI: 10.1002/dac.3808

RESEARCH ARTICLE

WILEY

Formal verification approaches in the web service composition: A comprehensive analysis of the current challenges for future research

Alireza Souri[1] | Amir Masoud Rahmani[1,2] | Nima Jafari Navimipour[3]

[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] Computer Science, University of Human Development, Sulaimanyah, Iraq

[3] Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

Correspondence
Amir Masoud Rahmani, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
Email: rahmani@srbiau.ac.ir

Summary

Today, service composition is emerging paradigm on the communication networks such as cloud environments, internet of things, wireless sensor network, and software-defined network. The goal of service composition method is to provide the interactions between user requirements and smart objects of intelligent communication systems. There have been many efforts to use formal verification and behavioral modeling methods to evaluate the service composition mechanisms. Up to now, there is not a comprehensive analysis research on this topic. Therefore, this paper focuses on several formal verification approaches that are performed to confirm the service composition correctness in communication networks. The objective of this paper is to comprehensively categorize and examine current research techniques on formal verification of the service composition. This research analysis provides an overview of recent service composition approaches according to structural and functional properties. Comparison results show that most of the verification approaches in explanation of the service composition correctness are semantic-aware approach with 43%. The most used verification method for the service composition is model checking with 69%. The process algebra is used 29%, and some theorem proving methods are applied in 9% of the investigated mechanism. Moreover, most widely used modeling tools are NuSMV (22%), SPIN (17%), CPN (12%), UPPAAL (12%), Event-B (10%), and PAT (5%).

KEYWORDS

correctness, formal verification, research analysis, service composition, web service

1 | INTRODUCTION

In many fields of communication networks such as cloud environments,[1-4] web technologies,[5,6] internet of things (IoT),[7,8] wireless sensor network (WSN),[9,10] and software-defined network (SDN),[11,12] service composition is currently a significant challenge due to the growth of the service providers and customer requests. Many sensors, smart devices, and intelligent mobile nodes are composed together to realize a suitable composited service for user's requests in a communication network. The service composition approach has an identical procedure to provide user requirements based on a pool of heterogeneous web services, various providers, and transactional business processes.[13-15]

The service composition as the NP-complete problem is evaluated using different methods such as computational knowledge methods, statistical testing, and operational research in communication systems.[5,16,17] Since formal verification can be used to evaluate the complex communication systems and NP-complete problems,[18] recent research studies have analyzed proposed methods using formal verification methods[19] in forms of model checking,[20-27] process algebra,[8,28-31] and theorem proving[32-34] methods. In other hand, formal verification method is an essential issue in the complex communication systems development such as cloud computing,[35] SDN, WSN,[36,37] and IoT.[24,26] By means of formal verification, we can authorize the functional specifications of distributed systems.[38,39] Also we can prove the conceptual interactions between behavioral modeling system and web services for existing communication systems.[40-42]

Checking correctness of the service composition is 1 of the fundamental advantages of formal verification. This advantage guarantees the existing specifications of the system behavior in the service selection and composition. Consequently, the main goal of the formal verification is to conduct a mathematical approach where the constraints of the system can be simply adapted to achieve some new results. There are some technical surveys and research studies on various parts of the service composition; however, they do not concentrate all of the formal verification aspects in the service composition mechanisms. To fill this weakness, this paper presents a comprehensive analysis on current opportunities of the formal verification approaches in the service composition. To the best of our knowledge, this is the first systematic analysis that gives the important knowledge about the formal verification techniques for the service composition. The ultimate objective of this research is to provide the conceptual aspects of formal verification for the service composition mechanisms. The key goals of this research are highlighted as follows:

- Presenting a comprehensive analysis of the formal verification approaches in the service composition.
- Designing a systematic classification to analyze the formal verification approaches in different service composition methods.
- Providing a statistical analysis and comparison of the existing formal verification approaches in the service composition.
- Emphasizing the future research directions and open perspectives on the related topics.

The structure of this research is organized as follows: The related work is demonstrated in Section 2. Section 3 addresses a research paper selection method based on the systematic analysis approach. Section 4 illustrates a systematic classification and analysis for selected formal verification approaches in the service composition methods. In addition, Section 5 shows discussion and comparison of the existing approaches. In Section 6, the formal verification issues in the service composition are described as new challenges and open issues. Finally, Section 7 shows the conclusion, future work and paper limitations.

2 | BACKGROUND AND RELATED WORK

In this section, some preliminaries and related works for formal verification of service composition are illustrated in intelligent communication systems.

2.1 | Background

In the communication networks such as cloud computing,[43] fog computing, WSN, and IoT environments as emerging technologies, service selection and composition process provides an outstanding platform for web services[44,45] that numerous categories of telecommunication services can be composed and interacted with each other. Interactions between telecommunication systems describe a category of sophisticated on-demand computing services between cloud and fog service providers (such as Amazon, Google, and Microsoft[46]) and physical smart devices (such as sensors, mobile devices, and smart things). Different web services are combined with each other from various cloud or fog service providers into composite services.

In a telecommunication framework, web service interactions with interface consistency are important matter. Service composition is 1 of the important problems in communication systems that many researchers proposed different approaches to solve it.[47,48] Usually, a composition request in distributed environments is represented in the form of a workflow.[49] The nodes of the requested services demonstrate the possible dependencies between tasks. There are several candidate web services with equal functionalities for realizing each task. Quality of service (QoS)-aware web service composition is the problem of selecting services for each task from several candidate web services with different nonfunctional properties meeting some user requirements. By considering the dominant use of service-oriented architecture in communication networks, QoS has received significant attention.[50] Quality of service factors are used to state the nonfunctional properties of web services including response time, availability, reliability, throughput, latency, and price.[44]

Figure 1 shows existing service composition designs that level (a) illustrates a sequential combination, (b) shows parallel composition, (c) presents switch composition, and (d) shows loop service composition. Of course, all of the service patterns can be composed and transferred with together. Calculating the QoS of the composite pattern suggests an initial calculating for other patterns. The aggregation functions are different for various QoS criteria and depend on the composition pattern.[51]

2.2 | Related work

This subsection describes a brief overview of the related survey papers. For example, Jatoth et al[5] presented a systematic literature-based review of the QoS-aware service composition approaches according to the computational intelligence mechanisms. They classified the existing methods into 3 categories that include heuristic, nonheuristic, and metaheuristic approaches. However, this paper suffers from some weaknesses; for example, the paper selection procedure is not clear and the studies are not compared completely.

One of the substantial studies of the service composition was presented by Han et al.[52] They presented a survey in the service composition of internet protocol smart objects. In this study, a comprehensive analysis was compared based on different factors such as smart object techniques, service modeling, target platforms, target applications, and composite service approaches for the internet protocol of IoT. The main defect of the presented review is that there is no categorized review on the incorporation of the existing studies. Also, the authors have not analyzed the evaluation factors such as nonfunctional and functional properties.

Also, Jula et al[53] presented a systematic review of the service composition approaches in cloud environments. They separated these approaches into 5 main algorithmic groups, including classical algorithms, combinatorial algorithms, machine-based methods, structural methods, and framework methods. Also, they analyzed the main QoS factors that are evaluated in the technical case studies. Of course, there is no comprehensive taxonomy and technical comparison in this review. In addition, the authors could not provide a technical comparison of the formal verification approaches in this field.

In addition, another survey was offered by Lemos et al[54] for the web service composition techniques. This survey addresses a web service composition taxonomy, including composition language, execution environment, automation platform, tool support, and users and knowledge reuse. However, the evaluation parameters to compare the research papers are not considered. Also, there is no structural mechanism to evaluate the fundamental composition techniques in this paper. Furthermore, this study does not analyze the formal verification approaches in the service composition mechanisms.

Finally, Garriga et al[55] presented a brief review of the web service composition approaches based on 4 main features, including adaptation view, REpresentational State Transfer (RESTful) view, verification and validation (V&V) view, and QoS awareness view. The RESTful view demonstrates HTTP properties of web services that can cover interoperability between cloud providers and end users. This review does not provide any systematic paper selection for comparing the gathered studies. Moreover, for each feature, the authors explained a brief description of the service composition approaches.

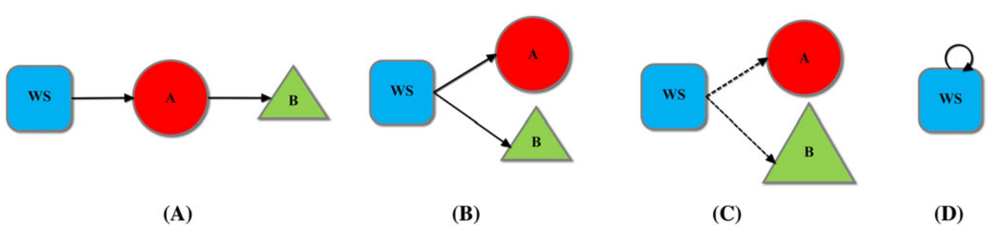

FIGURE 1 The composition pattern of the service composition. A, Sequential. B, Parallel. C, Switch. D, Loop

Due to the reviewed survey papers, there are some existing weaknesses as follows:

- The recent research studies do not present any analytical taxonomy and categorization in formal verification of the service composition.
- Some studies do not provide the formal approaches in the service composition techniques.
- Some studies do not evaluate the functional parameters in the service composition techniques.
- The structure of the existing studies does not have an efficient configuration for research study selection and analysis.
- The current review studies do not present the new challenges and open issues of the verification approaches in communication systems.
- Some studies do not consider the new range of the published studies in their comparison.

3 | RESEARCH SELECTION METHOD

This section demonstrates a systematic analysis as a research study assessment and critical evaluation for classifying the formal verification of service composition approaches.[5,56] The analysis process includes an explanation of finding the research study collection in scientific databases.[57] In this process, some electronic databases such as ACM, Science Direct, Springer, and IEEE are used. By stating synonyms and different words of the main features, the following search keywords are presented as follows[58-63]:

("Formal verification" OR "model checking" OR "theorem proving" OR "process algebra" OR "Formal analysis") AND ("service composition" OR "Service selection")

This systematic analysis was designed to respond the following analytical questions (AQ) according to the objectives and scope of the proposed study[64]:

- AQ1: Which service composition approaches are applied to verification?
- AQ2: What formal verification methods are usually used for service composition?
- AQ3: What current verification tools are used?
- AQ4: What modeling languages are provided to correctness the service composition?
- AQ5: What research approaches satisfy the specification rules of formal verification approaches?
- AQ6: What nonfunctional properties are evaluated in service composition approaches?

In Figure 2, we show 160 published papers by the year of publication. This figure illustrates the publications which are related to formal verification of the service composition approaches in an online environment (2010-2017). Furthermore, Figure 3 displays the distribution of the research papers over time according to some well-known journals such as IEEE, ACM, Springer, Elsevier, and Taylor and Francis. After providing the technical questions, we apply the inclusion/exclusion criteria for the final research selection. Due to the number of published research studies, only the journal articles and conference papers which are indexed in web of science and ISI proceedings are analyzed as important and peer-reviewed papers for the formal verification of service composition approaches in the last 7 years. Finally, 65 peer-reviewed papers are considered for further analysis in order to response our technical questions that

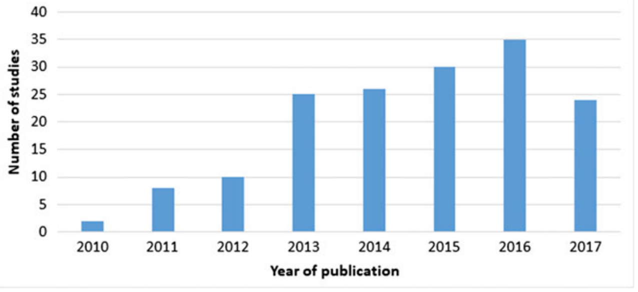

FIGURE 2 Distribution of the published papers by year of publication

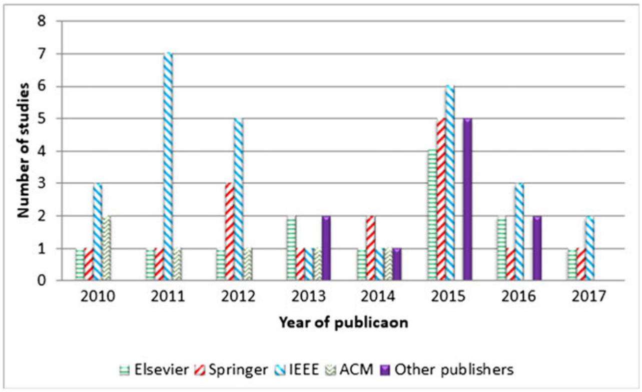

FIGURE 3 Distribution of the research papers based on publisher

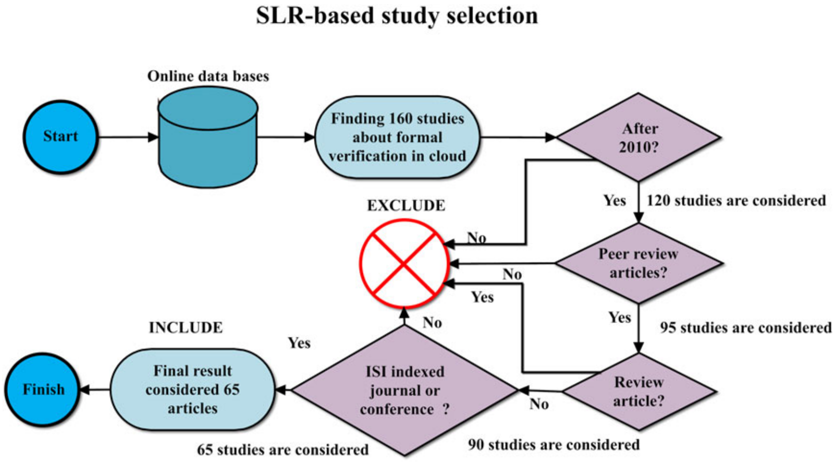

FIGURE 4 The selection criteria and evaluation framework of the research papers

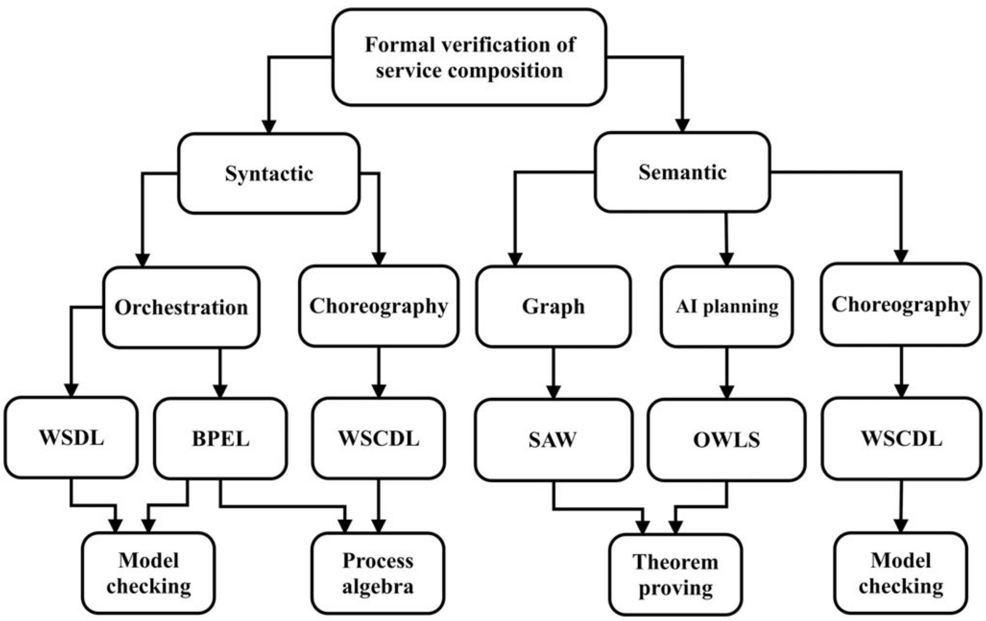

FIGURE 5 Taxonomy of the formal verification approaches on the service composition techniques

the details of the content are provided in Section 4. Figure 4 illustrates the selection procedure and evaluation framework for the existing papers. All of the non-ISI indexed papers are removed in this literature analysis. Also, we just select research studies that have English language context.

4 | CATEGORIZING OF SELECTED SERVICE COMPOSITION APPROACHES

This section presents a systematic analysis and classification for the selected formal verification techniques in the service composition according to the applied research analysis method. Service composition is a business-to-business platform to offer services with some relevant specification in order to generate an original functionality. For composing the required services dynamically, we need an iterative analysis of each specific service that is combined in a composition process. In spite of the service composition complication, it is still an exceptionally complex trend, and it is now beyond the human ability to manage the entire procedure substantially. For analyzing the automated composition of services, the formal verification approaches can be very useful to measure the functional and nonfunctional properties. These approaches are formalized into 3 main methods: process algebra,[30,65] model checking,[66,67] and automated theorem proving.[68] Figure 5 shows the taxonomy of the formal verification approaches in the service composition. Each service composition method includes 2 main classes: semantic and syntactic approach. Modeling the compositions is related to each approach according to the context of the technical study. In the semantic-aware service composition, the semantical information is specified according to the collected data from service providers' smart devices in IoT environments, sensors, and cloud infrastructures.[69] The requested services are preprocessed using context-aware ontologies that specify the interoperability between user-centric knowledge and smart objects.[70,71] Context-aware computing is based on the semantic relationships between user activities and smart devices that consider human objects, time, and situation of the event occurrence. For the semantic-aware composition method, there are 3 main methods, including graph, artificial intelligence (AI) and choreography methods with supporting Ontology Web Language Service (OWLS), Simple Additive Weighting (SAW), and Web Service Choreography Description Language (WSCDL) for service selection and composition. In the syntactic-aware service composition, service providers define common interfaces of the service nodes using service input/output specifications for requested determinations in the cloud.[72,73] In the syntactic-aware composition, there are 2 main methods, including orchestration and choreography methods with supporting Web Service Description Language (WSDL), Web Service Choreography Interface (WSCI), and Business Process Execution Language (BPEL).[18]

In this technical classification, we analyze formal verification approaches for service composition mechanisms dividing into 2 main categories that include structural and functional properties. The important structural properties are as follows:

- Verification method proves the system correctness.
- Modeling method demonstrates the modeling composition approach according to the existing algorithms such as metaheuristics and liner algorithms.
- Specification language describes the expected formal modeling properties that the system must satisfy them.
- Verification tool is a developed environment for analyzing and proving the system behavior.
- Development phase shows the extension of the studies on design or implementation of the system behavior.

The important functional properties are illustrated as follows[74]:

- Correctness property is specified to check the existing specification rules that should be satisfied in the state space of the system behavior.
- Reachability property is applied to check some assumed set of objective states that can be reached preliminary from a fixed set of initial states.
- Deadlock property is specified to analyze an individual state in a set of paths that state can be in a condition in which transition is not occurred.
- Safety property is specified to check a condition that a critical path will never occur in the total of the system behavior.
- Liveness property is applied to check when an event happens in a series of finite paths in parallel.
- State space reduction refers to decrease the nonseen states and events in a system behavior exploration that effects on verification time and system complexity.

Also, 3 important nonfunctional factors are defined as follows[75,76]:

- Availability: Availability can be appreciated in the service composition and application middleware phase for presenting various services at every time and different places according to user tasks.

- Response time: part of the time spent on combining the service and respond to a user task.
- Cost: The total price has to be paid by the service requester to achieve the best composite service.

After describing the evaluation factors, we compare and discuss the selected studies in the next subsections.

4.1 | Process algebra method in the service composition approach

In this subsection, we illustrate the process algebra approaches in the service composition. In addition, we discuss 14 selected algebraic-based service composition approaches in Section 4.1.1. Finally, their differences, advantages, and weaknesses are compared and discussed in Section 4.1.2. The main aspect of the process algebra method[30] is a formal characteristic mapping from system processes to a semantic behavioral model and conceptions of equality.[65]

4.1.1 | Analysis of the process algebra method in the service composition approaches

Rai et al[77] presented an approach based on process algebraic that applies the recursive communication messages for web service composition. This approach has 3 main operators, including successor process, composition, and recursive composition that facilitate the interconnections between different services. The authors analyzed the soundness of the composition method using a set of canonical behavioral equivalences that support deadlock freeness problem. The weakness of this paper is that the canonical behavioral equivalences are emphasized on the composition method statically, while current service composition methods have a dynamic behavior in responding to the customer requests. Also, the reachability, deadlock, and process equivalence proofing are not considered.

Also, Maroui et al[78] provided a formal method based on process algebra for model-driven approach in WSDL method. The authors applied the automated proof of the composition correctness approach using AcmeStudio tool.* The main framework of this research is to translate the WSDL models to the SysML method and formulate SysML method to the Acme tool. Some specification properties are checked with the verification process that shows the correctness of this method correctly. Some weaknesses of this study are addressed as follows: (1) This study has not implemented a practical context as the plug-in for the automatic WSDL models, (2) using a mutation analysis to prove the concrete specification rules in verification results, and (3) having a high complexity to evaluate the extended WSDL models.

Yan et al[79] presented a cost-aware service composition approach based on process algebra that satisfies the executive minimum paths using communicating sequential processes. The authors considered a single event interface for composing the selected services. This interface is a defect for current composition approaches because the service composition approaches have multiple reactions between existing user interfaces. Also, they have not evaluated some specification rules such as reachability, safety, and liveness in this scenario.

Wang et al[80] provided a formal specification method based on Object-Z to evaluate Web Service Modeling Ontology where different parts of the formal language were clearly defined in 1 composition system. This algebraic method supports both dynamic and static conditions of each composition approach by using semantic precondition and postcondition operators. Each service is translated to a specified element that connects to the appropriate semantic conditions. By allocating existing values to the semantic conditions, the consistency problem is proven for the service composition approach. The weakness of this formal method is that the complexity of the composition approach for translating all of the preconditions and postconditions to Object-Z is very high and increases exponentially.

Yeung[81] presented a process algebra approach for analyzing the correctness of the choreography-based web service composition approach. This approach has modeled the service discovery and service composition phases using the pi-calculus method. First, the WSCDL model is generated by some collaborative activities. Then, the generated model is formulated according to pi-calculus operators and translated to communicating sequential process language. Finally, the FDR tool† has applied the verification process on the translated model. Some relationships between the WSCDL model and service operators are evaluated using the specification rules. The main weaknesses of this study are low availability, low reachability, and omitting deadlock evaluation.

Some same studies such as[82-84] use pi-calculus method for specifying the correctness of the service composition method. These approaches provided a pi-calculus method on web service BPEL model that satisfies the subcontract

*acme.able.cs.cmu.edu/AcmeStudio/

†https://cran.r-project.org

TABLE 1 Comparison of the structural properties of the process algebra method

Research	Service Description	Service Model	Specification Language	Tool	Development Phase
Rai et al[77]	-	-	Pi-calculus	-	Design
Laneve and Padovani[82]	-	-	Pi-calculus	-	Design
Li et al[83]	-	-	Pi-calculus	-	Design
Lahouij et al[85]	BPEL	Orchestration	-	Event-B, ProB tool	Implementation
Babin et al[86]	BPEL	Orchestration	-	Event-B	Design
Maroui and Ayeb[78]	-	-	SysML	Acme studio	Design
Ait-Sadoune and Ait-Ameur[87]	BPEL	Orchestration	-	Event-B, ProB	Implementation
Chen et al[84]	-	-	Pi-calculus	-	Design
Yan et al[79]	-	-	CSP	-	Design
Hamel et al[88]	BPEL	Orchestration	-	Event-B	Implementation
Wang et al[80]	WSDL	Orchestration	-	Object-Z	Design
Yeung[81]	WSDL	Choreography	CSP	FDR	Implementation
Amato and Moscato[89]	WSDL	Choreography	OFL	-	Design
Wang et al[90]	BPEL	Orchestration	Pi-calculus	-	Design

relations in the principle of the service replacement safely. The authors used a formal semantic to translate the BPEL scenarios to the pi-calculus method according to the system-level abstraction. The main defect of these studies indicates the development phase just presented at design level without any implementation and verification tool. Some research papers such as[85-88] provide the evaluation environment using the Event-B tool as the formal specification method in the service composition methods. For example, Babin et al[86] provided a refinement-based process algebraic approach to BPEL service composition. They illustrated their case study by formalizing the channels and operators using Event-B. This approach has 4 steps, including transition-based composite service, creating failure modes, service recovery, and translating compensating service to the Event-B platform. Some of the existing failure events are checked according to the transition systems which are defined in the Event-B tool. Ait-Sadoune and Ait-Ameur[87] presented a new Event-B platform to formalize the semantic web service composition according to BPEL specifications. The recursive fault detection is shown in this research based on static and dynamic transition systems. Each composite service uses a fault handler that formalizes the existing variables on the appropriate selected services. The presented platform is implemented using BPEL2B[‡] plugin that is developed by eclipse environment. The main advantage of this approach is finding state faults in this plugin dynamically. The main defect of this study is low reachability of the fault handler events.

Furthermore, Amato et al[89] presented a formal specification method based on the Operational Flow Orchestration Language for designing some existing cloud features such as events and composition attributes in the cloud service composition approach. The Operational Flow Orchestration Language method qualifies the cloud elements based on pattern prediction methods in the cloud service compositions. This study presents 2 main novelty as the main advantages, including (1) providing interactions of service components directly using pattern prediction methods and (2) analyzing the cloud composite services according to existing QoS factors semantically. The main weakness of this study is omitting the evaluation of some critical conditions such as deadlock, reachability, availability, and safety using model transformations.

Wang et al[90] proposed the interoperable cloud service composition based on pi-calculus process algebra methods. This formal method is applied to evaluate the mobile cloud service composition, including mobile terminal unit, cloud unit, and communications unit. The applied pi-calculus method used space-time slots to facilitate the interconnections of the service units according to operational semantics and time operators. The verification results have shown the feasibility of the proposed formal specification method to select the appropriate mobile service composition in a parking electric fence system as the technical case study. The main advantage of this study is considering time operators for

[‡]https://eclipse.org/bpel/

TABLE 2 Comparison of functional properties of the process algebra method

Research	Correctness Properties	Reachability	Deadlock	Safety	Liveness
Rai et al[77]	√	×	×	×	×
Laneve and Padovani[82]	√	×	√	×	×
Li et al[83]	×	×	√	×	×
Lahouij et al[85]	√	×	×	×	×
Babin et al[86]	√	×	×	×	×
Maroui and Ayeb[78]	×	√	√	×	×
Ait-Sadoune and Ait-Ameur[87]	√	×	×	×	×
Chen et al[84]	√	×	×	×	×
Yan et al[79]	√	×	√	×	×
Hamel et al[88]	√	×	×	×	×
Wang et al[80]	√	√	×	×	×
Yeung[81]	√	√	×	×	×
Amato and Moscato[89]	×	√	×	×	×
Wang et al[90]	√	×	×	×	×

communications unit in mobile services. The important weaknesses of this study are low availability and high complexity without considering a verification environment to evaluate the correctness of the case study.

4.1.2 | Summary of the process algebra method in the service composition approaches

According to the discussed and reviewed process algebra-based service composition approaches, the comparison of the structural properties of the process algebraic methods and their metrics is illustrated in Table 1. Table 1 shows the service description, service model, specification language, tool, and development phase of each technical article in formal verification of the service composition approaches. The main weakness of the process algebra-based approach is omitting the satisfaction of specification rules in the state space exploration.

Table 2 shows a side-by-side comparison of the functional properties of the process algebraic-based service composition approaches. This comparison analyzes the correctness of properties, reachability condition, deadlock freeness condition, safety and liveness specifications.

4.2 | Model checking method in the service composition approach

In this subsection, we show the model checking-based approaches in the service composition. Also, we review 45 selected approaches in service composition according to Kripke structure (KS) and labeled transition systems (LTS) methods in Section 4.2.1. Finally, the technical results, advantages, and disadvantages are compared in Section 4.2.2. The model checking approach has 2 main methods including KS and LTS.[66,91-93] Some well-known model checkers are applied to verification approach such as NuSMV,[§] SPIN,[¶] PAT,[#] and UPPAAL**.[74,94] The existing model checking methods specify a state space exploration of the system behavior using reachable graph of binary decision diagrams.[18,95]

[§]http://nusmv.fbk.eu/

[¶]http://spinroot.com

[#]http://pat.sce.ntu.edu.sg/

**http://www.uppaal.org/

4.2.1 | Analysis of the model checking method

Bataineh et al[96] presented a model checking method on the agent-based service composition approach. In this method, a BPEL model is created to transform into an automata model. So, the proposed model is encoded into the interpreted structure programming language that is provided for the input of the MCMAS model checker. Also, this method presents a new property generation to evaluate the system behavior correctness according to the computation tree logic formulas which provide checking the amenability of the business contracts in web service composition formally. The weakness of this research is to mark the BPEL process manually, and there is still the state space explosion problem by increasing the business processes.

Groefsema et al[97] provided a Kripke-based model checking method for the business process service composition. First, a colored Petri net (CPN) methodology is applied to create reachability graph of the composition services using the LTS model. This model confirms the timed business processes for translating the LTS model to the Kripke model. Finally, the KS is generated and reduced according to the compliance specifications in selecting required atomic propositions. The final reduced Kripke model is implemented to the NuSMV by checking some linear temporal logic (LTL) and computation tree logic (CTL) specification rules. The execution time of the verification has shown that time of the reduced Kripke method is lower than the actual model. However, the simplicity of the proposed case study is specified obviously. Also, the specified temporal properties are not complex that satisfy some logical problems such as reachability, deadlock, safety, and liveness properties.

Furthermore, Huynh et al[98] provided the LTS-based verification method to compose the web services in BPEL model. The proposed approach presents the on-the-fly sate exploration of model checking method for web services that provide a bitwise-indexing method to find the best composition. In the experimental results, a real dataset is chosen to analyze the number of the extended states, visited states, and the execution time of service composition approach. The weakness of this study is omitting the specification rules such as deadlock and reachability conditions, without considering the implementation platform for the proposed composition approach.

Mi et al[99] presented a probabilistic-based model checking to compose subweb services using the Markova model. The BPEL as the service modeling language is used to add the reliability attributes to each subservice in composition approach. Each BPEL activity sample code is translated to a probabilistic LTS model using Markova decision process. Then, the proposed model is transferred to the probabilistic LTS model in order to analyze the CTL formulas in PRISM model checker. The main advantage of this study is considering a high-quality time probabilistic factor to design the recommender service composition approach. This study suffers from low availability and high complexity conditions.

Nagamouttou et al[100] provided an enhanced stacked automata model to compose the BPEL service model. This model is based on a combined Muller and push-down automata approach to achieve the best service composition approach. The correctness of the proposed model is analyzed using Promela language according to liveness, deadlock, and reachability conditions in the SPIN tool. In the evaluation results, the number of the examined states and response time is compared with the other studies. Also, the authors have not used temporal logic formulas to assess the correctness of the system.

In addition, Ghannoudi et al[101] presented the LTS-based model checking approach to compose the B2B e-commerce services. This approach identifies the internal actions for each invocated service in composition method. The proposed LTS model is implemented in UPPAAL model checker. The CTL formulas are specified to evaluate the correctness of the presented approach in terms of nonblocking, safety, and liveness properties.

Sun et al[102] addressed a verification approach for a core subset of BPEL to compose the web services using mCRL2. The specified functional activities of the service composition approach are translated to the structured values of the business process using the automatic translation tool in Java. Finally, some simple properties are evaluated in the mCRL2 tool according to the generated state transition graph.

Furthermore, Kil et al[103] presented the LTS-based verification approach for semantic service composition approach. This research addresses the OWL ontology for composing the web services dynamically. The experimental results are evaluated using SMV, SAT Solver, and zChaff[††] model checkers that compare the execution time and the number of the explored states. The main weakness of this study is that the authors have not considered the specification rule evaluation for the proposed service composition approach.

Oussalah et al[104] provided a formal architecture for the flexible web service composition approach dynamically. This research used Maude tool for verifying the abstract level of the black-box components in each composed services. This research provides a proven correctness for existing specifications in Cafe-object tool.

[††] https://www.princeton.edu/~chaff/zchaff.html

Dumez et al[105] proposed a model-driven composition approach to verify the web services. They used the LOTOS formal description method to evaluate the verification time and the correctness of the proposed approach. First, the WSDL model is imported to the design level, and the best solution for the composition of existing services is applied. Then, the BPEL codes are generated for translating to the formal specification of the LOTOS. Finally, the specification model is created using LTS method to run in the CADP tool. The verification results have shown that the proposed approach is well-corrected for composing the best services. Of course, this research omits the specification rules for analyzing the correctness of properties.

Moreover, Mukherjee et al[106] presented a spring framework for BPEL activities to compose the web services. First, the proposed method is modeled using CPN tool for generating the state reachability graph. Then, the state graph is translated to the Promela according to LTS model. Finally, the translated model is implemented using SPIN model checker. There is no specification proofing for the temporal logic formulas in this research.

Also, Bianculli et al[107] proposed the LTS-based framework to decompose and compose the web services in WSDL. A service behavior analysis is used for specifying the formal description of the service composition method. Finally, the translated model is implemented using LTSA tool for evaluating some critical problems such as safety reachability properties.

Guermouche et al[108] addressed a formal explanation of timed choreographies in service composition approach. This explanation is based on timing and behavioral features of the system that are specified with Fiacre language. The created service composition is translated to the TINA model checker to evaluate the specified temporal properties. The main weaknesses of this study are low reachability and high complexity.

Chen et al[109] designed a verification tool by VeriWS for evaluating nonfunctional and functional properties of web service composition approach. This verification tool has addressed the semantic web service composition approach using semantic principles according to the formal operational semantics of WS-BPEL. This tool has 2 main verification sides in which 1 side operates the verifier process in terms of LTL verifier, reachability, and deadlock verifier. The other side of this tool has an aggregator process that evaluates the response time, cost, and availability of the composition approach.

Luo et al[110] presented a transitional formal model for a multiagent-based web service composition approach. The proposed formal model is developed using the epistemic MCTK model checker. First, a translation process is presented to convert the OWLS to the finite state machine model using agent observable variables. This translation is based on LTS method that looks like the *smv* codes to implement temporal properties in forms of CTL formulas. Of course, the verification time and memory usage have not been evaluated in this research.

Bourne et al[111] presented a KS-based approach to evaluate the business process as a service models in service composition approach. First, the authors proposed a transactional behavior model for the business process as a service using the selected features of the services in composition approach. Then, the proposed model is translated to the KS method in order to perform in NuSMV. Also, a state space reduction method is considered for the final Kripke model in composition approach. The experimental results have shown that the verification time for the reduced model is lower than the original model. However, the LTL and CTL formulas have not been considered to evaluate the correctness of the transactional behavioral system.

Hu et al[20] provided a multitenant architecture for verifying the dynamic service composition approach in a cloud environment. This architecture presents a verification as a service method with 4 main layers that include a user interface, model translation, verification controlling, and database layers. The verification side of this research is based on the LTS method that evaluates the CTL formulas in forms of deadlock, dependency detection, and reachability properties. Finally, the number of the properties per response time is evaluated in this architecture.

Xi et al[112] presented a secure LTS-based model checking approach to evaluate a service chain composition approach. The authors evaluated the proposed approach using the number of the candidate services per verification time using SPIN model checker.

Furthermore, Salva et al[113] provided a symbolic transition model for monitoring the service composition approach in cloud computing. This model has 2 main levels, including proxy tester and spectator composition to evaluate the best composition service. This model is implemented to Z3 model checker that evaluates the verification time for each number of the visited services.

Also, some research papers[114-118] considered the verification process of the proposed WSCDL-based service composition approach in timed transition systems using UPPAAL model checker. For example, Emilia Cambronero et al[117] presented formal timed automata for verification of the service composition approach in UPPAAL model checker. This approach provides a translation method to convert WSCDL indexes to timed metamodel in LTS method.

TABLE 3 Comparison of the structural properties of the model checking method

Research	Service Description	Service Model	System Model	Property Language	Tool	Development Phase
Bataineh et al[96]	BPEL	-	KS	CTLC	MCMAS	Implementation
Tout et al[123]	BPEL	Orchestration	KS	-	CPN	Design
Groefsema et al[97]	-	-	KS	CTL	NuSMV, CPN	Implementation
Bourouis et al[132]	SOG	-	LTS	LTL	-	Design
Huynh et al[98]	-	-	LTS	-	PAT	Implementation
Yuan et al[119]	BPEL	Orchestration	KS	-	SMV	Implementation
Zhao et al[133]	-	-	LTS	CSP, LTL	PAT	Implementation
Mi et al[99]	BPEL	Orchestration	LTS	PCTL	Prism	Implementation
Rai et al[134]	WSDL	-	-	LTL	-	Implementation
Bourne et al[120]	-	-	KS	-	TL-VIEWS NuSMV	Design
Du et ak[114]	-	-	LTS	LTL	UPPAAL	Implementation
Nagamouttou et al[100]	WSCDL	Choreography	LTS	LTL, Promela	SPIN	Implementation
Ghannoudi and Chainbi[101]	-	-	LTS	LTL	UPPAAL	Implementation
Ben Azaiez and Sbaï[121]	-	-	KS	CTL	NuSMV	Implementation
Rebai et al[124]	-	-	LTS	LTL	SPIN	Implementation
Chen et al[109]	-	-	KS	LTL	VeriWS	Implementation
Martinelli et al[135]	BPEL	Orchestration	KS	-	-	Design
Sun et al[102]	BPEL	Orchestration	LTS	-	mCRL2	Implementation
Xiangyu et al[136]	BPEL	Orchestration	LTS	-	ZING	Design
Li et al[125]	WSDL	-	KS	-	PIPE tool	Implementation
Mukherjee et al[106]	BPEL	Orchestration	KS	-	CPN	Implementation
Luo et al[110]	OWLS	-	KS	CTL	MCTK	Implementation
Kacem et al[126]	BPEL	Orchestration	LTS	LTL	SPIN	Implementation
Guermouche and Zilio[108]	BPEL	Choreography	LTS	-	TINA	Implementation
Ibrahim and Khalil[115]	BPEL	Choreography	LTS	-	UPPAAL	Implementation
Zhao et al[127]	BPEL	Orchestration	LTS	-	LOTOS	Implementation
Todica et al[128]	BPEL	Orchestration	LTS	LTL	SPIN	Implementation
Zhang et al[116]	WSCDL	Choreography	LTS	-	UPPAAL	Implementation
Kil and Nam[103]	WSCDL	Choreography	KS	-	SMV	Implementation
Chemaa et al[129]	BPEL	-	KS	LTL	Maude	Implementation
Oussalah and Zeghib[104]	BPEL	-	KS	LTL	Maude	Implementation
Yau et al[137]	BPEL	Orchestration	KS	LTL	-	Design
Fan et al[138]	WSDL	Orchestration	KS	-	CPN	Design
Bentahar et al[122]	BPEL	Orchestration	KS	CTL	NuSMV	Implementation
Dumez et al[105]	BPEL	Orchestration	LTS	-	LOTOS, CADP	Implementation
Ni and Fan[131]	OWLS	Orchestration	KS	-	CPN	Implementation
Kokash et al[139]	BPEL	Orchestration	LTS	Reo	mCRL2	Implementation
Quyet et al[130]	BPEL	Orchestration	LTS	LTL, Promela	SPIN	Implementation
Emilia Cambronero et al[117]	WSDL	Choreograph	LTS	-	UPPAAL	Implementation
Bianculli et al[107]	BPEL	Orchestration	LTS	-	LTSA	Implementation

(Continues)

TABLE 3 (Continued)

Research	Service Description	Service Model	System Model	Property Language	Tool	Development Phase
Machado et al[118]	SWRL	Orchestration	LTS	CTL	UPPAAL	Implementation
Bourne et al[111]	BPEL	Orchestration	KS	LTL	NuSMV	Implementation
Hu et al[20]	BPEL	Orchestration	Graph	CTL	SPIN	Implementation
Xi et al[112]	BPEL	Orchestration	LTS	-	SPIN	Implementation
Salva and Cao[113]	BPEL	Orchestration	LTS	-	SMT solver Z3	Implementation

Then, the contextual time traces are applied to translate the semantic principles of each composition service to the network of timed automata. Finally, some logical properties are specified to analyze the correctness of the composition behavior.

In the other researches such as Yuan,[119] Bourne et al,[120] Ben Azaiez and Sbaï,[121] and Bentahar,[119-122] the authors considered the KS method to verify the service composition approaches using NuSMV model checker. For example, Ben Azaiez and Sbaï[121] presented a Kripke-based modeling approach for the service composition approach that WSCDL attributes are mapped into the behavioral open workflow net (OWF-net) model. The final OWF-net is implemented with NuSMV model checker. Some CTL deadlock properties are applied to evaluate the proposed service composition approach. Also, the complexity of the algorithm and the verification time of the proposed approach are evaluated according to the statistical analysis.

Tout et al[123] provided a new aspect-oriented grammar offering higher adaptability and preventing interactive engagements between unlike features of the service composition. The aspect-oriented method has proposed for WS-BPEL to permit description analysis of the conceptual features. Moreover, the proposed method formally is verified. A real case study, samples, and experimental outcomes used in the BPEL development environment have established the correctness and effectiveness of this research. As the weakness of this research, a new model should be proposed to avoid the existing conflicts between conceptual features.

Furthermore, Rebai et al[124] presented a formal method for web services using the SPIN tool. This method automatically translates WS-CDL choreography conditions to the Promela modeling language. The authors focused on the verification of nonfunctional specifications that are classified according to their nature into 2 categories: universal and internal specifications. Moreover, the CDLVT tool is presented to implement the proposed approach using an online shopping application as a case study which shows the usefulness and applicability of this solution. The verification process could be generalized in order to apply to a set of locally collected choreographies using various methods for service composition such as BPEL4Chor.

Li et al[125] presented a formal approach for context-independent connection that a web service can be replaced with other peers of the comparable behavior without any other web service composition approaches. So, the verification cost of service substitutability has reduced largely. Moreover, an algorithm provided for the verification process which made the verification of behavioral similarity of the web services automatic. Also, this study developed a petri net tool called PIPE that models some service composition architectures in verification approaches.

Kacem et al[126] presented the verification of the service orchestrations using model checking. This study has an adoption of WS-BPEL 2.0 as the language that defines the orchestration. The SPIN tool has been used for the verification process of this model. The WS-BPEL characteristics are transformed into Promela code as the SPIN model checker input language examining generic and specific properties expressed with the LTL formulas. An environment called BPELVT as BPEL verification tool is presented to evaluate the proposed approach. It has provided the WS-BPEL manager as the automatic procedure transformation of WS-BPEL to Promela code and model-checking views. The correctness of the transformation rules from WS-BPEL to Promela does not discuss in the verification process.

Also, in Zhao,[127] Todica et al,[128] Chemma et al,[129] and Quyet et al,[130] the authors addressed a BPEL modeling for service composition approach using LTS-based model checking method. In another study, Ni et al[131] provided a composition model of the CPN which is converted from OWLS model. This model expresses the consistent associations among the subprocesses of the service composition obviously and verifies the correctness of the service composition using formal methods of CPN tool. The verification algorithms are presented for the reachability, semantic consistency, and boundness of composed services.

TABLE 4 Comparison of functional properties of the model checking method

Research	Correctness Properties	Reachability	Deadlock	Safety	Liveness	State Space Reduction	
						Theory	Implementation
Bataineh et al[96]	√	×	×	×	×	×	×
Tout et al[123]	×	×	√	×	×	×	×
Groefsema et al[97]	√	√	×	×	×	√	×
Bourouis et al[132]	×	×	×	×	×	×	×
Huynh et al[98]	×	×	×	×	×	×	×
Yuan et al[119]	√	×	×	×	×	×	×
Zhao et al[133]	√	×	✓	×	×	×	×
Mi et al[99]	√	×	×	×	×	×	×
Rai and Gangadharan[134]	√	×	√	×	×	×	×
Bourne et al[120]	×	×	×	×	×	×	×
Du et al[114]	√	√	×	×	×	×	×
Nagamouttou et al[100]	√	×	√	√	×	×	×
Ghannoudi and Chainbi[101]	√	×	×	×	×	×	×
Ben Azaiez and Sbaï[121]	√	×	×	×	×	×	×
Rebai et al[124]	√	×	×	√	√	×	×
Chen et al[109]	√	√	√	×	×	×	×
Martinelli and Matteucci[135]	×	×	×	×	×	×	×
Sun et al[102]	√	×	×	×	×	×	×
Xiangyu et al[136]	√	×	×	×	×	×	×
Li et al[125]	×	√	√	×	×	×	×
Mukherjee et al[106]	×	×	×	×	×	×	×
Luo et al[110]	√	×	×	×	×	×	×
Kacem et al[126]	√	×	×	√	√	×	×
Guermouche and Zilio[108]	√	×	×	×	×	×	×
Ibrahim and Khalil[115]	√	√	×	√	×	×	×
Zhao et al[127]	√	×	×	×	√	×	×
Todica et al[128]	√	×	×	√	√	×	×
Zhang et al[116]	√	×	×	×	×	×	×
Kil and Nam[103]	×	×	×	×	×	×	×
Chemaa et al[129]	√	×	×	×	×	×	×
Oussalah and Zeghib[104]	√	×	×	×	×	×	×
Yau and Chua[137]	√	×	×	×	×	×	×
Fan et al[138]	×	×	×	×	×	×	×
Bentahar et al[122]	√	√	×	√	×	√	×
Dumez et al[105]	×	×	×	×	×	×	×
Ni and Fan[131]	√	√	×	√	×	×	×
Kokash et al[139]	×	×	×	×	×	×	×

(Continues)

TABLE 4 (Continued)

Research	Correctness Properties	Reachability	Deadlock	Safety	Liveness	State Space Reduction Theory	State Space Reduction Implementation
Quyet et al[130]	×	×	×	√	√	×	×
Emilia Cambronero et al[117]	√	√	√	×	×	×	×
Bianculli et al[107]	√	×	×	×	×	×	×
Machado et al[118]	√	×	×	×	×	×	×
Bourne et al[111]	√	√	×	×	×	√	√
Hu et al[20]	√	√	√	×	×	×	×
Xi et al[112]	×	×	×	×	×	×	×
Salva and Cao[113]	×	×	×	√	×	×	×

4.2.2 | Summary of the model checking methods in the service composition approaches

Table 3 illustrates the structural properties of the model checking service composition approaches. This table contains service description, service model, modeling method, property language, tool, and development phase of each technical article in model checking of the service composition approaches.

Table 4 presents a comprehensive comparison of the functional properties of the verification of the service composition approaches. This comparison analyzes the correctness of properties, reachability condition, deadlock freeness condition, safety and liveness specifications, and the state space reduction procedure. Just 1 research paper by Bourne et al[111] provides the state space reduction technique in forms of both theory and implementation levels.

4.3 | Automated theorem proving methods in the service composition approach

This subsection illustrates 6 technical studies related to the theorem proving-based service composition approaches in Section 4.3.1. Also, the comparison of the related approaches is addressed using some advantages and weaknesses in Section 4.3.2.

4.3.1 | Analysis of the automated theorem proving methods

Menadjelia[140] presented a formal verification approach based on theorem proving and analyzing the correctness of the protocol complexity of the recovery method in the service composition. This paper has proved the relations of the decision paths with the transition-based method. A delegator with 6 operations as deciding factors to execute 3 services is considered as a technical case study. Finally, the NP-complete problem for replacement paths using recovery method is proven. The main defect of this study is that the number of the deciding factors is very low and simple according to the existing transition relations. Zhao et al[141] presented a behavioral equivalent method to evaluate the choreography-based service composition approach. The behavioral equivalence method is based on the theorem proving approach that provides the strong and weak biosimilar analysis according to Chor language.[‡‡] The choreographies of the WSDL model is applied to some specification rules in 2 biosimilar analyses. The main weaknesses of this study are low availability and high complexity.

Moreover, Papapanagiotou et al[142] provided higher order logic (HOL)-based classical linear process to assess the web service composition approach. First, the authors presented a pi-calculus transition procedure based on proofs as processes for handling the web service selection. In addition, the structure of the web service composition is proven with proofs as processes. Of course, some specification properties such as liveness, deadlock, and safety have not been applied in this research. Jeffords et al[143] presented a fault-tolerant-based service composition approach that is evaluated with a transition-aware theorem proving method. This research has provided the strong proof definition for figuring out an

[‡‡]www.chor-lang.org

TABLE 5 Comparison of the structural properties of the automated theorem proving method

Research	Service Description	Service Model	Theorem Method	Tool	Development Phase
Menadjelia[140]	-	-	Graph isomorphic	-	Design
Zhao et al[141]	WSDL	Choreography	Property-based	-	Design
Papapanagiotou and Fleuriot[142]	BPEL	Orchestration	-	HOL, Piviz	Implementation
Jeffords et al[143]	BPEL	Orchestration	Partial refinement	-	Design
Papapanagiotou et al[144]	BPEL	Orchestration	-	HOL	Implementation
Hahn et al[145]	BPEL	-	Pattern-based	-	Design

approach, where a model or potentially code of the fault tolerant framework as of now exists, and the client needs to exhibit that the fault-tolerant framework fulfills the properties of a completely dedicated fault-tolerant augmentation. Finally, some fault specifications are considered to evaluate the proposed service composition approach. This study suffers from the high response time of verification and high complexity.

Papapanagiotou et al[144] provided a diagrammatic-based theorem proving a method for web service composition. The authors illustrated an arrangement of graphical creation that monitors to prove levels in the classical linear process. The composition approach was proven with a tensor view mechanism to provide a transition-based relation between services. The HOL-GUI[§§] environment is used for evaluating the correctness of the composition approach. Hahn et al[145] presented a proof-based language PEWS to evaluate the correctness of the web service composition approach. This language supports multiple instance synchronization for composing the selected services. The syntax of this language is based on high order theorem proving method.

4.3.2 | Summary of the automated theorem proving methods in the service composition approaches

Table 5 shows the structural properties of the theorem proving-based service composition approaches. This table contains service description, service model, theorem method, tool, and development phase of each technical article in the theorem proving-based service composition approaches.

Table 6 shows a side-by-side comparison of the functional properties in the process algebraic-based service composition approaches. This comparison analyzes the correctness of properties, reachability condition, deadlock freeness condition, and safety specifications. A few studies have evaluated the reachability and safety factors for the service composition approach.

5 | DISCUSSION AND COMPARISON

In this section, an analytical discussion of declared formal verification approaches in service composition is considered. We classify the discussed research papers based on the several structures to respond to the analytical questions in Section 3 as follows:

- AQ1: Which service composition approaches are applied to verification?

Figure 6 presents a comparison side of the service composition approaches up to now in which formal verification methods are used for evaluating their correctness. In this literature, there are 6 service composition approaches, including semantic-aware, syntactic-aware, QoS-aware, secure-aware, trust-aware, and model-driven. The semantic-aware approach has the highest percentage of the verification usage in the literature. The QoS-aware service composition as the NP-complete problem is determined and resolved through some proof techniques such as formal verification and evolutionary computing methods with 15 percentage usage of verification. Also, the trust-aware composition approaches are provided topically on high-level prediction approaches to increase the scalability of the system. In addition, 43% of the studies focus on semantic-aware service composition, 31% of the studies concentrate on

§§

TABLE 6 Comparison of functional properties of the automated theorem proving method

Research	Correctness Properties	Reachability	Deadlock	Safety
Menadjelia[140]	×	√	×	×
Zhao et al[141]	√	×	√	×
Papapanagiotou and Fleuriot[142]	√	×	×	×
Jeffords et al[143]	√	×	×	×
Papapanagiotou et al[144]	×	×	×	×
Hahn et al[145]	×	×	√	√

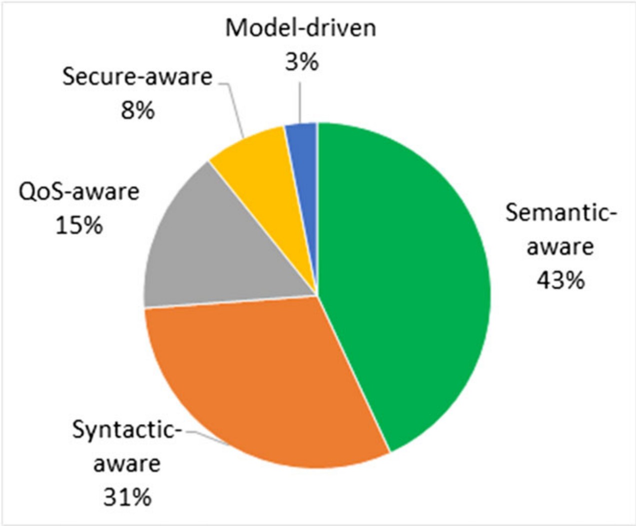

FIGURE 6 The statistical percentage of the service composition approaches in the literature

syntactic-aware service composition, 15% of the studies focus on QoS-aware service composition, 8% of the studies concentrate on secure-aware service composition, and 3% of the studies focus on model-driven service composition comprehensively.

- AQ2: What formal verification methods are usually used for service composition?

According to Figure 7, the model checking method has the number of highest usage in 65 research papers for the formal verification of the service composition. Because automated theorem proving method is a very hard-proof mechanism for evaluating the correctness of the system behavior, the number of research studies in this area is lower than the other verification approaches. Also, pi-calculus and Event-B tool have the highest percentage of the verification

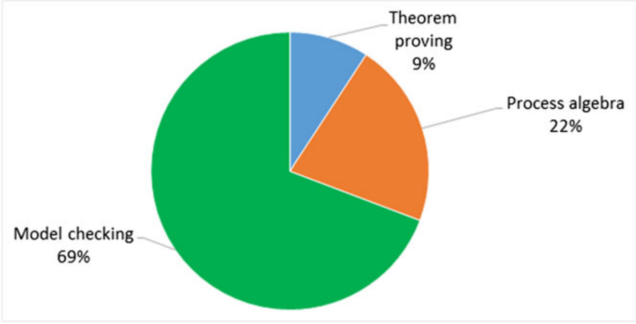

FIGURE 7 The analytical percentage of the formal verification approaches in service composition studies

usage for service composition in the literature. For modeling proof, the LTS method is higher than KS method. However, the process algebra approach has the following limitations: defect in generating the state space exploration and high time complexity. The model checking approach has the following defects: state space explosion problem, data path complexity, and memory consumption. The main disadvantages of theorem proving approach are as follows: writing hard mathematical proofs, hard operational semantics, no implementation platform, and high time complexity. However, there is no composite verification method to prove the correctness of the service composition as a new challenge, increase the scalability, decrease response time and verification cost, and avoid the above defects for each verification approach.

- AQ3: What current verification tools are used?

Also, Figure 8 displays the statistical percentage of the modeling tools that are applied for this literature review. The NuSMV tool is applied to 22% of the case studies in service composition papers. In addition, the LTL formula usage is more than the CTL formulas in the specification methods. Also, the existing research papers based on NuSMV model checker apply the modeling method on the proposed service composition approach using KS while some service composition approaches can be modeled using LTS with respect to the action-based behavior of the composition approach in the NuSMV platform.

- AQ4: What modeling languages are provided to correctness the service composition?

In addition, Figure 9 illustrates the percentage of the modeling languages in service composition literature. The BPEL is a popular modeling language in the service composition approach with 70% usage. Also, some articles have used the WSDL modeling language with 12% of the proposed composition approaches. In addition, the WSCDL has 7%, the OWLS has 5%, and the SAW has 3% usage.

- AQ5: What research approaches satisfy the specification rules of formal verification approaches?

We review used functional properties of each research to answer TQ5. The functional properties help researchers to evaluate correctness of the proposed service composition approach in all of the state space. Figure 10 displays statistical percentage of the functional properties that are applied to this literature review. The key functional properties are applied to the research papers that include correctness of property, reachability, deadlock-free, safety, and liveness. Most of the research studies have applied the correctness of property for evaluating the proposed case studies with 47%.

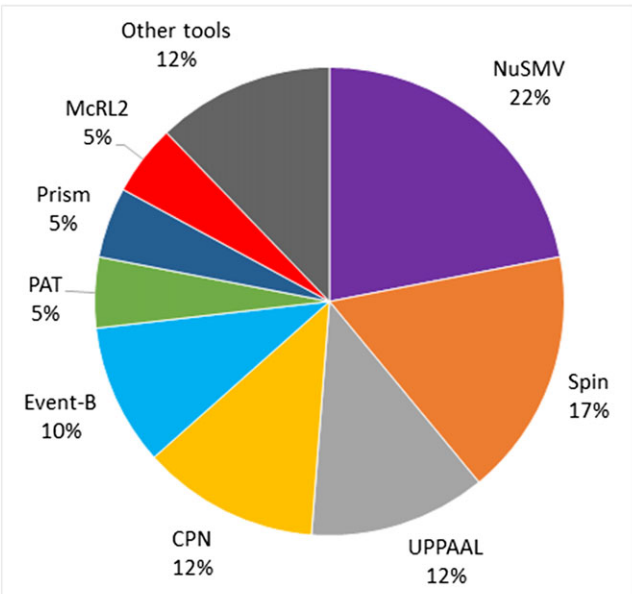

FIGURE 8 The percentage of the model checker tools in the literature

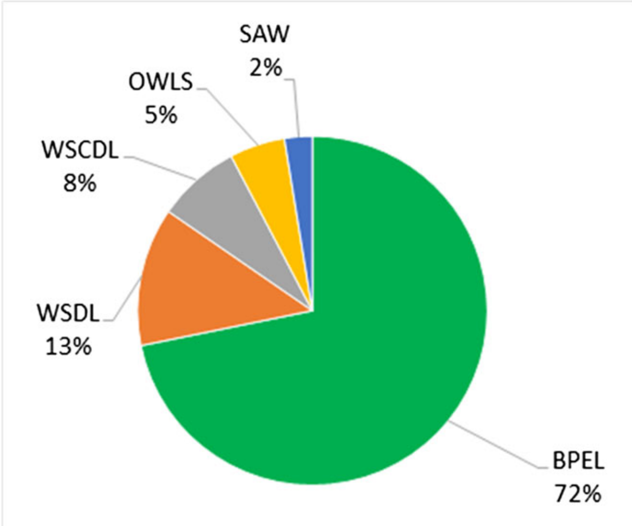

FIGURE 9 The percentage of the modeling languages in the literature

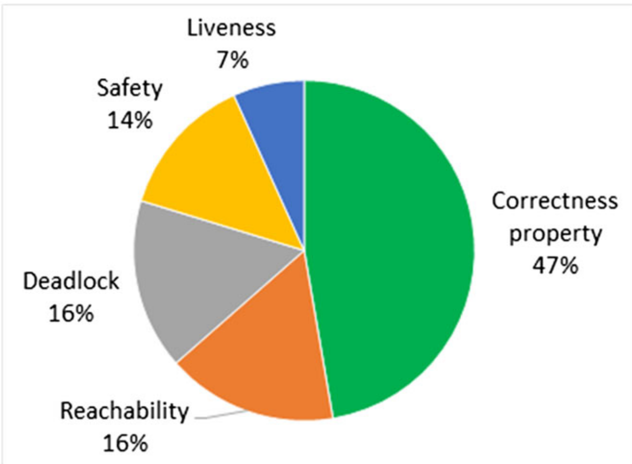

FIGURE 10 The percentage of the applied functional properties in this literature

We have observed that the LTL specifications have been most used for specifying correctness of the properties in this literature. Safety property is 1 of the main factors for safety-critical systems such as trust and secure-aware service composition approaches that a few papers have applied this property in their specification rules. There are only 4 papers that apply both formal verification and simulation methods on the presented service composition approach. This challenge helps proposed service composition approach to be evaluated completely with functional and nonfunctional properties. In addition, the reachability condition has 16%, the deadlock property has 16%, the safety property has 14%, and the liveness property has 7% usage.

- AQ6: What nonfunctional properties are evaluated in service composition approaches?

The nonfunctional properties that are evaluated in the service composition approaches are shown in Figure 11. The statistical percentage of the nonfunctional properties presents that the response time factor has most usage in the evaluation of the composition approaches by 59%, the cost has 31%, and availability has 10%.

Table 7 displays the distribution of the research papers for each publication in the literature. In addition, the country and quality rank of each journal paper is illustrated. The *IEEE Transactions on Service Computing* journal has involved some additional authors in the field of the formal verification of the service composition than any other journal. Also, most researchers in this field have presented their studies at the Procedia computer science in service composition and

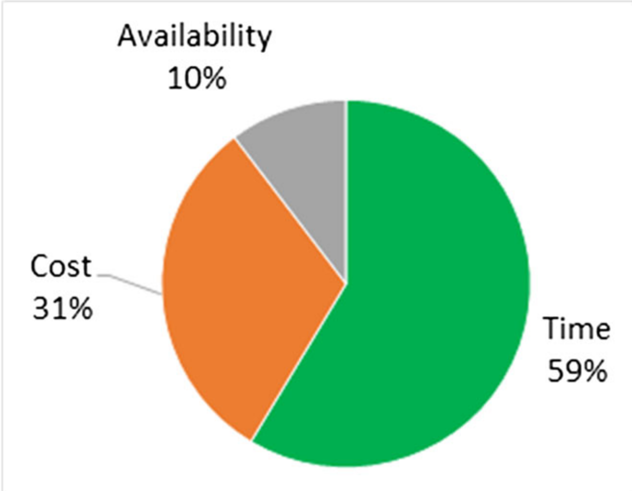

FIGURE 11 The percentage of the applied non-functional properties in the literature

TABLE 7 List of distribution of the articles per publications

Nom.	Source	Q metric in 2017	Country	Number
1	Expert Systems with Applications	Q1	United Kingdom	6
2	IEEE Transactions on Services Computing	Q1	United States	4
3	Journal of Network and Computer Applications	Q1	United States	4
4	Information Systems	Q1	United Kingdom	2
5	Computers & Electrical Engineering	Q2	United Kingdom	2
6	Formal Aspects of Computing	Q3	Germany	2
7	IEEE Transactions on Systems, Manufacturing, and Cybernetics	Q1	United States	1
8	Future Generation Computer Systems	Q1	Netherlands	1
9	Advances in Engineering Software	Q1	United Kingdom	1
10	Simulation Modelling Practice and Theory	Q1	Netherlands	1
11	IEEE Access	Q2	United States	1
12	Formal Methods in System Design	Q2	Germany	1
13	Journal of Logic and Algebraic Programming	Q2	Netherlands	1
14	International Journal of Computer and Mathematics	Q3	United Kingdom	1
15	International Journal of Web and Grid Services	Q3	United Kingdom	1
16	International Journal of Computer Applications in Technology	Q3	United Kingdom	1
17	International Journal of Critical Computer-Based Systems	Q3	Switzerland	1
18	International Journal of Web Engineering and Technology	Q4	United Kingdom	1
19	International Journal of Autonomous and Adaptive Communications Systems	Q4	United Kingdom	1
20	Vietnam Journal of Computer Science	Q4	Vietnam	1

verification than any other conference. Most of the papers in the service composition are published in ICSE, ICWEW, ICSC, ICWS, and other major conferences. Between 65 papers, 30 papers were published in major journals, including *IEEE Transaction of Service Computing*, *Expert Systems with Applications*, *Future Generation Computer Systems*, *Information Systems*, *Journal of Network and Computer Applications*, *Computers & Electrical Engineering*, and *Formal Aspects of Computing*. A list of distribution of articles for each publication is presented in Table 7.

6 | OPEN ISSUES

Due to applying the comprehensive analysis method on data collection of the formal verification approaches, some new open issues and research directions are illustrated as follows:

- In the model checking approaches, generating and analyzing the expected temporal logic specifications are very critical issues in forms of LTL and CTL formulas. Of course, existing verification tools only provide an editor environment for writing temporal logic formulas. It is important that a model verifier environment can generate the temporal logic formulas automatically. Due to advantages of model checking approach according to editable design, coding, and without using mathematical proof, some complex reactive and concurrent systems such as IoT, cloud, and SDN can be modeled and analyzed in the model checker tools.[146]
- The state space explosion problem should be optimized to achieve optimal service verification approaches using new modeling techniques such as Kripke transition systems and multilabeled transition systems. In addition, some metaheuristic algorithms can be used for minimizing the state space generation. Up to now, some important trends in service composition such as multidimensional service composition, interoperability composition, and energy consumption in the service composition were introduced which have not been addressed by using formal verification methods specially.
- Formal verification of the service composition in IoT[147,148] is a main challenge due to interconnecting heterogeneous smart devices. The correctness analysis of IoT service composition in some critical systems such as health care and industrial systems is an important challenge.[149] So, evaluation of the complex communications between IoT smart devices and cloud services[150] using formal verification is a challengeable issue.[151]
- Verification of service interoperability[152,153] between enterprise systems and cloud services is a new challenge for the service composition issue. The service interoperability provides an integrated architecture to interact the cloud service providers and enterprise systems.
- Formal verification of multicloud service composition[154] is an open issue for minimizing the number of cloud providers and communication cost between existing services. Because these factors are the NP-complete, formal verification methods can influence on feasibility and the correctness of their satisfactions.

7 | CONCLUSION AND LIMITATION

This paper presents a comprehensive analysis for the formal verification methods in the service composition approaches. During this research, we have contracted a comprehensive understanding addicted to the service composition and considerations of open issues to synthesize the collected data. In this literature, we have presented comprehensive analysis method by applying the exploration query on 160 studies that were published between 2010 and 2017. Finally, we have examined 65 studies that focused on formal verification of service composition. We have observed that the most used verification approach for explaining service composition correctness is semantic-aware (43%). As regards TQ2, we have found that the most used verification method for service composition is model checking by 69%. The process algebra has used 29%, and theorem proving method has applied 9% for verifying the service composition approaches. Due to TQ3, we have observed that the most widely used modeling tools are NuSMV (22%), SPIN (17%), UPPAAL (12%), Event-B (10%), and PAT (5%). According to TQ4, we have found that the most widely applied modeling language in the literature is BPEL (47%). As the TQ5, most of the research studies have applied the correctness of property for evaluating the proposed case studies with 47%. We have observed that the LTL specifications have been the most useful for specifying the correctness of properties in this literature. As the TQ6, the statistical percentage of the QoS properties presents that the response time factor has the most usage in the evaluation of the composition approaches by 59%.

Some limitations of this study are presented as follows: (1) omitting non-English studies; (2) removing non-ISI indexed studies; (3) removing all of the thesis dissertations, book chapters, and white papers; and (4) omitting nonpeer-reviewed published studies. We have believed that this literature analysis addresses the conceptual features of the formal verification for the service composition mechanisms.

In the future work, we will examine a hybrid formal verification approach to assess the correctness of a QoS-aware cloud service composition approach with increasing high performance efficiency and minimizing response time and memory usage factors. Merging a metaheuristic algorithm with formal verification approach can be efficient to analyze and evaluate the complex service composition approaches.

ORCID

Alireza Souri http://orcid.org/0000-0001-8314-9051
Amir Masoud Rahmani http://orcid.org/0000-0001-8641-6119
Nima Jafari Navimipour http://orcid.org/0000-0002-5514-5536

REFERENCES

1. Lin Y-D, Thai MT, Wang CC, Lai YC. Two-tier project and job scheduling for SaaS cloud service providers. *J Netw Comput Appl*. 2015;52:26-36.
2. Chi Y, Li X, Wang X, Leung VCM, Shami A. A fairness-aware pricing methodology for revenue enhancement in service cloud infrastructure. *IEEE Sys J*. 2017;11(2):1006-1017.
3. Souri A, Asghari P, Rezaei R. Software as a service based CRM providers in the cloud computing: challenges and technical issues. *J Service Sci Res*. 2017;9(2):219-237.
4. Mishra SK, Puthal D, Sahoo B, Jena SK, Obaidat MS. An adaptive task allocation technique for green cloud computing. *The J Supercomput*. 2018;74(1):370-385.
5. Jatoth C, Gangadharan GR, Buyya R. Computational intelligence based QoS-aware web service composition: a systematic literature review. *IEEE Trans Services Comput*. 2017;10(3):475-492.
6. Wang P, Ding ZJ, Jiang CJ, Zhou MC, Zheng YW. Automatic web service composition based on uncertainty execution effects. *IEEE Trans Services Comput*. 2016;9(4):551-565.
7. Kaur N, Sood SK. An energy-efficient architecture for the internet of things (IoT). *IEEE Syst J*. 2017;11(2):796-805.
8. Chen L, Thombre S, Jarvinen K, et al. Robustness, security and privacy in location-based services for future IoT: a survey. *IEEE Access*. 2017;5:8956-8977.
9. Amin R, Hafizul Islam SK, Biswas GP, Obaidat MS. A robust mutual authentication protocol for WSN with multiple base-stations. *Ad Hoc Netw*. 2018;75-76:1-18.
10. Mostafaei H, Chowdhury MU, Obaidat MS. Border surveillance with WSN systems in a distributed manner. *IEEE Syst J*. 2018;1-10.
11. Lin Y-D, Lai YC, Teng HY, Liao CC, Kao YC. Scalable multicasting with multiple shared trees in software defined networking. *J Netw Comput Appl*. 2017;78:125-133.
12. Lin P-C, Lin YD, Wu CY, Lai YC, Kao YC. Balanced service chaining in software-defined networks with network function virtualization. *Computer*. 2016;49(11):68-76.
13. Pan, L., Bo An, Shijun Liu Nash equilibrium and decentralized pricing for QoS aware service composition in cloud computing environments. in 2017 *IEEE International Conference on Web Services (ICWS)*. 2017.
14. Wang H, Zou B, Guo G, Yang D, Zhang J. Integrating trust with user preference for effective web service composition. *IEEE Trans Services Comput*. 2017;10(4):574-588.
15. Rodríguez-Mier P, Mucientes M, Lama M. Hybrid optimization algorithm for large-scale QoS-aware service composition. *IEEE Trans Services Comput*. 2017;10(4):547-559.
16. Liu, H., Farong Zhong, Bang Ouyang, Jiajie Wu An approach for QoS-aware web service composition based on improved genetic algorithm. in 2010 *International Conference on Web Information Systems and Mining*. 2010.
17. Sheng QZ, Qiao X, Vasilakos AV, Szabo C, Bourne S, Xu X. Web services composition: a decade's overview. *Inform Sci*. 2014;280:218-238.
18. Souri A, Navimipour NJ, Rahmani AM. Formal verification approaches and standards in the cloud computing: a comprehensive and systematic review. *Comput Standards & Interfaces*. 2018;58:1-22.
19. Ahmad W, Hasan O, Pervez U, Qadir J. Reliability modeling and analysis of communication networks. *J Netw Comput Appl*. 2017;78:191-215.
20. Hu K, Lei L, Tsai W-T. Multi-tenant verification-as-a-service (VaaS) in a cloud. *Simulation Modelling Prac Theory*. 2016;60:122-143.
21. Jafari Navimipour N. A formal approach for the specification and verification of a trustworthy human resource discovery mechanism in the expert cloud. *Expert Syst Appl*. 2015;42(15–16):6112-6131.
22. Jafari Navimipour N, Habibizad Navin A, Rahmani AM, Hosseinzadeh M. Behavioral modeling and automated verification of a cloud-based framework to share the knowledge and skills of human resources. *Comput Industry*. 2015;68:65-77.

23. Keshanchi B, Souri A, Navimipour NJ. An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing. *J Syst Softw*. 2017;124:1-21.
24. Souri A. Formal specification and verification of a data replication approach in distributed systems. *Int J Next-Generation Computing*. 2016;7:-1.
25. Souri A, Jafari Navimipour N. Behavioral modeling and formal verification of a resource discovery approach in grid computing. *Expert Syst Appl*. 2014;41(8):3831-3849.
26. Souri A, Norouzi M, Safarkhanlou A, Haghi Sardroud SHES. A dynamic data replication with consistency approach in data grids: modeling and verification. *Baltic J Modern Computing*. 2016;4(3):546.
27. Roumane A, Kechar B, Kouninef B. Formal verification of a radio network random access protocol. *Int J Commun Sys*. 2017;30(18): e3447.
28. Rezaee A, Rahmani AM, Movaghar A, Teshnehlab M. Formal process algebraic modeling, verification, and analysis of an abstract fuzzy inference cloud service. *The J Supercomputing*. 2014;67(2):345-383.
29. Ruiz MC, Cazorla D, Pérez D, Conejero J. Formal performance evaluation of the map/reduce framework within cloud computing. *The J Supercomputing*. 2016;72(8):3136-3155.
30. Hermanns H, Herzog U, Katoen J-P. Process algebra for performance evaluation. *Theoretical Comput Sci*. 2002;274(1):43-87.
31. Tini S, Larsen KG, Gebler D. Compositional bisimulation metric reasoning with probabilistic process calculi. *Logical Methods Comput Sci*. 2017;12.
32. Sardar MU, Hasan O, Shafique M, Henkel J. Theorem proving based formal verification of distributed dynamic thermal management schemes. *J Parallel Distributed Computing*. 2017;100:157-171.
33. Srikanth, A., B. Sahin, and W.R. Harris. Complexity verification using guided theorem enumeration. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*. 2017. ACM
34. Xue T, Ying S, Wu Q, et al. Verifying integrity of exception handling in service-oriented software. *Int J Grid Utility Computing*. 2017;8(1):7-21.
35. Hajimirzaei, B. and N.J. Navimipour, Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. ICT Express, 2018.
36. Sutrala Anil K, Sutrala AK, Das AK, et al. On the design of secure user authenticated key management scheme for multigateway-based wireless sensor networks using ECC. *Int J Commun Sys*. 2018;31(8):e3514.
37. Ahmadinia M, Movaghar A, Rahmani AM. Ontology-based modelling and information extracting of physical entities in semantic sensor networks. *IETE J Res*. 2018;1-17.
38. Qin Z, Sun J, Wahaballa A, Zheng W, Xiong H, Qin Z. A secure and privacy-preserving mobile wallet with outsourced verification in cloud computing. *Comput Standards & Interfaces*. 2017;54, **Part 1**:55-60.
39. Copet PB, Marchetto G, Sisto R, Costa L. Formal verification of LTE-UMTS and LTE–LTE handover procedures. *Comput Standards & Interfaces*. 2017;50:92-106.
40. Dennis L, Fisher M, Slavkovik M, Webster M. Formal verification of ethical choices in autonomous systems. *Robotics Autonomous Syst*. 2016;77(Supplement C):1-14.
41. Folwarczný L, Knop D. IV-matching is strongly NP-hard. *Inf Process Lett*. 2017;125(Supplement C):5-8.
42. Chen Z. Parametric runtime verification is NP-complete and coNP-complete. *Inf Process Lett*. 2017;123(Supplement C):14-20.
43. Ghobaei-Arani M, Rahmanian AA, Shamsi M, Rasouli-Kenari A. A learning-based approach for virtual machine placement in cloud data centers. *Int J Commun Syst*. 2018;31(8):e3537.
44. Hayyolalam V, Pourhaji Kazem AA. A systematic literature review on QoS-aware service composition and selection in cloud environment. *J Netw Comput Appl*. 2018;110:52-74.
45. Fang M et al. Web service discovery utilizing logical reasoning and semantic similarity. *Int J Commun Sys*. 0(0):e3561.
46. Barakat L, Miles S, Luck M. Adaptive composition in dynamic service environments. *Future Generation Comput Sys*. 2018;80:215-228.
47. Reisig, W., Towards a conceptual foundation of service composition. Computer Science - Research and Development, 2017.
48. Chaib A, Boussebough I, Chaoui A. Adaptive service composition in an ambient environment with a multi-agent system. *J Ambient Intell Humanized Computing*. 2018;9(2):367-380.
49. Wang H, Yang D, Yu Q, Tao Y. Integrating modified cuckoo algorithm and creditability evaluation for QoS-aware service composition. *Knowledge-Based Systems*. 2018;140:64-81.
50. Jian C, Li M, Kuang X. Edge cloud computing service composition based on modified bird swarm optimization in the internet of things. *Cluster Computing*. 2018.
51. Asghari S, Navimipour NJ. Nature inspired meta-heuristic algorithms for solving the service composition problem in the cloud environments. *Int J Commun Sys*. 2018;31(12):e3708.

52. Han SN, Khan I, Lee GM, Crespi N, Glitho RH. Service composition for IP smart object using realtime web protocols: concept and research challenges. *Computer Standards & Interfaces*. 2016;43:79-90.
53. Jula A, Sundararajan E, Othman Z. Cloud computing service composition: a systematic literature review. *Expert Sys Appl*. 2014;41(8):3809-3824.
54. Lemos AL, Daniel F, Benatallah B. Web service composition: a survey of techniques and tools. *ACM Comput Surv*. 2015;48(3):1-41.
55. Garriga M, Flores A, Cechich A, Zunino A. Web services composition mechanisms: a review. *IETE Technical Review*. 2015;32(5):376-383.
56. Alyari F, Jafari Navimipour N. Recommender systems: a systematic review of the state of the art literature and suggestions for future research. *Kybernetes*. 2018;47(5):985-1017.
57. Charband Y, Jafari Navimipour N. Online knowledge sharing mechanisms: a systematic review of the state of the art literature and recommendations for future research. *Inf Syst Frontiers*. 2016;1-21.
58. Aznoli F, Navimipour NJ. Cloud services recommendation: reviewing the recent advances and suggesting the future research directions. *J Netw Comput Appl*. 2017;77:73-86.
59. Navimipour NJ, Vakili A. Comprehensive and systematic review of the service composition mechanisms in the cloud environments. *J Netw Comput Appl*. 2017.
60. Souri A, Rahmani AM. A survey for replica placement techniques in data grid environment. *Int J Modern Education Comput Sci*. 2014;6(5):46-51.
61. Souri A, Pashazadeh S, Navin AH. CONSISTENCY OF data replication protocols in database systems: a review. *Int J Inf Theory (IJIT)*. 2014;3(4):19-32.
62. Jamshidi P, Ahmad A, Pahl C. Cloud migration research: a systematic review. *IEEE Trans on Cloud Comput*. 2013;1(2):142-157.
63. Kitchenham B, Pretorius R, Budgen D, et al. Systematic literature reviews in software engineering—a tertiary study. *Inf Softw Technol*. 2010;52(8):792-805.
64. Pashazadeh A, Navimipour NJ. Big data handling mechanisms in the healthcare applications: a comprehensive and systematic literature review. *J Biomed Inform*. 2018;82:47-62.
65. Mauw, S., An algebraic specification of process algebra, including two examples, in algebraic methods: theory, tools and applications [papers from a workshop in Passau*, Germany, June 9-11, 1987]*. 1989, Springer-Verlag. p. 507-554.
66. Clarke EM, Grumberg JO, Peled DA. *Model Checking*. MIT Press; 1999:314.
67. Baier C, Katoen J-P. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press; 2008:975.
68. Cook, S.A.. The complexity of theorem-proving procedures, in *Proceedings of the third annual ACM symposium on Theory of computing*. 1971, ACM: Shaker Heights, Ohio, USA P 151-158.
69. Song, Z., A.A. Cardenas, and R. Masuoka. Semantic middleware for the Internet of Things. in 2010 Internet of Things (IOT) 2010.
70. Gyrard, A., M. Serrano, and G.A. Atemezing. Semantic web methodologies, best practices and ontology engineering applied to internet of things. in 2015 *IEEE 2nd World Forum on Internet of Things (WF-IoT)*. 2015.
71. Payam B, Wang W, Henson CA, Taylor K. Semantics for the internet of things: early progress and back to the future. *Int J Semantic Web and Information Systems (IJSWIS)*. 2012;8(1):1-21.
72. Jabbar S, Ullah F, Khalid S, Khan M, Han K. Semantic interoperability in heterogeneous IoT infrastructure for healthcare. *Wireless Commun Mobile Comput*. 2017;2017:10.
73. Gabrel V, Manouvrier M, Moreau K, Murat C. QoS-aware automatic syntactic service composition problem: complexity and resolution. *Future Generation Comput Sys*. 2017.
74. Safarkhanlou A, Souri A, Norouzi M, Sardroud SHEH. Formalizing and verification of an antivirus protection service using model checking. *Procedia Computer Science*. 2015;57:1324-1331.
75. Alodib M. QoS-aware approach to monitor violations of SLAs in the IoT. *J Innovation in Digital Ecosystems*. 2016;3(2):197-207.
76. Anastasi GF, Carlini E, Coppola M, Dazzi P. QoS-aware genetic cloud brokering. *Future Generation Comput Syst*. 2017;75(Supplement C):1-13.
77. Rai GN, Gangadharan GR, Padmanabhan V. Algebraic modeling and verification of web service composition. *Procedia Computer Science*. 2015;52:675-679.
78. Maroui, R. and B. Ayeb. Integrating the SysML and ACME in a model driven engineering approach to verify the web service composition. in 2015 *IEEE 24th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 2015.
79. Yan, B., X. Wu, and Q. Ye. Minimum cost composition algorithm for web service based on CSP. in 2011 *International Conference on Internet Technology and Applications*. 2011.
80. Wang HH, Gibbins N, Payne TR, Redavid D. A formal model of the semantic web service ontology (WSMO). *Inf Sys*. 2012;37(1):33-60.
81. Yeung WL. A formal and visual modeling approach to choreography based web services composition and conformance verification. *Expert Sys Appl*. 2011;38(10):12772-12785.

82. Laneve C, Padovani L. An algebraic theory for web service contracts. *Formal Aspects of Comput*. 2015;27(4):613-640.
83. Li, Y., Shenghui Zhao, Hailun Diao, Haibao Chen A formal validation method for trustworthy services composition. in 2016 *International Conference on Networking and Network Applications (NaNA)*. 2016.
84. Chen, F., et al. Modeling cross-organizational services composition with Pi-calculus. in *Proceedings of* 2011 *IEEE International Conference on Service Operations, Logistics and Informatics*. 2011.
85. Lahouij, A., L. Hamel, and M. Graiet. Formal modeling for verifying SCA dynamic composition with Event-B. in 2015 *IEEE 24th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 2015.
86. Babin, G., Y.A. Ameur, and M. Pantel. Formal verification of runtime compensation of web service compositions: a refinement and proof based proposal with Event-B. in 2015 *IEEE International Conference on Services Computing*. 2015.
87. Ait-Sadoune I, Ait-Ameur Y. Formal modelling and verification of transactional web service composition: a refinement and proof approach with Event-B. In: Thalheim B et al., eds. *Correct Software in Web Applications and Web Services*. Cham: Springer International Publishing; 2015:1-27.
88. Hamel, L., Mohamed Graiet, Mourad Kmimech, Mohamed Tahar Bhiri, Walid Gaaloul Verifying composite service transactional behavior with EVENT-B. in 2011 *Seventh International Conference on Semantics, Knowledge and Grids*. 2011.
89. Amato F, Moscato F. Pattern-based orchestration and automatic verification of composite cloud services. *Comput Electrical Eng*. 2016;56:842-853.
90. Wang, P., L. Yang, and G.W. Li, Mobile cloud computing system components composition formal verification method based on space-time pi-calculus, in *Cloud Computing: 6th International Conference, CloudComp 2015, Daejeon, South Korea, October 28-29, 2015, Revised Selected Papers*, Y. Zhang, L. Peng, and C.-H. Youn, Editors. 2016, Springer International Publishing: Cham. p. 159-167.
91. Ouchani S, Debbabi M. Specification, verification, and quantification of security in model-based systems. *Comput Secur*. 2015;97(7):691-711.
92. Zhang B, Orchard M, Saha B, Saxena A, Lee YJ, Vachtsevanos G. A verification framework with application to a propulsion system. *Expert Sys Appl*. 2014;41(13):5669-5679.
93. Souri A, Sharifloo MA, Norouzi M. Analyzing SMV & UPPAAL model checkers in real-time systems. *Global J Technol*. 2012;1.
94. Souri, A. and M. Norouzi. A new probable decision making approach for verification of probabilistic real-time systems. in *Software Engineering and Service Science (ICSESS),* 2015 *6th IEEE International Conference on*. 2015.
95. Al-Saqqar F, Bentahar J, Sultan K, Wan W, Asl EK. Model checking temporal knowledge and commitments in multi-agent systems using reduction. *Simulation Modelling Practice and Theory*. 2015;51:45-68.
96. Bataineh AS, Bentahar J, el Menshawy M, Dssouli R. Specifying and verifying contract-driven service compositions using commitments and model checking. *Expert Sys Appl*. 2017;74:151-184.
97. Groefsema HN, Beest V, Aiello M. A formal model for compliance verification of service compositions. *IEEE Trans Services Comput*. 2016;PP(99):-1. -1
98. Huynh KT, Quan TT, Bui TH. A bitwise-based indexing and heuristic-driven on-the-fly approach for web service composition and verification. *Vietnam J Comput Sci*. 2017;4(2):111-126.
99. Mi, C., et al. Reliability modeling and verification of BPEL-based web services composition by probabilistic model checking. in 2016 *IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*. 2016.
100. Nagamouttou D, Egambaram I, Krishnan M, Narasingam P. A verification strategy for web services composition using enhanced stacked automata model. *SpringerPlus*. 2015;4(1):98.
101. Ghannoudi, M. and W. Chainbi. Formal verification for web service composition: a model-checking approach. in 2015 *International Symposium on Networks, Computers and Communications (ISNCC)*. 2015.
102. Sun, M., S. Li, and Y. Ou. Model checking business processes for web service compositions in mCRL2. in 2014 *Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*. 2014.
103. Kil H, Nam W. Semantic web service composition via model checking techniques. *Int J of Web and Grid Services*. 2013;9(4):339-350.
104. Oussalah Y, Zeghib N. Formal specification of flexible web service composition. *Int J Critical Computer-Based Sys*. 2015;6(2):81-99.
105. Dumez C, Bakhouya M, Gaber J, Wack M, Lorenz P. Model-driven approach supporting formal verification for web service composition protocols. *J Netw Comput Appl*. 2013;36(4):1102-1115.
106. Mukherjee, A., Z. Tari, and P. Bertok. A spring based framework for verification of service composition. in 2011 *IEEE International Conference on Services Computing*. 2011.
107. Bianculli, D., et al., Interface decomposition for service compositions, in *Proceedings of the 33rd International Conference on Software Engineering*. 2011, ACM: Waikiki, Honolulu, HI, USA. P. 501-510.
108. Guermouche, N. and S.D. Zilio. Towards timed requirement verification for service choreographies. in *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. 2012.

109. Chen, M., et al., VeriWS: a tool for verification of combined functional and non-functional requirements of web service composition, in *Companion Proceedings of the 36th International Conference on Software Engineering*. 2014, ACM: Hyderabad, India P 564-567.

110. Luo, X., L. Lingjie, and Z. Mengmeng. An epistemic model checking approach for OWL-S web services. in 2012 *IEEE Symposium on Electrical & Electronics Engineering (EEESYM)*. 2012.

111. Bourne S, Szabo C, Sheng Q. Transactional behavior verification in business process as a service configuration. *IEEE Trans Services Comput*. 2017;PP(99):1. -1

112. Xi N, Sun C, Ma J, Shen Y. Secure service composition with information flow control in service clouds. *Future Generation Comput Sys*. 2015;49:142-148.

113. Salva S, Cao T-D. A model-based testing approach combining passive conformance testing and runtime verification: application to web service compositions deployed in clouds. In: Lee R, ed. *Software Engineering Research, Management and Applications*. Vol.99-116 Heidelberg: Springer International Publishing; 2014.

114. Du, Y., B. Yang, and W. Tan. A model checking approach to analyzing timed compatibility in mediation-aided composition of web services. in 2015 *IEEE International Conference on Web Services*. 2015.

115. Ibrahim, N. and I. Khalil. Verifying web services compositions using UPPAAL. in 2012 *International Conference on Computer Systems and Industrial Informatics*. 2012.

116. Zhang, G., et al., Specification and verification of data and time in web service composition, in *Web Information Systems and Mining: International Conference, WISM* 2011*, Taiyuan, China, September 24-25, 2011, Proceedings, Part II*, Z. Gong, et al., Editors. 2011, Springer Berlin Heidelberg: Berlin, Heidelberg p 436-443.

117. Emilia Cambronero M, Díaz G, Valero V, Martínez E. Validation and verification of web services choreographies by using timed automata. *The J Logic Algebraic Programming*. 2011;80(1):25-49.

118. Machado, E.Z.D.A., D.B. Claro, and A.M.S. Andrade, Generating correct compositions of semantic web services with respect to temporal constraints, in *Proceedings of the 18th Brazilian symposium on Multimedia and the web*. 2012, ACM: São Paulo/SP, Brazil p 197-206.

119. Yuan, M., et al. Towards a formal verification approach for business process coordination. in 2010 *IEEE International Conference on Web Services*. 2010.

120. Bourne, S., C. Szabo, and Q.Z. Sheng, TL-VIEWS: a tool for temporal logic verification of transactional behavior of web service compositions. 2014. p. 418-422.

121. Ben Azaiez, A. and Z. Sbaï, Model checking web services choreography, in *Enterprise and Organizational Modeling and Simulation: 11th International Workshop, EOMAS 2015, Held at CAiSE 2015, Stockholm, Sweden, June 8-9, 2015, Selected Papers*, J. Barjis, R. Pergl, and E. Babkin, Editors. 2015, Springer International Publishing: Cham. p. 171-186.

122. Bentahar J, Yahyaoui H, Kova M, Maamar Z. Symbolic model checking composite web services using operational and control behaviors. *Expert Sys Appl*. 2013;40(2):508-522.

123. Tout H, Mourad A, Talhi C, Otrok H. AOMD approach for context-adaptable and conflict-free web services composition. *Comput Electrical Eng*. 2015;44:200-217.

124. Rebai, S., et al. A service-oriented architecture (SOA) framework for choreography verification. in 2015 *IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*. 2015.

125. Li X, Fan Y, Sheng QZ, Maamar Z, Zhu H. A petri net approach to analyzing behavioral compatibility and similarity of web services. *IEEE Trans Syst Man Cybern Part A Syst Hum*. 2011;41(3):510-521.

126. Kacem, H.H., W. Sellami, and A.H. Kacem. A formal approach for the validation of web service orchestrations. in 2012 *IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 2012.

127. Zhao, H., et al. Research on formal modeling and verification of BPEL-based web service composition. in 2012 *IEEE/ACIS 11th International Conference on Computer and Information Science*. 2012.

128. Todica, V., M.F. Vaida, and M. Cremene. Formal verification in web services composition. in *Proceedings of* 2012 IEEE international conference on automation, quality and testing, Robotics 2012.

129. Chemaa S, Bouarioua M, Chaoui A. A high-level petri net based model for web services composition and verification. *Int J Comput Appl Technol*. 2015;51(4):306-323.

130. Quyet, T.H., Q.P. Thi, and D.B. Hoang, A method of verifying web service composition, in *Proceedings of the* 2010 *Symposium on Information and Communication Technology*. 2010, ACM: Hanoi, Viet Nam P 155-162.

131. Ni Y, Fan Y. Model transformation and formal verification for semantic web services composition. *Adv Eng Softw*. 2010;41(6):879-885.

132. Bourouis A, Klai K, Hadj-Alouane NB, Touati YE. On the verification of opacity in web services and their composition. *IEEE Trans Services Comput*. 2017;10(1):66-79.

133. Zhao Y, Xiao H, Wang Z, Pu G, Su T. The semantics and verification of timed service choreography. *Int J Comput Math*. 2014;91(3):384-402.

134. Rai GN, Gangadharan GR. Set partition and trace based verification of web service composition. *Procedia Comput Sci*. 2015;52:278-285.

135. Martinelli, F. and I. Matteucci, Partial model checking for the verification and synthesis of secure service compositions, in *Public Key Infrastructures, Services and Applications: 10th European Workshop, EuroPKI 2013, Egham, UK, September 12-13, 2013, Revised Selected Papers*, S. Katsikas and I. Agudo, Editors. 2014, Springer Berlin Heidelberg: Berlin, Heidelberg. P. 1-11.
136. Xiangyu, L., et al. Translation-based verification of web services composition via ZING. in 2010 *International Conference on Intelligent Computing and Integrated Systems*. 2010.
137. Yau Y-S, Chua F-F. Efficient run-time verification of web service composition. *Int J Web Eng Technol*. 2015;10(2):170-198.
138. Fan G, Yu H, Chen L, Liu D. A formal aspect-oriented method to model and analyse secure service composition. *Int J Autonomous Adaptive Commun Syst*. 2015;8(2-3):119-140.
139. Kokash, N., C. Krause, and E.P.d. Vink, Data-aware design and verification of service compositions with Reo and mCRL2, in *Proceedings of the* 2010 *ACM Symposium on Applied Computing*. 2010, ACM: Sierre, Switzerland P 2406-2413.
140. Menadjelia N. Towards a formal study of automatic failure recovery in protocol-based web service composition. *Service Oriented Computing Appl*. 2016;10(2):173-184.
141. Zhao, Y., et al. Formal analysis of behavioural equivalence for trustworthy and composite web services. in 2010 *7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing*. 2010.
142. Papapanagiotou, P. and J. Fleuriot. Formal verification of web services composition using linear logic and the pi-calculus. in 2011 IEEE Ninth European Conference on Web Services 2011.
143. Jeffords RD, Heitmeyer CL, Archer MM, Leonard EI. Model-based construction and verification of critical systems using composition and partial refinement. *Formal Methods Sys Des*. 2010;37(2):265-294.
144. Papapanagiotou, P., J. Fleuriot, and S. Wilson, Diagrammatically-driven formal verification of web-services composition, in *Proceedings of the 7th international conference on Diagrammatic Representation and Inference*. 2012, Springer-Verlag: Canterbury, UK P 241-255.
145. Hahn, M.G., et al., Formal semantics and expressiveness of a web service composition language, in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. 2013, ACM: Coimbra, Portugal P 1667-1673.
146. Ghobaei-Arani M, Rahmanian AA, Souri A. A moth-flame optimization algorithm for web service composition in cloud computing: simulation and verification. *Softw: Pract Exp*. 2018.
147. Díaz M, Martín C, Rubio B. State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing. *J Netw Comput Appl*. 2016;67:99-117.
148. Ganzha M, Paprzycki M, Pawłowski W, Szmeja P, Wasielewska K. Semantic interoperability in the internet of things: an overview from the INTER-IoT perspective. *J Netw Comput Appl*. 2017;81:111-124.
149. Shadroo S, Rahmani AM. Systematic survey of big data and data mining in internet of things. *Comput Netw*. 2018;139:19-47.
150. Hanani A, Rahmani AM, Sahafi A. A multi-parameter scheduling method of dynamic workloads for big data calculation in cloud computing. *The J Supercomputing*. 2017;73(11):4796-4822.
151. Hosseini Shirvani M, Rahmani AM, Sahafi A. An iterative mathematical decision model for cloud migration: A cost and security risk approach. *Softw: Pract Exp*. 2018;48(3):449-485.
152. Arunkumar G, Venkataraman N. A novel approach to address interoperability concern in cloud computing. *Procedia Comput Sci*. 2015;50:554-559.
153. Rezaei R, Chiew TK, Lee SP, Shams Aliee Z. A semantic interoperability framework for software as a service systems in cloud computing environments. *Expert Syst Appl*. 2014;41(13):5751-5770.
154. Entezari-Maleki R, Etesami SE, Ghorbani N, Niaki AA, Sousa L, Movaghar A. Modeling and evaluation of service composition in commercial multiclouds using timed colored petri nets. *IEEE Trans Syst, Man, Cybernetics: Syst*. 2017;1-15.

How to cite this article: Souri A, Rahmani AM, Jafari Navimipour N. Formal verification approaches in the web service composition: A comprehensive analysis of the current challenges for future research. *Int J Commun Syst*. 2018;31:e3808. https://doi.org/10.1002/dac.3808